RESEARCH CENTER

FIELD
Algorithmics, Programming, Software and Architecture

Activity Report 2015

Section New Results

Edition: 2016-03-21

ALGORITHMICS, COMPUTER ALGEBRA AND CRYPTOLOGY

ARCHITECTURE, LANGUAGES AND COMPILATION

EMBEDDED AND REAL-TIME SYSTEMS

PROOFS AND VERIFICATION

SECURITY AND CONFIDENTIALITY

ARIC Project-Team

7. New Results

7.1. Floating-point Arithmetic

7.1.1. On the maximum relative error when computing integer powers by iterated multiplications in floating-point arithmetic

We improve the usual relative error bound for the computation of x^n through iterated multiplications by x in binary floating-point arithmetic. The obtained error bound is only slightly better than the usual one, but it is simpler. We also discuss the more general problem of computing the product of n terms. [5]

7.1.2. Formally verified certificate checkers for hardest-to-round computation

In order to derive efficient and robust floating-point implementations of a given function f, it is crucial to compute its hardest-to-round points, i.e. the floating-point numbers x such that $f(x)$ is closest to the midpoint of two consecutive floating-point numbers. Depending on the floating-point format one is aiming at, this can be highly computationally intensive. In this paper, we show how certificates based on Hensel's lemma can be added to an algorithm using lattice basis reduction so that the result of a computation can be formally checked in the Coq proof assistant. [7]

7.1.3. On the error of Computing $ab + cd$ using Cornea, Harrison and Tang's method

In their book, Scientific Computing on the Itanium, Cornea et al. (2002) introduce an accurate algorithm for evaluating expressions of the form $ab + cd$ in binary floating-point arithmetic, assuming an FMA instruction is available. They show that if p is the precision of the floating-point format and if $u = 2^{-p}$, the relative error of the result is of order u. We improve their proof to show that the relative error is bounded by $2u + 7u^2 + 6u^3$. Furthermore, by building an example for which the relative error is asymptotically (as $p \to \infty$ or, equivalently, as $u \to 0$) equivalent to $2u$, we show that our error bound is asymptotically optimal. [8]

7.1.4. Improved error bounds for floating-point products and Horner's scheme

Let u denote the relative rounding error of some floating-point format. Recently it has been shown that for a number of standard Wilkinson-type bounds the typical factors $\gamma_k := ku/(1-ku)$ can be improved into ku, and that the bounds are valid without restriction on k. Problems include summation, dot products and thus matrix multiplication, residual bounds for LU- and Cholesky-decomposition, and triangular system solving by substitution. In this note we show a similar result for the product $\prod_{i=0}^{k} x_i$ of real and/or floating-point numbers x_i, for computation in any order, and for any base $\beta \geq 2$. The derived error bounds are valid under a mandatory restriction of k. Moreover, we prove a similar bound for Horner's polynomial evaluation scheme. [9]

7.1.5. Comparison between binary and decimal floating-point numbers

In collaboration with Christoph Lauter and Marc Mezzarobba (LIP6 laboratory, Paris), Nicolas Brisebarre and Jean-Michel Muller introduce an algorithm to compare a binary floating-point (FP) number and a decimal FP number, assuming the "binary encoding" of the decimal formats is used, and with a special emphasis on the basic interchange formats specified by the IEEE 754-2008 standard for FP arithmetic. It is a two-step algorithm: a first pass, based on the exponents only, quickly eliminates most cases, then, when the first pass does not suffice, a more accurate second pass is performed. They provide an implementation of several variants of our algorithm, and compare them [26].

7.2. Lattices: algorithms and cryptology

7.2.1. Linearly Homomorphic Encryption from DDH

We design a linearly homomorphic encryption scheme whose security relies on the hardness of the decisional Diffie-Hellman problem. Our approach requires some special features of the underlying group. In particular, its order is unknown and it contains a subgroup in which the discrete logarithm problem is tractable. Therefore, our instantiation holds in the class group of a non maximal order of an imaginary quadratic field. Its algebraic structure makes it possible to obtain such a linearly homomorphic scheme whose message space is the whole set of integers modulo a prime p and which supports an unbounded number of additions modulo p from the ciphertexts. A notable difference with previous works is that, for the first time, the security does not depend on the hardness of the factorization of integers. As a consequence, under some conditions, the prime p can be scaled to fit the application needs. [13]

7.2.2. Secure Efficient History-Hiding Append-Only Signatures in the Standard Model

As formalized by Kiltz et al. (ICALP'05), append-only signatures (AOS) are digital signature schemes where anyone can publicly append extra message blocks to an already signed sequence of messages. This property is useful, e.g., in secure routing, in collecting response lists, reputation lists, or petitions. Bethencourt, Boneh and Waters (NDSS'07) suggested an interesting variant, called history-hiding append-only signatures (HH-AOS), which handles messages as sets rather than ordered tuples. This HH-AOS primitive is useful when the exact order of signing needs to be hidden. When free of subliminal channels (i.e., channels that can tag elements in an undetectable fashion), it also finds applications in the storage of ballots on an electronic voting terminals or in other archival applications (such as the record of petitions, where we want to hide the influence among messages). However, the only subliminal-free HH-AOS to date only provides heuristic arguments in terms of security: Only a proof in the idealized (non-realizable) random oracle model is given. This paper provides the first HH-AOS construction secure in the standard model. Like the system of Bethencourt et al., our HH-AOS features constant-size public keys, no matter how long messages to be signed are, which is atypical (we note that secure constructions often suffer from a space penalty when compared to their random-oracle-based counterpart). As a second result, we show that, even if we use it to sign ordered vectors as in an ordinary AOS (which is always possible with HH-AOS), our system provides considerable advantages over existing realizations. As a third result, we show that HH-AOS schemes provide improved identity-based ring signatures (i.e., in prime order groups and with a better efficiency than the state-of-the-art schemes). [17]

7.2.3. Compactly Hiding Linear Spans: Tightly Secure Constant-Size Simulation-Sound QA-NIZK Proofs and Applications

Quasi-adaptive non-interactive zero-knowledge (QA-NIZK) proofs is a powerful paradigm, suggested recently by Jutla and Roy (Asiacrypt'13), which is motivated by the Groth-Sahai seminal techniques for efficient non-interactive zero-knowledge (NIZK) proofs. In this paradigm, the common reference string may depend on specific language parameters, a fact that allows much shorter proofs in important cases. It even makes certain standard model applications competitive with the Fiat-Shamir heuristic in the Random Oracle idealization (such QA-NIZK proofs were recently optimized to constant size by Jutla and Roy (Crypto'14) and Libert et al. (Eurocrypt'14) for the important case of proving that a vector of group elements belongs to a linear subspace). While, e.g., the QA-NIZK arguments of Libert et al. provide unbounded simulation-soundness and constant proof length, their simulation-soundness is only loosely related to the underlying assumption (with a gap proportional to the number of adversarial queries) and it is unknown how to alleviate this limitation without sacrificing efficiency. Here, we deal with the basic question of whether and to what extent we can simultaneously optimize the proof size and the tightness of security reductions, allowing for important applications with tight security (which are typically to date quite lengthy) to be of shorter size. In this paper, we resolve this question by describing a novel simulation-sound QA-NIZK argument showing that a vector v ∈ G n belongs to a subspace of rank t < n using a constant number of group elements. Unlike previous constant-size QA-NIZK proofs of such statements, the unbounded simulation-soundness of our system is nearly tightly related (i.e., the reduction only loses a factor proportional to the security parameter) to the standard Decision

Linear assumption. To show simulation-soundness in the constrained context of tight reductions, we employ a number of techniques, and explicitly point at a technique – which may be of independent interest – of hiding the linear span of a structure-preserving homomorphic signature (which is part of an OR proof). As an application, we design a public-key cryptosystem with almost tight CCA2-security in the multi-challenge, multiuser setting with improved length (asymptotically optimal for long messages). We also adapt our scheme to provide CCA security in the key-dependent message scenario (KDM-CCA2) with ciphertext length reduced by 75% when compared to the best known tightly secure KDM-CCA2 system so far. [18]

7.2.4. *Short Group Signatures via Structure-Preserving Signatures: Standard Model Security from Simple Assumptions*

Group signatures are a central cryptographic primitive which allows users to sign messages while hiding their identity within a crowd of group members. In the standard model (without the random oracle idealization), the most efficient constructions rely on the Groth-Sahai proof systems (Eurocrypt'08). The structure-preserving signatures of Abe et al. (Asiacrypt'12) make it possible to design group signatures based on well-established, constant-size number theoretic assumptions (a.k.a. "simple assumptions") like the Symmetric eXternal Diffie-Hellman or Decision Linear assumptions. While much more efficient than group signatures built on general assumptions, these constructions incur a significant overhead w.r.t. constructions secure in the idealized random oracle model. Indeed, the best known solution based on simple assumptions requires 2.8 kB per signature for currently recommended parameters. Reducing this size and presenting techniques for shorter signatures are thus natural questions. In this paper, our first contribution is to significantly reduce this overhead. Namely, we obtain the first fully anonymous group signatures based on simple assumptions with signatures shorter than 2 kB at the 128-bit security level. In dynamic (resp. static) groups, our signature length drops to 1.8 kB (resp. 1 kB). This improvement is enabled by two technical tools. As a result of independent interest, we first construct a new structure-preserving signature based on simple assumptions which shortens the best previous scheme by 25%. Our second tool is a new method for attaining anonymity in the strongest sense using a new CCA2-secure encryption scheme which is simultaneously a Groth-Sahai commitment. [19]

7.2.5. *Implementing Candidate Graded Encoding Schemes from Ideal Lattices*

Multilinear maps have become popular tools for designing cryptographic schemes since a first approximate realisation candidate was proposed by Garg, Gentry and Halevi (GGH). This construction was later improved by Langlois, Stehlé and Steinfeld who proposed GGHLite which offers smaller parameter sizes. In this work, we provide the first implementation of such approximate multilinear maps based on ideal lattices. Implementing GGH-like schemes naively would not allow instantiating it for non-trivial parameter sizes. We hence propose a strategy which reduces parameter sizes further and several technical improvements to allow for an efficient implementation. In particular, since finding a prime ideal when generating instances is an expensive operation, we show how we can drop this requirement. We also propose algorithms and implementations for sampling from discrete Gaussians, for inverting in some Cyclotomic number fields and for computing norms of ideals in some Cyclotomic number rings. Due to our improvements we were able to compute a multilinear jigsaw puzzle for $\kappa = 52$ (resp. $\kappa = 38$) and $\lambda = 52$ (resp. $\lambda = 80$). [10]

7.2.6. *Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance*

The Rényi divergence is a mean to measure the closeness of two distributions. We show that it can often be used as an alternative to the statistical distance in security proofs for lattice-based cryptography. Using the Rényi divergence is particularly suited for security proofs of primitives in which the attacker is required to solve a search problem (e.g., forging a signature). We show that it may also be used in the case of distinguishing problems (e.g., semantic security of encryption schemes), when they enjoy a public sampleability property. The techniques lead to security proofs for schemes with smaller parameters. [11]

7.2.7. *Fully Secure Functional Encryption for Inner Products, from Standard Assumptions*

Functional encryption is a modern public-key paradigm where a master secret key can be used to derive sub-keys SKF associated with certain functions F in such a way that the decryption operation reveals $F(M)$, if

M is the encrypted message, and nothing else. Recently, Abdalla et al. gave simple and effient realizations of the primitive for the computation of linear functions on encrypted data: given an encryption of a vector y over some specific base ring, a secret key SK_x for the vector x allows computing $< x, y >$. Their technique surprisingly allows for instantiations under standard assumptions, like the hardness of the Decision Diffie-Hellman (DDH) and Learning-with-Errors (LWE) problems. Their constructions, however, are only proved secure against selective adversaries, which have to declare the challenge messages M_0 and M_1 at the outset of the game. In this paper, we provide constructions that provably achieve security against more realistic adaptive attacks (where the messages M_0 and M_1 may be chosen in the challenge phase, based on the previously collected information) for the same inner product functionality. Our constructions are obtained from hash proof systems endowed with homomorphic properties over the key space. They are (almost) as efficient as those of Abdalla et al. and rely on the same hardness assumptions. In addition, we obtain a solution based on Paillier's composite residuosity assumption, which was an open problem even in the case of selective adversaries. We also propose LWE-based schemes that allow evaluation of inner products modulo a prime p, as opposed to the schemes of Abdalla et al. that are restricted to evaluations of integer inner products of short integer vectors. We finally propose a solution based on Paillier's composite residuosity assumption that enables evaluation of inner products modulo an RSA integer $N = pq$. We demonstrate that the functionality of inner products over a prime field is very powerful and can be used to construct bounded collusion FE for all circuits. [23]

7.2.8. Fully Homomophic Encryption over the Integers Revisited

Two main computational problems serve as security foundations of current fully homomorphic encryption schemes: Regev's Learning With Errors problem (LWE) and Howgrave-Graham's Approximate Greatest Common Divisor problem (AGCD). Our first contribution is a reduction from LWE to AGCD. As a second contribution, we describe a new AGCD-based fully homomorphic encryption scheme, which outperforms all prior AGCD-based proposals: its security does not rely on the presumed hardness of the so-called Sparse Subset Sum problem, and the bit-length of a ciphertext is only $\widetilde{O}\lambda$, where λ refers to the security parameter. [15]

7.2.9. Cryptanalysis of the Multilinear Map over the Integers

We describe a polynomial-time cryptanalysis of the (approximate) multilinear map of Coron, Lepoint and Tibouchi (CLT). The attack relies on an adaptation of the so-called zeroizing attack against the Garg, Gentry and Halevi (GGH) candidate multilinear map. Zeroizing is much more devastating for CLT than for GGH. In the case of GGH, it allows to break generalizations of the Decision Linear and Subgroup Membership problems from pairing-based cryptography. For CLT, this leads to a total break: all quantities meant to be kept secret can be efficiently and publicly recovered. [14]

7.2.10. Cryptanalysis of Gu's ideal multilinear map

In March, 2015 Gu Chunsheng proposed a candidate ideal multilinear map [eprint 2015/269]. An ideal multilinear map allows to perform as many multiplications as desired, while in k-multilinear maps like GGH [EC 2013] or CLT [CR2013, CR2015] one we canperform at most a predetermined number k of multiplications. In this note, we show that the extraction Multilinear Computational Diffie-Hellman problem (ext-MCDH) associated to Gu's map can be solved in polynomial-time: this candidate ideal multilinear map is insecure. We also give intuition on why we think that the two other ideal multilinear maps proposed by Gu in [eprint 2015/269] are not secure either. [39]

7.2.11. Worst-case to average-case reductions for module lattices

Most lattice-based cryptographic schemes are built upon the assumed hardness of the Short Integer Solution (SIS) and Learning With Errors (LWE) problems. Their efficiencies can be drastically improved by switching the hardness assumptions to the more compact Ring-SIS and Ring-LWE problems. However, this change of hardness assumptions comes along with a possible security weakening: SIS and LWE are known to be at least as hard as standard (worst-case) problems on euclidean lattices, whereas Ring-SIS and Ring-LWE are only known to be as hard as their restrictions to special classes of ideal lattices, corresponding to ideals of some polynomial rings. In this work, we define the Module-SIS and Module-LWE problems, which bridge SIS with Ring-SIS, and LWE with Ring-LWE, respectively. We prove that these average-case problems are at least as hard as standard lattice problems restricted to module lattices (which themselves generalize arbitrary and ideal lattices). As these new problems enlarge the toolbox of the lattice-based cryptographer, they could prove useful for designing new schemes. Importantly, the worst-case to average-case reductions for the module problems are (qualitatively) sharp, in the sense that there exist converse reductions. This property is not known to hold in the context of Ring-SIS/Ring-LWE: Ideal lattice problems could reveal easy without impacting the hardness of Ring-SIS/Ring-LWE. [6]

7.2.12. Reducing Communication Overhead of the Subset Difference Scheme

In Broadcast Encryption (BE) systems like Pay-TV, AACS, online content sharing and broadcasting, reducing the header length (communication overhead per session) is of practical interest. The Subset Difference (SD) scheme due to Naor-Naor-Lotspiech (NNL) is the most popularly used BE scheme. This work introduced the (a,b,γ) augmented binary tree subset difference ((a,b,γ)-ABTSD) scheme which is a generalization of the NNL-SD scheme. By varying the parameters (a,b,γ), it is possible to obtain $O(n\log n)$ different schemes. In addition to the underlying binary tree structure of the NNL-SD scheme, the new scheme uses an additional binary tree structure of height a augmented with each internal node. The SD subsets in this scheme arise due to nodes that are at a distance at most b from each other. In the augmented tree of height a, at most c leaves are considered together in creating the SD subsets for the scheme. The average header length achieved by the new schemes is smaller than all known schemes having the same decryption time as that of the NNL-SD scheme and achieving non-trivial trade-offs between the user storage and the header size. The amount of key material that a user is required to store increases. For the earlier mentioned applications, reducing header size and achieving fast decryption is perhaps more of a concern than the user storage

7.3. Algebraic computing and high performance kernels

7.3.1. Complexity of the F5 Gröbner basis algorithm

We study the complexity of Gröbner bases computation, in particular in the generic situation where the variables are in simultaneous Noether position with respect to the system. We give a bound on the number of polynomials of degree d in a Gröbner basis computed by Faugère's F5 algorithm (2002) in this generic case for the grevlex ordering (which is also a bound on the number of polynomials for a reduced Gröbner basis, independently of the algorithm used). Next, we analyse more precisely the structure of the polynomials in the Gröbner bases with signatures that F5 computes and use it to bound the complexity of the algorithm. Our estimates show that the version of F5 we analyse, which uses only standard Gaussian elimination techniques, outperforms row reduction of the Macaulay matrix with the best known algorithms for moderate degrees, and even for degrees up to the thousands if Strassen's multiplication is used. The degree being fixed, the factor of improvement grows exponentially with the number of variables. [1]

7.3.2. Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations

The interpolation step in the Guruswami-Sudan algorithm is a bivariate interpolation problem with multiplicities commonly solved in the literature using either structured linear algebra or basis reduction of polynomial lattices. This problem has been extended to three or more variables; for this generalization, all fast algorithms proposed so far rely on the lattice approach. In this work, we reduce this multivariate interpolation problem to a problem of simultaneous polynomial approximations, which we solve using fast structured linear algebra. This improves the best known complexity bounds for the interpolation step of the list-decoding of Reed-Solomon codes, Parvaresh-Vardy codes, and folded Reed-Solomon codes. In particular, for Reed-Solomon list-decoding with re-encoding, our approach has complexity $\widetilde{O}(\ell^{\omega-1}m^2(n-k))$, where ℓ, m, n, k are the list size, the multiplicity, the number of sample points and the dimension of the code, and ω is the exponent of linear algebra; this accelerates the previously fastest known algorithm by a factor of ℓ/m. [3]

7.3.3. Recursion based parallelization of exact dense linear algebra routines for Gaussian elimination

We present block algorithms and their implementation for the parallelization of sub-cubic Gaussian elimination on shared memory architectures. Contrarily to the classical cubic algorithms in parallel numerical linear algebra, we focus here on recursive algorithms and coarse grain parallelization. Indeed, sub-cubic matrix arithmetic can only be achieved through recursive algorithms making coarse grain block algorithms perform more efficiently than fine grain ones. This work is motivated by the design and implementation of dense linear algebra over a finite field, where fast matrix multiplication is used extensively and where costly modular reductions also advocate for coarse grain block decomposition. We incrementally build efficient kernels, for matrix multiplication first, then triangular system solving, on top of which a recursive PLUQ decomposition algorithm is built. We study the parallelization of these kernels using several algorithmic variants: either iterative or recursive and using different splitting strategies. Experiments show that recursive adaptive methods for matrix multiplication, hybrid recursive-iterative methods for triangular system solve and tile recursive versions of the PLUQ decomposition, together with various data mapping policies, provide the best performance on a 32 cores NUMA architecture. Overall, we show that the overhead of modular reductions is more than compensated by the fast linear algebra algorithms and that exact dense linear algebra matches the performance of full rank reference numerical software even in the presence of rank deficiencies. [4]

7.3.4. Computing the Rank Profile Matrix

The row (resp. column) rank profile of a matrix describes the staircase shape of its row (resp. column) echelon form. In an ISSAC'13 paper, we proposed a recursive Gaussian elimination that can compute simultaneously the row and column rank profiles of a matrix as well as those of all of its leading sub-matrices, in the same time as state of the art Gaussian elimination algorithms. Here we first study the conditions making a Gaus-sian elimination algorithm reveal this information. Therefore, we propose the definition of a new matrix invariant, the rank profile matrix, summarizing all information on the row and column rank profiles of all the leading sub-matrices. We also explore the conditions for a Gaussian elimination algorithm to compute all or part of this invariant, through the corresponding PLUQ decomposition. As a consequence, we show that the classical iterative CUP decomposition algorithm can actually be adapted to compute the rank profile matrix. Used, in a Crout variant, as a base-case to our ISSAC'13 implementation, it delivers a significant improvement in efficiency. Second, the row (resp. column) echelon form of a matrix are usually computed via different dedicated triangular decompositions. We show here that, from some PLUQ decompositions, it is possible to recover the row and column echelon forms of a matrix and of any of its leading sub-matrices thanks to an elementary post-processing algorithm. [16]

7.3.5. Formulas for Continued Fractions. An Automated Guess and Prove Approach

We describe a simple method that produces automatically closed forms for the coefficients of continued fractions expansions of a large number of special functions. The function is specified by a non-linear differential equation and initial conditions. This is used to generate the first few coefficients and from there a conjectured formula. This formula is then proved automatically thanks to a linear recurrence satisfied by some remainder terms. Extensive experiments show that this simple approach and its straightforward generalization to difference and q-difference equations capture a large part of the formulas in the literature on continued fractions. [20]

7.3.6. Algebraic Diagonals and Walks

The diagonal of a multivariate power series F is the univariate power series $\operatorname{Diag} F$ generated by the diagonal terms of F. Diagonals form an important class of power series; they occur frequently in number theory, theoretical physics and enumerative combinatorics. We study algorithmic questions related to diagonals in the case where F is the Taylor expansion of a bivariate rational function. It is classical that in this case $\operatorname{Diag} F$ is an algebraic function. We propose an algorithm that computes an annihilating polynomial for $\operatorname{Diag} F$. Generically, it is its minimal polynomial and is obtained in time quasi-linear in its size. We show that this minimal polynomial has an exponential size with respect to the degree of the input rational function. We then address the related problem of enumerating directed lattice walks. The insight given by our study leads to a new method for expanding the generating power series of bridges, excursions and meanders. We show that their first N terms can be computed in quasi-linear complexity in N, without first computing a very large polynomial equation. [12]

CARAMEL Project-Team

7. New Results

7.1. The Logjam attack against the discrete logarithm

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Paul Zimmermann.

Together with colleagues from the Prosecco project-team and with other colleagues, we exhibited a new attack again the TLS protocol when using discrete logarithms [15]. A proof-of-concept of the attack was demonstrated using the CADO-NFS software. This paper obtained the best paper award at the ACM CCS 2015 conference, and received significant media coverage both in the specialized and non-specialized press.

7.2. Other results related to discrete logarithm

Participant: Pierrick Gaudry [contact].

Our 2014 work [16], in collaboration with Barbulescu, Guillevic and Morain, improving the practical aspects of discrete logarithm computation in quadratic extensions and reducing the theoretical complexity in the “medium characteristic case” has been published in Eurocrypt 2015.

In collaboration with Barbulescu and Kleinjung we have proposed in [17] to revisit an old construction of Schirokauer for discrete logarithms in extension fields. It is well suited for problems coming from pairings where the primes often have a special form.

With Galbraith we wrote a survey about the discrete logarithm problem in the context of elliptic curves [13].

7.3. Fast arithmetic for faster integer multiplication

Participants: Svyatoslav Covanov [contact], Emmanuel Thomé.

The paper [20] describes an algorithm for the multiplication of two n-bit integers. It achieves the best asymptotic complexity bound $O(n \log n \cdot 4^{\log^* n})$ under a hypothesis on the distribution of generalized Fermat primes of the form $r^{2^\lambda}+1$. This hypothesis states that there always exists a sufficiently small interval in which we can find such a prime. Experimental results give evidence in favor of this assumption. A journal submission is planned shortly.

7.4. Certificates for exact linear algebra computations

Participant: Emmanuel Thomé [contact].

The paper [21], in collaboration with Jean-Guillaume Dumas and Erich Kaltofen, is a preliminary version of a research work that has then been pursued, and that solves an open question of proving the correctness of some specific linear algebra computations. It emerged from practical techniques which had been used for this purpose for a while, and for which improvements were obtained. Submission plans for this work are yet to be finalized.

7.5. Computing Jacobi’s theta function in quasi-linear time

Participant: Hugo Labrande [contact].

We designed a new algorithm that improves the complexity of computing the value of the Jacobi theta function, $\theta(z, \tau)$ to arbitrary precision [23]. The algorithm uses a quadratically convergent sequence similar to the complex AGM, as well as Newton’s method; its complexity is $O(\mathcal{M}(n) \log n)$ for computing the value up to an error bounded by 2^{-n}, which is an improvement over the state-of-the-art complexity of $O(\mathcal{M}(n)\sqrt{n})$. Here, $\mathcal{M}(n)$ denotes the time taken by a multiplication of two n-bit numbers. We provide bounds on the loss of significant digits incurred during the computation. The algorithm was implemented using GNU MPC, showing practical improvement over (our optimized implementation of) existing algorithms for precision above approximately $300,000$ bits. The paper was submitted to *Mathematics of Computation*.

7.6. Construction of sparse polynomial systems with many positive solutions

Participant: Pierre-Jean Spaenlehauer [contact].

In collaboration with Frédéric Bihan (Univ. Savoie Mont-Blanc), we propose a variant of the classical Viro method to construct polynomial systems with prescribed monomial support and many solutions whose coordinates are all positive [19]. This is an asymptotic construction which has strong connections with tropical and convex geometry, and which involves computational problems such as low-rank matrix completion.

7.7. Small certificates of inconsistency of quadratic fewnomial systems

Participant: Pierre-Jean Spaenlehauer [contact].

In collaboration with Jean-Charles Faugère (EPI PolSys) and Jules Svartz (Min. de Éducation Nationale), we studied the problem of certifying the inconsistency of sparse quadratic polynomial systems. Finding certificates of inconsistency is a classical problem in computational commutative algebra, and these certificates are in general of size exponential in the input size. We identify families of quadratic fewnomial systems for which there exist certificates of size linear in the size of the input and we propose algorithms to compute them in polynomial time.

7.8. Cracking passphrases based on famous sentences

Participant: Hugo Labrande [contact].

We proposed a method to attack passwords based on famous sentences, which are rather widespread [18]: we showed a method to construct large dictionaries using only publicly-available sources (e.g. Wikipedia) and modest computing power. The resulting dictionaries were able to crack millions of passphrases, among which a 55-character long one, and some that do not appear to have been cracked before. Our work thus shows that using famous sentences as passwords is not secure at all, as any attacker, even those with low skills and very modest computational resources, can guess them.

CASCADE Project-Team

6. New Results

6.1. Results

All the results of the team have been published in journals or conferences (see the list of publications). They are all related with the research program (see before) and the research projects (see after):

- New zero-knowledge proofs
- Advanced families of hash proofs
- More efficient constructions with lattices
- New e-cash constructions
- Advanced primitives for the privacy in the cloud
- Efficient functional encryption
- Various predicate encryption schemes
- Cryptanalysis of symmetric primitives
- New leakage-resilient primitives
- Stronger security with related-key security

CRYPT Team (section vide)

GALAAD2 Team

6. New Results

6.1. Certifying isolated singular points and their multiplicity structure

Participant: Bernard Mourrain.

The paper [4] presents two new constructions related to singular solutions of polynomial systems. The first is a new deflation method for an isolated singular root. This construction uses a single linear differential form defined from the Jacobian matrix of the input, and defines the deflated system by applying this differential form to the original system. The advantages of this new deflation is that it does not introduce new variables and the increase in the number of equations is linear instead of the quadratic increase of previous methods. The second construction gives the coefficients of the so-called inverse system or dual basis, which defines the multiplicity structure at the singular root. We present a system of equations in the original variables plus a relatively small number of new variables. We show that the roots of this new system include the original singular root but now with multiplicity one, and the new variables uniquely determine the multiplicity structure. Both constructions are "exact", meaning that they permit one to treat all conjugate roots simultaneously and can be used in certification procedures for singular roots and their multiplicity structure with respect to an exact rational polynomial system.

Joint work with Agnes Szanto, Department of Mathematics, North Carolina State University, Raleigh, USA; Jonathan D. Hauenstein, Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, USA.

6.2. On the construction of general cubature formula by flat extensions

Participants: Marta Abril-Bucero, Bernard Mourrain.

We describe a new method to compute general cubature formulae [5]. The problem is initially transformed into the computation of truncated Hankel operators with flat extensions. We then analyse the algebraic properties associated to flat extensions and show how to recover the cubature points and weights from the truncated Hankel operator. We next present an algorithm to test the flat extension property and to additionally compute the decomposition. To generate cubature formulae with a minimal number of points, we propose a new relaxation hierarchy of convex optimization problems minimizing the nuclear norm of the Hankel operators. For a suitably high order of convex relaxation, the minimizer of the optimization problem corresponds to a cubature formula. Furthermore cubature formulae with a minimal number of points are associated to faces of the convex sets. We illustrate our method on some examples, and for each we obtain a new minimal cubature formula.

This is a joint work with C. Bajaj (Univ. of Austin, Texas, USA).

6.3. A moment matrix approach to computing symmetric cubatures

Participants: Mathieu Collowald, Evelyne Hubert.

A quadrature is an approximation of the definite integral of a function by a weighted sum of function values at specified points, or nodes, within the domain of integration. Gaussian quadratures are constructed to yield exact results for any polynomial of degree $2r-1$ or less by a suitable choice of r nodes and weights. Cubature is a generalization of quadrature in higher dimension. Constructing a cubature amounts to find a linear form

$$\Lambda : \mathbb{R}[x] \to \mathbb{R}, p \mapsto \sum_{j=1}^{r} a_j\, p(\xi_j)$$

from the knowledge of its restriction to $\mathbb{R}[x]_{\leq d}$. The unknowns are the number of nodes r, the weights a_j and the nodes ξ_j.

In [7] we use a basis-free version of an approach to cubatures based on moment matrices in terms of the Hankel operator $\mathcal{H}$ associated to Λ. The existence of a cubature of degree d with r nodes boils down to conditions of ranks and positive semidefiniteness on $\mathcal{H}$. We then recognize the nodes as the solutions of a generalized eigenvalue problem.

Standard domains of integration are symmetric under the action of a finite group. It is natural to look for cubatures that respect this symmetry. Introducing adapted bases obtained from representation theory, the symmetry constraint allows to block diagonalize the Hankel operator $\mathcal{H}$. We then deal with smaller-sized matrices both for securing the existence of the cubature and computing the nodes. The sizes of the blocks are furthermore explicitly related to the orbit types of the nodes with the new concept of the matrix of multiplicities of a finite group. It provides preliminary criteria of existence of a cubature with a given organisation of the nodes in orbit types.

The Maple implementation of the presented algorithms allows to determine, with moderate computational efforts, all the symmetric cubatures of a given degree. We present new relevant cubatures.

6.4. Invariantization of symmetric polynomial systems

Participants: Mathieu Collowald, Evelyne Hubert.

Assuming the variety of a set of polynomials is invariant under a group action, we provide a set of invariants that define the same variety. The contribution is about infinite algebraic groups, the case of finite group being previously known. We introduce for those a new concept of algebraic invariantization. It is based on the construction of rational invariants by Hubert and Kogan [14], a construction for which we provide here new simplified proofs.

6.5. Effective criterions for bigraded birational maps

Participant: Laurent Busé.

A rational map $\mathcal{F} : \mathbb{P}^m \dashrightarrow \mathbb{P}^n$ between projective spaces is defined by a collection of homogeneous polynomials $\mathbf{f} := (f_0, ..., f_n)$ in $m + 1$ variables of the same degree. The problem of deciding or providing sufficient conditions for such a map $\mathcal{F}$ to be birational have attracted a lot of interest in the past and it is still an active area of research. Methods that are based of some properties of the syzygies of **f** are definitely the more adapted for computational purposes in the sense that they make the problem of birationality effectively computable in the usual implementation of the Gröbner basis algorithm. The goal of this work is to extend these syzygies-based methods and techniques to the context of rational maps whose source is a product of two projective spaces $\mathbb{P}^r \times \mathbb{P}^s$ instead.

An important motivation for considering bi-graded rational maps comes from the field of geometric modeling. Indeed, the geometric modeling community uses almost exclusively bi-graded rational maps for parameterizing curves, surfaces or volumes under the name of rational tensor-product Bézier parameterizations. It turns out that an important property is to guarantee the birationality of these parameterizations onto their images. An even more important property is to preserve this birationality property during a design process, that is to say when the coefficients of the defining polynomials are continuously modified. As a first attempt to tackle these difficult problems, we analyze in detail birational maps from $\mathbb{P}^1 \times \mathbb{P}^1$ to $\mathbb{P}^2$ in low bi-degree by means of syzygies.

This work is done in the context of the SYRAM project which is funded by the MathAmSud programme. It is a collaboration with N. Botbol (University of Buenos Aires), M. Chardin (University of Paris 6), H. Hassanzadeh (University of Rio de Janeiro), A. Simis (University de Pernambuco) and Q. H. Tran (University of Paris 6). A paper is in preparation.

6.6. Orthogonal projection of points on Bézier curves and surfaces

Participant: Laurent Busé.

In this work, we introduce a new method for computing the orthogonal projections of a point onto a Bézier curve or surface. It is based on the concept of matrix representation we have introduced and developed in some previous works, which is here applied to the parameterizations of the normal planes or lines of a curve or surface, respectively. It consists in the computation of a matrix depending of the ambient space variables, which is done in a pre-processing step, and then the use of tools from numerical linear algebra for a fast and accurate solving of each instance of the problem.

This is an on going work done in the context of the SYRAM project which is funded by the MathAmSud programme. It is a collaboration with N. Botbol (University of Buenos Aires) and M. Chardin (University of Paris 6).

6.7. Extraction of cylinders and cones from minimal point sets

Participants: Laurent Busé, André Galligo, Jiajun Zhang.

The extraction of geometric primitives from 3D point clouds is an important problem in reverse engineering. These 3D point clouds are typically obtained by means of accurate 3D scanners and there exists several methods for performing the 3D geometric primitives extraction. An important category among these method are based on a RANSAC method. For such methods, the primitives are directly extracted for the input point cloud. The basic idea is to extract a particular elementary type of shape, such as planes, spheres, cylinders, cones or tori, from the smallest possible set of points and then to judge if this extracted primitive is relevant to the full point cloud. Therefore, for this category of methods it is very important to compute a particular type of shape through the smallest possible number of points, including normals or not. If the extraction of planes and spheres is easy to treat, the cases of cylinders, cones and tori are more involved. In this work, we aim at developing methods for extracting these geometric primitives from the smaller possible number of points (counting multiplicities if normals are taken into account). Another objective is also to provide methods for extraction without using estimated normals in order to improve the accuracy of the extracted geometric primitive, or to use mixed data depending of the applied context (some points with normals and some other points without normals). A paper is in preparation.

6.8. Discriminant of a complete intersection space curve

Participant: Laurent Busé.

In this work, we develop the formalism of the discriminant of a complete intersection curve in the three dimensional projective space, that is to say a curve which is represented as the zero locus of two homogeneous polynomials in four variables. Our main objective is to provide a new computational approach to this object without relying on the so-called "Cayley trick" for which it is necessary to introduce new variables. We also aim at getting a universal definition of this discriminant over the integers so that it holds under any specialization of the coefficients to an arbitrary commutative ring. Another aspect of this work is to explore properties of this discriminant, typically invariance, covariance and change of basis properties.

This is an on going work which is done in collaboration with Ibrahim Nonkane (University of Ouagadougou, Burkina Faso).

6.9. Resultants, flexes, and the generalization of Salmon's formula

Participant: Laurent Busé.

Given an algebraic variety $S \subset \mathbb{P}^n$ and a point $p \in S$, the osculation order of the point p is the maximum of the multiplicity of intersection at p of S with any line through p. We denote it by μ_p and define $Flex(S) = \{p \in \mathbb{P}^n | \mu_p > n\}$.

If $n = 2$, it is known that if C is a plane algebraic curve of degree d then $Flex(C)$ is the intersection of C with its Hessian, this latter being of degree $3d - 6$. A famous generalization of this result to the case $n = 3$ has been obtained by Salmon in 1860: for a general variety S, $Flex(S)$ is the intersection of S with another hypersurface of degree $11d - 24$. In this work, we are studying the generalization of this formula to arbitrary dimension n. We proved that given $S \subset \mathbb{P}^n$ of degree d, $Flex(S)$ is obtained by intersecting S with another hypersurface of degree

$$d\left(\sum_{k=1}^{n} \frac{n!}{k}\right) - n!$$

We are also looking for an explicit expression of an equation of this latter hypersurface.

This is a work in progress which is done in the context of a PICS collaboration funded by CNRS. It is a joint work with M. Chardin (University Paris 6), C. D'Andrea (University of Barcelona), M. Sombra (University of Barcelona) and M. Weiman (University of Caen).

6.10. Computer Algebra Applied to a Solitary Waves Study

Participant: André Galligo.

In [3], we apply Computer algebra techniques, such as algebraic computations of resultants and discriminants, certified drawing (with a guaranteed topology) of plane curves, to a problem in Fluid dynamics: We investigate "capillary-gravity" solitary waves in shallow water, relying on the framework of the Serre-Green-Naghdi equations. So, we deal with 2 dimensional surface waves, propagating in a shallow water of constant depth. By a differential elimination process, the study reduces to describing the solutions of an ordinary non linear first order differential equation, depending on two parameters. The paper is illustrated with examples and pictures computed with the computer algebra system Maple.

Joint work with Didier Clamond (University of Nice, France) and Denys Dutykh (University of Le Bourget, France).

6.11. H1-parameterizations of plane physical domains with complex topology in Isogeometric analysis

Participants: André Galligo, Bernard Mourrain, Meng Wu.

Isogeometric analysis (IGA) is a method for solving geometric partial differential equations(PDEs). Generating parameterizations of a PDE's physical domain is a basic and important issue within IGA framework. In [13] , we present a global H1-parameterization method for a planar physical domain with complex topology.

Joint work with B. NKonga, Univeristy of Nice - Sophia Antipolis and EPI CASTOR, Inria.

GEOMETRICA Project-Team

7. New Results

7.1. Mesh Generation and Geometry processing

7.1.1. Discrete Derivatives of Vector Fields on Surfaces An Operator Approach

Participants: Frédéric Chazal, Maksim Ovsjanikov.

In collaboration with O. Azencot, M. Ben Chen (Technion, Israel Institute of Technology).

Vector fields on surfaces are fundamental in various applications in computer graphics and geometry processing. In many cases, in addition to representing vector fields, the need arises to compute their derivatives, for example, for solving partial differential equations on surfaces or for designing vector fields with prescribed smoothness properties. In this work, we consider the problem of computing the Levi-Civita covariant derivative, that is, the tangential component of the standard directional derivative, on triangle meshes. This problem is challenging since, formally, tangent vector fields on polygonal meshes are often viewed as being discontinuous, hence it is not obvious what a good derivative formulation would be. We leverage the relationship between the Levi-Civita covariant derivative of a vector field and the directional derivative of its component functions to provide a simple, easy-to-implement discretization for which we demonstrate experimental convergence. In addition, we introduce two linear operators which provide access to additional constructs in Riemannian geometry that are not easy to discretize otherwise, including the parallel transport operator which can be seen simply as a certain matrix exponential. Finally, we show the applicability of our operator to various tasks, such as fluid simulation on curved surfaces and vector field design, by posing algebraic constraints on the covariant derivative operator.

7.1.2. Isotopic Meshing within a Tolerance Volume

Participant: David Cohen-Steiner.

In collaboration with M. Mandad, P. Alliez (Titane Project-team).

We give an algorithm [22] that generates from an input tolerance volume a surface triangle mesh guaranteed to be within the tolerance, intersection free and topologically correct. A pliant meshing algorithm is used to capture the topology and discover the anisotropy in the input tolerance volume in order to generate a concise output. We first refine a 3D Delaunay triangulation over the tolerance volume while maintaining a piecewise-linear function on this triangulation, until an isosurface of this function matches the topology sought after. We then embed the isosurface into the 3D triangulation via mutual tessellation, and simplify it while preserving the topology. Our approach extends to surfaces with boundaries and to non-manifold surfaces. We demonstrate the versatility and efficacy of our approach on a variety of data sets and tolerance volumes.

7.1.3. CGALmesh: A Generic Framework for Delaunay Mesh Generation

Participants: Jean-Daniel Boissonnat, Clément Jamin, Mariette Yvinec.

In collaboration with P. Alliez (Titane Project-team).

CGALmesh [21] is the mesh generation software package of the Computational Geometry Algorithm Library (CGAL). It generates isotropic simplicial meshes—surface triangular meshes or volume tetrahedral meshes—from input surfaces, 3D domains, and 3D multidomains, with or without sharp features. The underlying meshing algorithm relies on restricted Delaunay triangulations to approximate domains and surfaces and on Delaunay refinement to ensure both approximation accuracy and mesh quality. CGALmesh provides guarantees on approximation quality and on the size and shape of the mesh elements. It provides four optional mesh optimization algorithms to further improve the mesh quality. A distinctive property of CGALmesh is its high flexibility with respect to the input domain representation. Such a flexibility is achieved through a careful software design, gathering into a single abstract concept, denoted by the oracle, all required interface features between the meshing engine and the input domain. We already provide oracles for domains defined by polyhedral and implicit surfaces.

7.2. Topological and Geometric Inference

7.2.1. Subsampling Methods for Persistent Homology

Participants: Frédéric Chazal, Bertrand Michel.

In collaboration with B.T. Fasy, F. Lecci, A. Rinaldo and L. Wasserman (Carnegie Mellon University).

Persistent homology is a multiscale method for analyzing the shape of sets and functions from point cloud data arising from an unknown distribution supported on those sets. When the size of the sample is large, direct computation of the persistent homology is prohibitive due to the combinatorial nature of the existing algorithms. We propose to compute the persistent homology of several subsamples of the data and then combine the resulting estimates. We study the risk of two estimators and we prove that the subsampling approach carries stable topological information while achieving a great reduction in computational complexity.

7.2.2. Efficient and Robust Persistent Homology for Measures

Participants: Frédéric Chazal, Steve Oudot.

In collaboration with M. Buchet (Ohio State University) and Donald Sheehy (University of Connecticut).

A new paradigm for point cloud data analysis has emerged recently, where point clouds are no longer treated as mere compact sets but rather as empirical measures. A notion of distance to such measures has been defined and shown to be stable with respect to perturbations of the measure. This distance can eas- ily be computed pointwise in the case of a point cloud, but its sublevel-sets, which carry the geometric infor- mation about the measure, remain hard to compute or approximate. This makes it challenging to adapt many powerful techniques based on the Euclidean distance to a point cloud to the more general setting of the distance to a measure on a metric space. We propose [28] an efficient and reliable scheme to approximate the topological structure of the family of sublevel-sets of the distance to a measure. We obtain an algorithm for approximating the persistent homology of the distance to an empirical measure that works in arbitrary metric spaces. Precise quality and complexity guarantees are given with a discussion on the behavior of our approach in practice.

7.2.3. Topological analysis of scalar fields with outliers

Participants: Frédéric Chazal, Steve Oudot.

In collaboration with M. Buchet, T.K. Dey, F. Fan, Y. Wang (Ohio State University).

Given a real-valued function f defined over a manifold M embedded in Euclidean space, we are interested in recovering structural information about f from the sole information of its values on a finite sample P [27]. Existing methods provide approximation to the persistence diagram of f when the noise is bounded in both the functional and geometric domains. However, they fail in the presence of aberrant values, also called outliers, both in theory and practice. We propose a new algorithm that deals with outliers. We handle aberrant functional values with a method inspired from the k-nearest neighbors regression and the local median filtering, while the geometric outliers are handled using the distance to a measure. Combined with topological results on nested filtrations, our algorithm performs robust topological analysis of scalar fields in a wider range of noise models than handled by current methods. We provide theoretical guarantees on the quality of our approximation and some experimental results illustrating its behavior.

7.2.4. Zigzag Persistence via Reflections and Transpositions

Participants: Clément Maria, Steve Oudot.

We introduce [33] a simple algorithm for computing zigzag persistence, designed in the same spirit as the standard persistence algorithm. Our algorithm reduces a single matrix, maintains an explicit set of chains encoding the persistent homology of the current zigzag, and updates it under simplex insertions and removals. The total worst-case running time matches the usual cubic bound.

A noticeable difference with the standard persistence algorithm is that we do not insert or remove new simplices "at the end" of the zigzag, but rather "in the middle". To do so, we use arrow reflections and transpositions, in the same spirit as reflection functors in quiver theory. Our analysis introduces a new kind of reflection called the "weak-diamond", for which we are able to predict the changes in the interval decomposition and associated compatible bases. Arrow transpositions have been studied previously in the context of standard persistent homology, and we extend the study to the context of zigzag persistence. For both types of transformations, we provide simple procedures to update the interval decomposition and associated compatible homology basis.

7.2.5. Stable Topological Signatures for Points on 3D Shapes

Participants: Mathieu Carrière, Steve Oudot, Maksims Ovsjanikovs.

Comparing points on 3D shapes is among the fundamental operations in shape analysis. To facilitate this task, a great number of local point signatures or descriptors have been proposed in the past decades. However, the vast majority of these descriptors concentrate on the local geometry of the shape around the point, and thus are insensitive to its connectivity structure. By contrast, several *global* signatures have been proposed that successfully capture the overall topology of the shape and thus characterize the shape as a whole. We propose [29], [43] the first point descriptor that captures the topology structure of the shape as 'seen' from a single point, in a multiscale and provably stable way. We also demonstrate how a large class of topological signatures, including ours, can be mapped to vectors, opening the door to many classical analysis and learning methods. We illustrate the performance of this approach on the problems of supervised shape labeling and shape matching. We show that our signatures provide complementary information to existing ones and allow to achieve better performance with less training data in both applications.

7.2.6. Structure and Stability of the 1-Dimensional Mapper

Participants: Mathieu Carrière, Steve Oudot.

Given a continuous function $f : X \to \mathbb{R}$ and a cover I of its image by intervals, the Mapper is the nerve of a refinement of the pullback cover $f^{-1}(I)$. Despite its success in applications, little is known about the structure and stability of this construction from a theoretical point of view. As a pixelized version of the Reeb graph of f, it is expected to capture a subset of its features (branches, holes), depending on how the interval cover is positioned with respect to the critical values of the function. Its stability should also depend on this positioning. We propose [44] a theoretical framework that relates the structure of the Mapper to the one of the Reeb graph, making it possible to predict which features will be present and which will be absent in the Mapper given the function and the cover, and for each feature, to quantify its degree of unstability. Using this framework, we can derive guarantees on the structure of the Mapper, on its stability, and on its convergence to the Reeb graph as the granularity of the cover I goes to zero.

7.2.7. Persistence Theory: From Quiver Representations to Data Analysis

Participant: Steve Oudot.

Persistence theory emerged in the early 2000s as a new theory in the area of applied and computational topology. This book [35] provides a broad and modern view of the subject, including its algebraic, topological, and algorithmic aspects. It also elaborates on applications in data analysis. The level of detail of the exposition has been set so as to keep a survey style, while providing sufficient insights into the proofs so the reader can understand the mechanisms at work.

7.3. Data Structures and Robust Geometric Computation

7.3.1. A probabilistic approach to reducing the algebraic complexity of computing Delaunay triangulations

Participant: Jean-Daniel Boissonnat.

In collaboration with Ramsay Dyer (Johann Bernoulli Institute, University of Groningen, Netherlands) and Arijit Ghosh (Max-Planck-Institut für Informatik, Saarbrücken, Germany).

Computing Delaunay triangulations in $\mathbb{R}^d$ involves evaluating the so-called in_sphere predicate that determines if a point x lies inside, on or outside the sphere circumscribing $d+1$ points $p_0, ..., p_d$. This predicate reduces to evaluating the sign of a multivariate polynomial of degree $d+2$ in the coordinates of the points $x, p_0, ..., p_d$. Despite much progress on exact geometric computing, the fact that the degree of the polynomial increases with d makes the evaluation of the sign of such a polynomial problematic except in very low dimensions. In this paper, we propose a new approach that is based on the witness complex, a weak form of the Delaunay complex introduced by Carlsson and de Silva. The witness complex $\mathrm{Wit}(L, W)$ is defined from two sets L and W in some metric space X: a finite set of points L on which the complex is built, and a set W of witnesses that serves as an approximation of X. A fundamental result of de Silva states that $\mathrm{Wit}(L, W) = \mathrm{Del}(L)$ if $W = X = \mathbb{R}^d$. In [25], [41], we give conditions on L that ensure that the witness complex and the Delaunay triangulation coincide when W is a finite set, and we introduce a new perturbation scheme to compute a perturbed set L' close to L such that $\mathrm{Del}(L') = \mathrm{Wit}(L', W)$. Our perturbation algorithm is a geometric application of the Moser-Tardos constructive proof of the Lovász local lemma. The only numerical operations we use are (squared) distance comparisons (i.e., predicates of degree 2). The time-complexity of the algorithm is sublinear in $|W|$. Interestingly, although the algorithm does not compute any measure of simplex quality, a lower bound on the thickness of the output simplices can be guaranteed.

7.3.2. Smoothed complexity of convex hulls

Participants: Marc Glisse, Rémy Thomasse.

In collaboration with O. Devillers (VEGAS Project-team) and X. Goaoc (Université Marne-la-Vallée)

We establish an upper bound on the smoothed complexity of convex hulls in $\mathbb{R}^d$ under uniform Euclidean (ℓ^2) noise. Specifically, let $\{p_1^*, p_2^*, ..., p_n^*\}$ be an arbitrary set of n points in the unit ball in $\mathbb{R}^d$ and let $p_i = p_i^* + x_i$, where $x_1, x_2, ..., x_n$ are chosen independently from the unit ball of radius δ. We show that the expected complexity, measured as the number of faces of all dimensions, of the convex hull of $\{p_1, p_2, ..., p_n\}$ is $O\left(n^{2-\frac{4}{d+1}}(1+1/\delta)^{d-1}\right)$; the magnitude δ of the noise may vary with n. For $d = 2$ this bound improves to $O\left(n^{\frac{2}{3}}(1+\delta^{-\frac{2}{3}})\right)$.

We also analyze the expected complexity of the convex hull of ℓ^2 and Gaussian perturbations of a nice sample of a sphere, giving a lower-bound for the smoothed complexity. We identify the different regimes in terms of the scale, as a function of n, and show that as the magnitude of the noise increases, that complexity varies monotonically for Gaussian noise but non-monotonically for ℓ^2 noise [31], [38].

7.3.3. Realization Spaces of Arrangements of Convex Bodies

Participant: Alfredo Hubard.

In collaboration with M. Dobbins (PosTech, South Korea) and A. Holmsen (KAIST, South Korea)

In [23], we introduce combinatorial types of arrangements of convex bodies, extending order types of point sets to arrangements of convex bodies, and study their realization spaces. Our main results witness a trade-off between the combinatorial complexity of the bodies and the topological complexity of their realization space. On one hand, we show that every combinatorial type can be realized by an arrangement of convex bodies and (under mild assumptions) its realization space is contractible. On the other hand, we prove a universality theorem that says that the restriction of the realization space to arrangements of convex polygons with a bounded number of vertices can have the homotopy type of any primary semialgebraic set.

7.3.4. Limits of order types

Participant: Alfredo Hubard.

In collaboration with X. Goaoc (Institut G. Monge), R. de Joannis de Verclos (CNRS-INPG), J-S. Sereni (LORIA), and J. Volec (ETH)

The notion of limits of dense graphs was invented, among other reasons, to attack problems in extremal graph theory. It is straightforward to define limits of order types in analogy with limits of graphs, and in [24] we examine how to adapt to this setting two approaches developed to study limits of dense graphs. We first consider flag algebras, which were used to open various questions on graphs to mechanical solving via semidefinite programming. We define flag algebras of order types, and use them to obtain, via the semidefinite method, new lower bounds on the density of 5- or 6-tuples in convex position in arbitrary point sets, as well as some inequalities expressing the difficulty of sampling order types uniformly. We next consider graphons, a representation of limits of dense graphs that enable their study by continuous probabilistic or analytic methods. We investigate how planar measures fare as a candidate analogue of graphons for limits of order types. We show that the map sending a measure to its associated limit is continuous and, if restricted to uniform measures on compact convex sets, a homeomorphism. We prove, however, that this map is not surjective. Finally, we examine a limit of order types similar to classical constructions in combinatorial geometry (Erdös-Szekeres, Horton...) and show that it cannot be represented by any somewhere regular measure; we analyze this example via an analogue of Sylvester's problem on the probability that k random points are in convex position.

GRACE Project-Team

7. New Results

7.1. Weight distribution of Algebraic-Geometry codes

V. Ducet worked on the weight distribution of geometric codes following a method initiated by Duursma. More precisely he implemented his method in magma and was able to compute the weight distribution of the geometric codes coming from two optimal curves of genus 2 and 3 over the finite fields of size 16 and 9 respectively. The aim is to compute the weight distribution of the Hermitian code over the finite field of size 16, for which computational improvements of the implementation are necessary.

7.2. Faster elliptic and hyperelliptic curve cryptography

B. Smith made several contributions to the development of faster arithmetic on elliptic curves and genus 2 Jacobians in 2015. First, an extended and more detailed treatment of his $\mathbb{Q}$-curve construction for endomorphism-accelerated elliptic curves (previously presented at ASIACRYPT 2013, and the basis of a successful implementation with C. Costello and H. Hisil presented at EUROCRYPT 2014) appeared in the Journal of Cryptology. A simplified approach to essential precomputations was published in the proceedings of AGCT-14. Finally, with C. Costello and P.-N. Chung, he gave a new, efficient, uniform, and constant-time scalar multiplication algorithm for genus 2 Jacobians exploiting fast Kummer surface arithmetic and features of differential addition chains.

7.3. Quantum factoring

Integer factorization via Shor's algorithm is a benchmark problem for general quantum computers, but surprisingly little work has been done on optimizing the algorithm for use as a serious factoring tool once large quantum computers are built (rather than as a proof of concept). In the meantime, given the limited size of contemporary quantum computers and the practical difficulties involved in building them, any optimizations to quantum factoring algorithms can lead to significant practical improvements. In a new interdisciplinary project with physicists F. Grosshans and T. Lawson, F. Morain and B. Smith have derived a simple new quantum factoring algorithm for cryptographic integers; its expected runtime is lower than Shor's factoring algorithm, and it should also be easier to implement in practice.

7.4. Cryptanalysis of code based cryptosystems by filtration attacks

The McEliece encryption scheme based on binary Goppa codes was one of the first public-key encryption schemes [35]. Its security rests on the difficulty of decoding an arbitrary code. The original proposal uses classical Goppa codes, and while it still remains unbroken, it requires a huge size of key. On the other hand, many derivative systems based on other families of algebraic codes have been subject to key recovery attacks. Up to now, key recovery attacks were based either on a variant of Sidelnikov and Shestakov's attack [36], where the first step involves the computation of minimum-weight codewords, or on the resolution of a system of polynomial equations using Gröbner bases.

In [3], A. Couvreur, P. Gaborit, V. Gauthier, A. Otmani and J.-P. Tillich introduced a new paradigm of attack called *filtration attacks*. The general principle decomposes in two steps:

1. **Distinguishing** the public code from a random one using the square code operation.
2. **Computing a filtration** of the public code using the distinguisher, and deriving from this filtration an efficient decoding algorithm for the public code.

This new style of attack allowed A. Couvreur, A. Otmani and J.-P. Tillich to break (in polynomial time) McEliece based on wild Goppa codes over quadratic extensions [7] and more recently to break the BBCRS cryptosystem [20]. A. Couvreur, Irene Márquez–Corbella, and R. Pellikaan broke McEliece based on algebraic geometry codes from curves of arbitrary genus [5], [6] by reconstructing optimal polynomial time decoding algorithms from the raw data of a generator matrix.

7.5. Quantum LDPC codes

Quantum codes are the analogous of error correcting codes for a quantum computer. A well known family of quantum codes are the CSS codes due to Calderbank, Shor and Steane can be represented by a pair of matrices (H_X, H_Z) such that $H_X H_Z^T = 0$. As in classical coding theory, if these matrices are sparse, then the code is said to be LDPC. An open problem in quantum coding theory is to get a family of quantum LDPC codes whose asymptotic minimum distance is in $\Omega(n^\alpha)$ for some $\alpha > 1/2$. No such family is known and actually, only few known families of quantum LDPC codes have a minimum distance tending to infinity.

In an article in preparation, Benjamin Audoux (I2M, Marseille) and A. Couvreur investigate a problem suggested by Bravyi and Hastings. They studied the behaviour of iterated tensor powers of CSS codes and prove in particular that such families always have a minimum distance tending to infinity. They propose also 3 families of LDPC codes whose minimum distance is in $\Omega(n^\beta)$ for all $\beta < 1/2$.

7.6. Discrete Logarithm computations in finite fields with the NFS algorithm

The best discrete logarithm record computations in prime fields and large characteristic finite fields are obtained with Number Field Sieve algorithm (NFS) at the moment. This algorithm is made of four steps:

1. polynomial selection;
2. relation collection (with a sieving technique);
3. linear algebra (computing the kernel of a huge matrix, of millions of rows and columns);
4. individual discrete logarithm computation.

The two more time consuming steps are the relation collection step and the linear algebra step. The polynomial selection is quite fast but is very important since it determines the complexity of the algorithm. Selecting better polynomials is a key to improve the overall running-time of the NFS algorithm. The final step: individual discrete logarithm, was though to be quite fast but F. Morain and A. Guillevic showed that it has an increasing complexity with respect to the extension degree of the finite field. A. Guillevic proposed a new method to reduce considerably the complexity, with at most a factor two speed-up in the exponent [22].

In 2015, F. Morain and A. Guillevic released with P. Gaudry and R. Barbulescu a major discrete logarithm record in a quadratic finite field GF(p^2) of 180 decimal digits (dd), corresponding to 595 bits. This was presented at the international conference Eurocrypt [19].

7.6.1. DL Record computation in a quadratic finite field GF(p^2)

In order to compare the practical running time of discrete logarithm computation in prime fields and quadratic finite fields, F. Morain and A. Guillevic with P. Gaudry and R. Barbulescu launched a DL record in a 180dd finite field. The last DL record in a prime field was held by the CARAMEL team of Nancy, in 2014, in a 180 dd prime field. The parameters chosen for the quadratic finite field are the following.

$$\begin{array}{rcl}
p & = & \texttt{314159265358979323846264338327950288419716939937510582097494459230}\diagdown \\
 & & \texttt{781640628620899877709223} \\
\ell & = & \texttt{392699081698724154807830422909937860524646174921888227621868074038}\diagdown \\
 & & \texttt{477050785776124847136536} \\
p-1 & = & 6 \cdot h_0 \text{ with } h_0 \text{ a 89 dd prime number} \\
p+1 & = & 8 \cdot \ell
\end{array}$$

The discrete logarithm computation was made modulo ℓ, the largest prime factor of the multiplicative subgroup $GF(p^2)^*$, so that a DL computation with generic methods of complexity $O(\sqrt{\ell})$ was impracticable.

The two polynomials used in the NFS algorithm were chosen to be the following:

$$\begin{aligned} f &= x^4 + 1 \\ g &= \texttt{448225077249286433565160965828828303618362474}\ x^2 - \texttt{29606109908476368046927513730655796265782} \\ &\quad + \texttt{448225077249286433565160965828828303618362474}\ . \end{aligned}$$

We indeed designed a new polynomial selection method, that we called the Conjugation method. It is very well suited for quadratic and cubic finite fields GF(p^2) and GF(p^3) for the size range of the records.

We finally computed the discrete logarithm in basis $G = T + 2$ of the target $s = \lfloor(\pi(2^{298})/8)\rfloor t + \lfloor(\gamma \cdot 2^{298})\rfloor$

$$\begin{aligned} \log_G s \quad &\equiv \quad \texttt{276214243617912804300337349268306605403758173819414418610}1\diagdown \\ &\quad\ \texttt{98322785683188853924304990580 12}\ \bmod\ \ell. \end{aligned}$$

The running time was very surprising: our record was much faster than the concurrent DL computation in a prime field of the same global size of 180dd, and even faster than the RSA modulus factorization of the same size.

Table 2. Comparison of running time for integer factorization (NFS-IF), discrete logarithm in prime field (NFS-DL(p)) and in quadratic field (NFS-DL(p 2)) of same global size 180 dd.

Algorithm	relation collection	linear algebra	total
NFS-IF	5 years	5.5 months	5.5 years
NFS-DL(p)	50 years	80 years	130 years
NFS-DL(p^2)	157 days	18 days (GPU)	0.5 years

7.6.2. *Individual discrete logarithm computation*

A big difference between prime fields and finite fields of small extension such as GF(p^3), GF(p^4) and GF(p^6) is the complexity of the final step of the NFS algorithm: computing the individual discrete logarithm of the target, given the large table of discrete logarithm of *small* elements. This table was obtained at the end of the linear algebra step. The target needs to be decomposed into small enough elements whose discrete logarithm is in the table, so that one can recompose the discrete logarithm of the target. This decomposition is quite fast for prime fields but we realized that is becomes more and more time consuming when the extension degree increase. A. Guillevic developed a new technique to improve considerably this step. The main idea is to use the structure of the finite field: the subfields. These improvements were presented at the Asiacrypt 2015 conference in Auckland, New Zealand and published in the proceedings [22].

7.7. Information sets of multiplicity codes

The codes we used in our PIR protocols, namely Reed-Muller and their generalization Multiplicity codes, are locally *correctable* : that means that local decoding allows to retrieve encoded symbols. In most applications, it is very desirable to retrieve *information* symbols. Another line of work in this topic was thus to find an encoding method for multiplicity codes so as to directly recover an information symbol from local correction, and not an encoded symbol. To do so we defined information sets for multiplicity codes, and design a systematic encoding based on this information set. This work was presented at ISIT'2015 in Hong-Kong in June [18].

7.8. Rank metric codes over infinite fields

Rank metric and Gabidulin codes over the rationals promise interesting applications to space-time coding. We have constructed optimal codes, similar to Gabidulin codes, in the case of infinite fields. We use algebraic extensions, and we have determined the condition on the considered extension to enable this construction. For example: we can design codes with complex coefficients, using number fields and Galois automorphisms. Then, in the rank metric setting, codewords can be seen as matrices. In this setting, a channel introduces errors (a matrix of small rank r added to the codeword) and erasures (s_r rows and s_c columns of the matrix are erased). We have developed an algorithm (adapted from the Welch–Berlekamp algorithm) to recover the right codeword in the presence of an error of rank weight up to $r + s_c + s_r \leqslant d - 1$, where d is the minimal distance of the code. As opposed to the finite field case, we are confronted by coefficient size growth. We solve this problem by computing modulo prime ideals. Using these codes we can completely bypass intermediate constructions using finite fields, which were the stumbling-block in classic constructions.

We also have used this framework to build rank-metric codes over the field of rational functions, using algebraic function fields with cyclic Galois group (Kummer and Artin extensions). These codes can be seen as a generator of infinitely many convolutional codes.

7.9. Hash function cryptanalysis

Cryptographic hash functions are versatile primitives that are used in many cryptographic protocols. The security of a hash function h is usually evaluated through two main notions: its preimage resistance (given a target t, the difficulty of finding a message m s.t. $h(m) = t$) and its collision resistance (the difficulty of finding two messages m, m' s.t. $h(m) = h(m')$).

A popular hash function is the SHA-1 algorithm. Although theoretical collision attacks were found in 2005, it is still being used in some applications, for instance as the hash function in some TLS certificates. Hence cryptanalysis of SHA-1 is still a major topic in cryptography.

In 2015, we improved the state-of-the-art on SHA-1 analysis in two ways:

- T. Espitau, P.-A. Fouque and P. Karpman improved the previous preimage attacks on SHA-1, reaching up to 62 rounds (out of 80), up from 57. The corresponding paper was published at CRYPTO 2015 [21].
- P. Karpman, T. Peyrin and M. Stevens developed collision attacks on the compression function of SHA-1 (i.e. freestart collisions). This exploits a model that is slightly more generous to the attacker in order to find explicit collisions on more rounds than what was previously possible. A first work resulted in freestart collisions for SHA-1 reduced to 76 steps; this attack takes less than a week to compute on a common GPU. The corresponding paper was published at CRYPTO 2015 [24]. This was later improved to attack the full compression function. Although the attack is more expensive it is still practical, taking less than two weeks on a 64 GPU cluster. The corresponding paper is currently under review for EUROCRYPT 2016 [32].

7.10. Block cipher design and analysis

Block ciphers are one of the most basic cryptographic primitives, yet block cipher analysis is still a major research topic. In recent years, the community also shifted focus to the more general setting of *authenticated encryption*, where one specifies an (set of) algorithm(s) providing both encryption and authentication for messages of arbitrary length. A major current event in that direction is the CAESAR academic competition, which aims to select a portfolio of good algorithms.

During this year, we helped to improve the state of the art in block cipher research in several ways:

- P. Karpman found a very efficient related-key attack on the CAESAR candidate Prøst-OTR. A related-key model is very generous to the attacker, but the attack in this case can be run instantaneously. The corresponding paper was published at ISC 2015 [23]

- B. Minaud, P. Derbez, P.-A. Fouque and P. Karpman developed a family of attacks that breaks all the remaining unbroken instances of the ASASA construction, that was presented at ASIACRYPT 2014. Using algebraic properties of the ciphers, for each type of instance, the attack allows to recover an algorithm equivalent to the secret key in near-practical time. This applies to a multivariate public-key scheme, a classical block cipher and small block ciphers used in white-box constructions. The corresponding paper was published at ASIACRYPT 2015 and was honoured as one of the three best papers of the conference [25].
- P. Karpman developed a compact 8-bit S-box with branch number three, which can be used as a basis to construct a lightweight block cipher particularly efficient on 8-bit microcontrollers. The corresponding paper is currently under review for FSE 2016.

LFANT Project-Team

6. New Results

6.1. Class groups and other invariants of number fields

Participants: Karim Belabas, Jean-Paul Cerri, Henri Cohen, Pınar Kılıçer, Pierre Lezowski.

Ohno and Nakagawa have proved relations between the counting functions of certain cubic fields. These relations may be viewed as complements to the Scholz reflection principle, and Ohno and Nakagawa deduced them as consequences of 'extra functional equations' involving the Shintani zeta functions associated to the prehomogeneous vector space of binary cubic forms. The paper [14] by Henri Cohen, Simon Rubinstein-Salzedo and Frank Thorne proves an identity relating certain degree fields with Galois groups D and F, respectively, for any odd prime, giving another proof of the Ohno–Nakagawa relation between the counting functions of certain cubic fields.

Pınar Kılıçer and Marco Streng have solved a variant of the class number 1 problem for quartic CM fields with a geometric motivation [27]; the question is whether a certain class group is trivial, which corresponds to a genus 2 curve with that complex multiplication being defined over a real-quadratic number field (instead of an extension). Using classical techniques provides a bound on the discriminant of such fields, which they refine taking ramification into account to obtain a practically useful bound. A carefully crafted enumeration algorithm finishes the proof.

In the article [28], P. Lezowski studies the Euclidean properties of matrix algebras $M_n(R)$ over commutative rings R. In particular, he shows that for any integer $n > 1$, $M_n(R)$ is a left and right Euclidean ring if and only if R is principal. The proof is constructive and allows to better understand the Euclidean order types of matrix algebras. Similar ideas are also applied to prove k-stage Euclidean properties of $M_n(R)$, linking them with Bézout property for the ring R. The article [28] has been submitted to *Journal of Algebra*.

The article by Aurel Page on the computation of arithmetic Kleinian groups has appeared [21].

6.2. Complex L-functions and certified arithmetic

Participants: Bill Allombert, Karim Belabas, Henri Cohen, Fredrik Johansson.

Fredrik Johansson's paper [23] has been published and presented at the 22nd IEEE Symposium on Computer Arithmetic (ARITH22), Lyon, France. This paper describes a new implementation of the elementary transcendental functions exp, sin, cos, log and atan for variable precision up to approximately 4096 bits. Compared to the MPFR library, it achieves a maximum speedup ranging from a factor 3 for cos to 30 for atan.

Bill Allombert, Karim Belabas, Henri Cohen and Pascal Molin (Paris 7) have implemented a new framework in PARI/GP to compute and manipulate complex L-functions and the associated ϑ and Λ functions, exporting 25 new functions to the GP computer algebra system.

6.3. Elliptic curve and Abelian varieties cryptology

Participants: Jean-Marc Couveignes, Andreas Enge, Enea Milio, Damien Robert.

In [29] David Lubicz and Damien Robert explain how to improve the arithmetic of Abelian and Kummer varieties. The speed of the arithmetic is a crucial factor in the performance of cryptosystems based on abelian varieties. Depending on the cryptographic application, the speed record holders are elliptic curves (in the Edwards model) or the Kummer surface of an hyperelliptic curves of genus 2 (in the level 2 theta model). One drawback of the Kummer surface is that only scalar multiplications are available, which may be a problem in certain cryptographic protocols. The previous known models to work on the Jacobian rather than the Kummer surface (Mumford coordinates or the theta model of level 4) are too slow and not competitive with elliptic curves. This paper explains how to use geometric properties (like projective normality) to speed up the arithmetic. In particular it introduces a novel addition algorithm on Kummer varieties (compatible addition), and uses it to speed up multi-exponentiations in Kummer varieties and to obtain new models of abelian surfaces in which the scalar multiplication is as fast as on the Kummer surface. This paper was written last year but heavily revised in 2015 and has been accepted (up to minor revisions) in the journal Finite Fields and Their Applications.

The paper [19] by David Lubicz and Damien Robert about computing certain isogenies in quasi optimal time has been published in the LMS Journal of Computation and Mathematics and the paper [18] by the same authors about optimal pairing computation on abelian varieties has been published in the Journal of Symbolic Computation. This paper expands the article [15] by Romain Cosset and Damien Robert about the computation of (ℓ, ℓ)-isogenies in dimension 2 which has been published in Mathematics of Computation.

Enea Milio has published one of the main results of his PhD thesis [20]. He has generalised the work of Régis Dupont for computing modular polynomials in dimension 2 to new invariants. He describes an algorithm to compute modular polynomials for invariants derived from theta constants and proves under some heuristics that this algorithm is quasi-linear in its output size. Some properties of the modular polynomials defined from quotients of theta constants are analysed and experiments with an implementation are related.

The paper [16] by Jean-Marc Couveignes and Tony Ezome explaining how to efficiently evaluate functions, including Weil functions and canonical theta functions, on Jacobian varieties and their quotients has been published in the LMS Journal of Computation and Mathematics. This paper also describes a quasi-optimal algorithm to compute (l, l)-isogenies between Jacobians of genus two curves, using a compact representation and differential characterisation of isogenies.

In [26], Sorina Ionica and Emmanuel Thomé look at the structure of isogeny graphs of genus 2 Jacobians with maximal real multiplication. They generalise a result of Kohel's describing the structure of the endomorphism rings of the isogeny graph of elliptic curves. Their setting considers genus 2 jacobians with complex multiplication, with the assumptions that the real multiplication subring is maximal and has class number 1. Over finite fields, they derive a depth first search algorithm for computing endomorphism rings locally at prime numbers, if the real multiplication is maximal.

6.4. Cryptology with quadratic fields

Participant: Guilhem Castagnos.

In [22] Guilhem Castagnos and Fabien Laguillaumie design a linearly homomorphic encryption scheme the security of which relies on the hardness of the decisional Diffie-Hellman problem. The approach requires some special features of the underlying group. In particular, its order is unknown and it contains a subgroup in which the discrete logarithm problem is tractable. Therefore, their instantiation holds in the class group of a non-maximal order of an imaginary quadratic field. Its algebraic structure makes it possible to obtain such a linearly homomorphic scheme in which the message space is the whole set of integers modulo a prime p and which supports an unbounded number of additions modulo p from the ciphertexts. A notable difference with previous work is that, for the first time, the security does not depend on the hardness of the factorisation of integers. As a consequence, under some conditions, the prime p can be scaled to fit the application needs. This paper has beenpresented at the cryptographer's track at the RSA Conference 2015.

POLSYS Project-Team

6. New Results

6.1. Fundamental algorithms and structured polynomial systems

6.1.1. On the complexity of the F5 Gröbner basis algorithm

We study the complexity of Gröbner bases computation, in particular in the generic situation where the variables are in simultaneous Noether position with respect to the system.

We give a bound on the number of polynomials of degree d in a Gröbner basis computed by F_5 algorithm in this generic case for the grevlex ordering (which is also a bound on the number of polynomials for a reduced Gröbner basis, independently of the algorithm used). Next, we analyse more precisely the structure of the polynomials in the Gröbner bases with signatures that F_5 computes and use it to bound the complexity of the algorithm.

Our estimates show that the version of F_5 we analyse, which uses only standard Gaussian elimination techniques, outperforms row reduction of the Macaulay matrix with the best known algorithms for moderate degrees, and even for degrees up to the thousands if Strassen's multiplication is used. The degree being fixed, the factor of improvement grows exponentially with the number of variables.

6.1.2. On the complexity of computing Gröbner bases for weighted homogeneous systems

Solving polynomial systems arising from applications is frequently made easier by the structure of the systems. Weighted homogeneity (or quasi-homogeneity) is one example of such a structure: given a system of weights $W = (w_1, ..., w_n)$, W-homogeneous polynomials are polynomials which are homogeneous w.r.t the weighted degree $\deg(X_1^{\alpha_1} \cdots X_n^{\alpha_n}) = \sum_{i=1}^{n} w_i \alpha_i$. Gröbner bases for weighted homogeneous systems can be computed by adapting existing algorithms for homogeneous systems to the weighted homogeneous case. In [6], we show that in this case, the complexity estimate for Algorithm F_5 $(\binom{n+d_{\max}-1}{d_{\max}}^{\omega}$ can be divided by a factor $(\prod_{i=1} w_i)^{\omega})$. For zero-dimensional systems, the complexity of Algorithm FGLM nD^{ω} (where D is the number of solutions of the system) can be divided by the same factor $(\prod_{i=1} w_i)^{\omega}$. Under genericity assumptions, for zero-dimensional weighted homogeneous systems of W-degree $(d_1, ..., d_n)$, these complexity estimates are polynomial in the weighted Bézout bound $\prod_{i=1}^{n} d_i / \prod_{i=1}^{n} w_i$. Furthermore, the maximum degree reached in a run of Algorithm F_5 is bounded by the weighted Macaulay bound $\sum_{i=1}^{n} (d_i - w_i) + w_n$, and this bound is sharp if we can order the weights so that $w_n = 1$. For overdetermined semi-regular systems, estimates from the homogeneous case can be adapted to the weighted case. We provide some experimental results based on systems arising from a cryptography problem and from polynomial inversion problems. They show that taking advantage of the weighted homogeneous structure yields substantial speed-ups, and allows us to solve systems which were otherwise out of reach.

6.1.3. Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences

Sakata generalized the Berlekamp – Massey algorithm to n dimensions in 1988. The Berlekamp – Massey – Sakata (BMS) algorithm can be used for finding a Gröbner basis of a 0-dimensional ideal of relations verified by a table. We investigate this problem using linear algebra techniques, with motivations such as accelerating change of basis algorithms (FGLM) or improving their complexity. In [12], we first define and characterize multidimensional linear recursive sequences for 0-dimensional ideals. Under genericity assumptions, we propose a randomized preprocessing of the table that corresponds to performing a linear change of coordinates on the polynomials associated with the linear recurrences. This technique then essentially reduces our problem to using the efficient 1-dimensional Berlekamp – Massey (BM) algorithm. However, the number of probes to the table in this scheme may be elevated. We thus consider the table in the *black-box* model: we assume probing the table is expensive and we minimize the number of probes to the table in our complexity model. We produce an FGLM-like algorithm for finding the relations in the table, which lets us use linear algebra techniques. Under some additional assumptions, we make this algorithm adaptive and reduce further the number of table probes. This number can be estimated by counting the number of distinct elements in a multi-Hankel matrix (a multivariate generalization of Hankel matrices); we can relate this quantity with the *geometry* of the final staircase. Hence, in favorable cases such as convex ones, the complexity is essentially linear in the size of the output. Finally, when using the LEX ordering, we can make use of fast structured linear algebra similarly to the Hankel interpretation of Berlekamp – Massey.

6.1.4. Nearly optimal computations with structured matrices

In [9] we estimate the Boolean complexity of multiplication of structured matrices by a vector and the solution of nonsingular linear systems of equations with these matrices. We study four basic and most popular classes, that is, Toeplitz, Hankel, Cauchy and Vandermonde matrices, for which the cited computational problems are equivalent to the task of polynomial multiplication and division and polynomial and rational multipoint evaluation and interpolation. The Boolean cost estimates for the latter problems have been obtained by Kirrinnis in [10], except for rational interpolation. We supply them now as well as the Boolean complexity estimates for the important problems of multiplication of transposed Vandermonde matrix and its inverse by a vector. All known Boolean cost estimates for such problems rely on using Kronecker product. This implies the d-fold precision increase for the d-th degree output, but we avoid such an increase by relying on distinct techniques based on employing FFT. Furthermore we simplify the analysis and make it more transparent by combining the representations of our tasks and algorithms both via structured matrices and via polynomials and rational functions. This also enables further extensions of our estimates to cover Trummer's important problem and computations with the popular classes of structured matrices that generalize the four cited basic matrix classes, as well as the transposed Vandermonde matrices. It is known that the solution of Toeplitz, Hankel, Cauchy, Vandermonde, and transposed Vandermonde linear systems of equations is generally prone to numerical stability problems, and numerical problems arise even for multiplication of Cauchy, Vandermonde, and transposed Vandermonde matrices by a vector. Thus our FFT-based results on the Boolean complexity of these important computations could be quite interesting because our estimates are reasonable even for more general classes of structured matrices, showing rather moderate growth of the complexity as the input size increases.

6.2. Solving Polynomial Systems over the Reals and Applications

6.2.1. Probabilistic Algorithm for Computing the Dimension of Real Algebraic Sets

Let $f \in \mathbb{Q}[X_1, ..., X_n]$ be a polynomial of degree D. We consider the problem of computing the real dimension of the real algebraic set defined by $f = 0$. Such a problem can be reduced to quantifier elimination. Hence it can be tackled with Cylindrical Algebraic Decomposition within a complexity that is doubly exponential in the number of variables. More recently, denoting by d the dimension of the real algebraic set under study, deterministic algorithms running in time $D^{O(d(n-d))}$ have been proposed. However, no implementation reflecting this complexity gain has been obtained and the constant in the exponent remains unspecified. In [11], we design a probabilistic algorithm which runs in time which is essentially cubic in $D^{d(n-d)}$. Our algorithm takes advantage of genericity properties of polar varieties to avoid computationally difficult steps of quantifier elimination. We also report on a first implementation. It tackles examples that are out of reach of the state-of-the-art and its practical behavior reflects the complexity gain.

6.2.2. Real root finding for determinants of linear matrices

Let $A_0, A_1, ..., A_n$ be given square matrices of size m with rational coefficients. The paper [7] focuses on the exact computation of one point in each connected component of the real determinantal variety $\{x \in \mathbb{R}^n : \det(A_0 + x_1 A_1 + \cdots + x_n A_n) = 0\}$. Such a problem finds applications in many areas such as control theory, computational geometry, optimization, etc. Using standard complexity results this problem can be solved using $m^{O(n)}$ arithmetic operations. Under some genericity assumptions on the coefficients of the matrices, we provide in an algorithm solving this problem whose runtime is essentially quadratic in $\binom{n+m}{n}^3$. We also report on experiments with a computer implementation of this algorithm. Its practical performance illustrates the complexity estimates. In particular, we emphasize that for subfamilies of this problem where m is fixed, the complexity is polynomial in n.

6.2.3. Real root finding for rank defects in linear Hankel matrices

Let $H_0, ..., H_n$ be $m \times m$ matrices with entries in $\mathbb{Q}$ and Hankel structure, i.e. constant skew diagonals. We consider the linear Hankel matrix $H(X) = H_0 + X_1 H_1 + \cdots + X_n H_n$ and the problem of computing sample points in each connected component of the real algebraic set defined by the rank constraint $\mathsf{rank}(H(X)) \leq r$, for a given integer $r \leq m - 1$. Computing sample points in real algebraic sets defined by rank defects in linear matrices is a general problem that finds applications in many areas such as control theory, computational geometry, optimization, etc. Moreover, Hankel matrices appear in many areas of engineering sciences. Also, since Hankel matrices are symmetric, any algorithmic development for this problem can be seen as a first step towards a dedicated exact algorithm for solving semi-definite programming problems, i.e. linear matrix inequalities. Under some genericity assumptions on the input (such as smoothness of an incidence variety), we design in [18] a probabilistic algorithm for tackling this problem. It is an adaptation of the so-called critical point method that takes advantage of the special structure of the problem. Its complexity reflects this: it is essentially quadratic in specific degree bounds on an incidence variety. We report on practical experiments and analyze how the algorithm takes advantage of this special structure. A first implementation outperforms existing implementations for computing sample points in general real algebraic sets: it tackles examples that are out of reach of the state-of-the-art.

6.2.4. Optimizing a Parametric Linear Function over a Non-compact Real Algebraic Variety

In [17], we consider the problem of optimizing a parametric linear function over a non-compact real trace of an algebraic set. Our goal is to compute a representing polynomial which defines a hypersurface containing the graph of the optimal value function. Rostalski and Sturmfels showed that when the algebraic set is irreducible and smooth with a compact real trace, then the least degree representing polynomial is given by the defining polynomial of the irreducible hypersurface dual to the projective closure of the algebraic set. First, we generalize this approach to non-compact situations. We prove that the graph of the opposite of the optimal value function is still contained in the affine cone over a dual variety similar to the one considered in compact case. In consequence, we present an algorithm for solving the considered parametric optimization problem for generic parameters' values. For some special parameters' values, the representing polynomials of the dual variety can be identically zero, which give no information on the optimal value. We design a dedicated algorithm that identifies those regions of the parameters' space and computes for each of these regions a new polynomial defining the optimal value over the considered region.

6.2.5. Bounds for the Condition Number of Polynomials Systems with Integer Coefficients

Polynomial systems of equations are a central object of study in computer algebra. Among the many existing algorithms for solving polynomial systems, perhaps the most successful numerical ones are the homotopy algorithms. The number of operations that these algorithms perform depends on the condition number of the roots of the polynomial system. Roughly speaking the condition number expresses the sensitivity of the roots with respect to small perturbation of the input coefficients. A natural question to ask is how can we bound, in the worst case, the condition number when the input polynomials have integer coefficients? In [19] we address this problem and we provide effective bounds that depend on the number of variables, the degree and the maximum coefficient bitsize of the input polynomials. Such bounds allows to estimate the bit complexity of the algorithms that depend on the separation bound, like the homotopy algorithms, for solving polynomial systems.

6.2.6. Nearly Optimal Refinement of Real Roots of a Univariate Polynomial

In [10] we assume that a real square-free polynomial A has a degree d, a maximum coefficient bitsize τ and a real root lying in an isolating interval and having no nonreal roots nearby (we quantify this assumption). Then, we combine the *Double Exponential Sieve* algorithm (also called the *Bisection of the Exponents*), the bisection, and Newton iteration to decrease the width of this inclusion interval by a factor of $t = 2^{-L}$. The algorithm has Boolean complexity $\widetilde{O}_B(d^2\tau + dL)$. Our algorithms support the same complexity bound for the refinement of r roots, for any $r \le d$.

6.2.7. Accelerated Approximation of the Complex Roots and Factors of a Univariate Polynomial

The known algorithms approximate the roots of a complex univariate polynomial in nearly optimal arithmetic and Boolean time. They are, however, quite involved and require a high precision of computing when the degree of the input polynomial is large, which causes numerical stability problems. We observe that these difficulties do not appear at the initial stages of the algorithms, and in [8] we extend one of these stages, analyze it, and avoid the cited problems, still achieving the solution within a nearly optimal complexity estimates, provided that some mild initial isolation of the roots of the input polynomial has been ensured. The resulting algorithms promise to be of some practical value for root-finding and can be extended to the problem of polynomial factorization, which is of interest on its own right. We conclude with outlining such an extension, which enables us to cover the cases of isolated multiple roots and root clusters.

6.2.8. Polynomial Interrupt Timed Automata

Interrupt Timed Automata (ITA) form a subclass of stopwatch automata where reachability and some variants of timed model checking are decidable even in presence of parameters. They are well suited to model and analyze real-time operating systems. Here we extend ITA with polynomial guards and updates, leading to the class of polynomial ITA (polITA). In [13], we prove that reachability is decidable in 2EXPTIME on polITA, using an adaptation of the cylindrical algebraic decomposition algorithm for the first-order theory of reals using symbolic computation. Compared to previous approaches, our procedure handles parameters and clocks in a unified way. We also obtain decidability for the model checking of a timed version of CTL and for reachability in several extensions of polITA.

6.3. Solving Systems in Finite Fields, Applications in Cryptology and Algebraic Number Theory

6.3.1. Polynomial-Time Algorithms for Quadratic Isomorphism of Polynomials: The Regular Case

Let $\mathbf{f} = (f_1, ..., f_m)$ and $\mathbf{g} = (g_1, ..., g_m)$ be two sets of $m \ge 1$ nonlinear polynomials in $\mathbb{K}[x_1, ..., x_n]$ ($\mathbb{K}$ being a field). In [3], we consider the computational problem of finding – if any – an invertible transformation on the variables mapping $\mathbf{f}$ to $\mathbf{g}$. The corresponding equivalence problem is known as *Isomorphism of Polynomials with one Secret* (IP1S) and is a fundamental problem in multivariate cryptography. Amongst its applications, we can cite Graph Isomorphism (GI) which reduces to equivalence of cubic polynomials with respect to an invertible linear change of variables, according to Agrawal and Saxena. The main result is a randomized polynomial-time algorithm for solving IP1S for quadratic instances, a particular case of importance in cryptography. To this end, we show that IP1S for quadratic polynomials can be reduced to a variant of the classical module isomorphism problem in representation theory. We show that we can essentially *linearize* the problem by reducing quadratic-IP1S to test the orthogonal simultaneous similarity of symmetric matrices; this latter problem was shown by Chistov, Ivanyos and Karpinski (ISSAC 1997) to be equivalent to finding an invertible matrix in the linear space $\mathbb{K}^{n \times n}$ of $n \times n$ matrices over $\mathbb{K}$ and to compute the square root in a certain representation in a matrix algebra. While computing square roots of matrices can be done efficiently using numerical methods, it seems difficult to control the bit complexity of such methods. However, we present exact and polynomial-time algorithms for computing a representation of the square root of a matrix in $\mathbb{K}^{n \times n}$, for various fields (including finite fields), as a product of two matrices. Each coefficient of these matrices lie in an extension field of $\mathbb{K}$ of polynomial degree. We then consider #IP1S, the counting version of IP1S for quadratic instances. In particular, we provide a (complete) characterization of the automorphism group of homogeneous quadratic polynomials. Finally, we also consider the more general *Isomorphism of Polynomials* (IP) problem where we allow an invertible linear transformation on the variables *and* on the set of polynomials. A randomized polynomial-time algorithm for solving IP when $\mathbf{f} = (x_1^d, ..., x_n^d)$ is presented. From an algorithmic point of view, the problem boils down to factoring the determinant of a linear matrix (*i.e.* a matrix whose components are linear polynomials). This extends to IP a result of Kayal obtained for PolyProj.

6.3.2. Factoring $N = p^r q^s$ for Large r and s

Boneh *et al.* showed at Crypto 99 that moduli of the form $N = p^r q$ can be factored in polynomial time when $r \simeq \log p$. Their algorithm is based on Coppersmith's technique for finding small roots of polynomial equations. In [15] we show that $N = p^r q^s$ can also be factored in polynomial time when r or s is at least $(\log p)^3$; therefore we identify a new class of integers that can be efficiently factored. We also generalize our algorithm to moduli with k prime factors $N = \prod_{i=1}^k p_i^{r_i}$; we show that a non-trivial factor of N can be extracted in polynomial-time if one of the exponents r_i is large enough.

6.3.3. On the Complexity of the BKW Algorithm on LWE

This work [1] presents a study of the complexity of the Blum–Kalai–Wasserman (BKW) algorithm when applied to the Learning with Errors (LWE) problem, by providing refined estimates for the data and computational effort requirements for solving concrete instances of the LWE problem. We apply this refined analysis to suggested parameters for various LWE-based cryptographic schemes from the literature and compare with alternative approaches based on lattice reduction. As a result, we provide new upper bounds for the concrete hardness of these LWE-based schemes. Rather surprisingly, it appears that BKW algorithm outperforms known estimates for lattice reduction algorithms starting in dimension $n \approx 250$ when LWE is reduced to SIS. However, this assumes access to an unbounded number of LWE samples.

6.3.4. Structural Cryptanalysis of McEliece Schemes with Compact Keys

A very popular trend in code-based cryptography is to decrease the public-key size by focusing on subclasses of alternant/Goppa codes which admit a very compact public matrix, typically quasi-cyclic (QC), quasi-dyadic (QD), or quasi-monoidic (QM) matrices. In [5], we show that the very same reason which allows to construct a compact public-key makes the key-recovery problem intrinsically much easier. The gain on the public-key size induces an important security drop, which is as large as the compression factor p on the public-key. The fundamental remark is that from the $k \times n$ public generator matrix of a compact McEliece, one can construct a $k/p \times n/p$ generator matrix which is - from an attacker point of view - as good as the initial public-key. We call this new smaller code the folded code. Any key-recovery attack can be deployed equivalently on this smaller generator matrix. To mount the key-recovery in practice, we also improve the algebraic technique of Faugère, Otmani, Perret and Tillich (FOPT). In particular, we introduce new algebraic equations allowing to include codes defined over any prime field in the scope of our attack. We describe a so-called “structural elimination" which is a new algebraic manipulation which simplifies the key-recovery system. As a proof of concept, we report successful attacks on many cryptographic parameters available in the literature. All the parameters of CFS-signatures based on QD/QM codes that have been proposed can be broken by this approach. In most cases, our attack takes few seconds (the harder case requires less than 2 hours). In the encryption case, the algebraic systems are harder to solve in practice. Still, our attack succeeds against several cryptographic challenges proposed for QD and QM encryption schemes, but there are still some parameters that have been proposed which are out of reach for the methods given here. However, regardless of the key-recovery attack used against the folded code, there is an inherent weakness arising from Goppa codes with QM or QD symmetries. It is possible to derive from the public key a much smaller public key corresponding to the folding of the original QM or QD code, where the reduction factor of the code length is precisely the order of the QM or QD group used for reducing the key size. To summarize, the security of such schemes are not relying on the bigger compact public matrix but on the small folded code which can be efficiently broken in practice with an algebraic attack for a large set of parameters.

6.3.5. A Polynomial-Time Key-Recovery Attack on MQQ Cryptosystems

In [16], we investigate the security of the family of MQQ public key cryptosystems using multivariate quadratic quasigroups (MQQ). These cryptosystems show especially good performance properties. In particular, the

MQQ-SIG signature scheme is the fastest scheme in the ECRYPT benchmarking of cryptographic systems (eBACS). We show that both the signature scheme MQQ-SIG and the encryption scheme MQQ-ENC, although using different types of MQQs, share a common algebraic structure that introduces a weakness in both schemes. We use this weakness to mount a successful polynomial time key-recovery attack that finds an equivalent key using the idea of so-called good keys. In the process we need to solve a MinRank problem that, because of the structure, can be solved in polynomial-time assuming some mild algebraic assumptions. We highlight that our theoretical results work in characteristic 2 which is known to be the most difficult case to address in theory for MinRank attacks and also without any restriction on the number of polynomials removed from the public-key. This was not the case for previous MinRank like-attacks against $\mathcal{MQ}$ schemes. From a practical point of view, we are able to break an MQQ-SIG instance of 80 bits security in less than 2 days, and one of the more conservative MQQ-ENC instances of 128 bits security in little bit over 9 days. Altogether, our attack shows that it is very hard to design a secure public key scheme based on an easily invertible MQQ structure.

6.3.6. Algebraic Cryptanalysis of a Quantum Money Scheme The Noise-Free Case

In [14], we investigate the Hidden Subspace Problem (HSP_q) over $\mathbb{F}_q$ which is as follows:

Input : $p_1, ..., p_m, q_1, ..., q_m \in \mathbb{F}_q[x_1, ..., x_n]$ of degree $d \geq 3$ (and $n \leq m \leq 2n$).
Find : a subspace $A \subset {\mathbb{F}_q}^n$ of dimension $n/2$ (n is even) such that

$$p_i(A) = 0 \ \forall i \in \{1, ..., m\} \text{ and } q_j(A^\perp) = 0 \ \forall j \in \{1, ..., m\},$$

where $A^\perp$ denotes the orthogonal complement of A with respect to the usual scalar product in $\mathbb{F}_q$.

This problem underlies the security of the first public-key quantum money scheme that is proved to be cryptographically secure under a non quantum but classic hardness assumption. This scheme was proposed by S. Aaronson and P. Christiano at STOC'12. In particular, it depends upon the hardness of HSP_2. More generally, Aaronson and Christiano left as an open problem to study the security of the scheme for a general field $\mathbb{F}_q$. We present a randomized polynomial-time algorithm that solves the HSP_q for $q > d$ with success probability $\approx 1 - 1/q$. So, the quantum money scheme extended to $\mathbb{F}_q$ is not secure for big q. Finally, based on experimental results and a structural property of the polynomials that we prove, we conjecture that there is also a randomized polynomial-time algorithm solving the HSP_2 with high probability. To support our theoretical results we also present several experimental results confirming that our algorithms are very efficient in practice. We emphasize that S. Aaronson and P. Christiano proposes a non-noisy and a noisy version of the public-key quantum money scheme. The noisy version of the quantum money scheme remains secure.

6.3.7. Folding Alternant and Goppa Codes with Non-Trivial Automorphism Groups

The main practical limitation of the McEliece public-key encryption scheme is probably the size of its key. A famous trend to overcome this issue is to focus on subclasses of alternant/Goppa codes with a non trivial automorphism group. Such codes display then symmetries allowing compact parity-check or generator matrices. For instance, a key-reduction is obtained by taking quasi-cyclic (QC) or quasi-dyadic (QD) alternant/Goppa codes. We show that the use of such symmetric alternant/Goppa codes in cryptography introduces a fundamental weakness. It is indeed possible to reduce the key-recovery on the original symmetric public-code to the key-recovery on a (much) smaller code that has not anymore symmetries. This result [4] is obtained thanks to a new operation on codes called folding that exploits the knowledge of the automorphism group. This operation consists in adding the coordinates of codewords which belong to the same orbit under the action of the automorphism group. The advantage is twofold: the reduction factor can be as large as the size of the orbits, and it preserves a fundamental property: folding the dual of an alternant (resp. Goppa) code provides the dual of an alternant (resp. Goppa) code. A key point is to show that all the existing constructions of alternant/Goppa codes with symmetries follow a common principal of taking codes whose support is globally invariant under the action of affine transformations (by building upon prior works of T. Berger and A. Dür). This enables not only to present a unified view but also to generalize the construction of QC, QD and even quasi-monoidic (QM) Goppa codes. All in all, our results can be harnessed to boost up any key-recovery attack on McEliece systems based on symmetric alternant or Goppa codes, and in particular algebraic attacks.

6.3.8. Improved Sieving on Algebraic Curves

The best algorithms for discrete logarithms in Jacobians of algebraic curves of small genus are based on index calculus methods coupled with large prime variations. For hyperelliptic curves, relations are obtained by looking for reduced divisors with smooth Mumford representation (Gaudry); for non-hyperelliptic curves it is faster to obtain relations using special linear systems of divisors (Diem, Diem and Kochinke). Recently, Sarkar and Singh have proposed a sieving technique, inspired by an earlier work of Joux and Vitse, to speed up the relation search in the hyperelliptic case. In [20], we give a new description of this technique, and show that this new formulation applies naturally to the non-hyperelliptic case with or without large prime variations. In particular, we obtain a speed-up by a factor approximately 3 for the relation search in Diem and Kochinke's methods.

SECRET Project-Team

6. New Results

6.1. Symmetric cryptology

Participants: Anne Canteaut, Pascale Charpin, Sébastien Duval, Virginie Lallemand, Gaëtan Leurent, Nicky Mouha, María Naya Plasencia, Joëlle Roué, Yann Rotella.

6.1.1. Block ciphers

Most of our work on block ciphers is related to an ANR Project named BLOC. Our recent results mainly concern either the analysis and design of lightweight block ciphers.

Recent results:

- Design and study of a new construction for low-latency block ciphers, named *reflection ciphers*, which generalizes the so-called α-reflection property exploited in PRINCE. This construction aims at reducing the implementation overhead of decryption on top of encryption [15], [60].
- Formalization and generic improvements of impossible differential cryptanalysis: our work provides a general framework for impossible differential cryptanalysis including a generic complexity analysis of the optimal attack [36].
- Cryptanalysis of several recently proposed block ciphers which offer an optimal resistance against side-channel attacks in the sense that the cost of Boolean masking is minimized. This includes an attack against Zorro and its variants [39], and an attack against Picaro in the related-key model [44].
- Cryptanalysis of Feistel constructions with secret Sboxes [42].
- Study of the security of the Even-Mansour construction in the multi-key setting [56].

6.1.2. Authenticated encryption

A limitation of all classical block ciphers is that they aim at protecting confidentiality only, while most applications need both encryption and authentication. These two functionalities are provided by using a block cipher like the AES together with an appropriate mode of operation. However, it appears that the most widely-used mode of operation for authenticated encryption, AES-GCM, is not very efficient for high-speed networks. Also, the security of the GCM mode completely collapses when an IV is reused. These severe drawbacks have then motivated an international competition named CAESAR, partly supported by the NIST, which has been recently launched in order to define some new authenticated encryption schemes [0]. Our work related to this competition is then two-fold: G. Leurent and N. Mouha have participated to the design of some CAESAR candidates; Also, the project-team is involved in a national cryptanalytic effort led by the BRUTUS project funded by the ANR.

Recent results:

- Design of new authenticated encryption schemes submitted to the CAESAR competition: SCREAM v3.0 [72] and PRIMATES 2[58]
- Cryptanalysis of the CAESAR candidates: collision attacks [49] against several candidates including AEZ and Marble, attack against LAC [53].

6.1.3. Stream ciphers

Stream ciphers provide an alternative to block-cipher-based encryption schemes. They are especially well-suited in applications which require either extremely fast encryption or a very low-cost hardware implementation.

[0] http://competitions.cr.yp.to/caesar.html

Recent results:

- Cryptanalysis of the recently proposed lightweight stream cipher Sprout [52], [71].
- New types of correlation attacks against filter generators exploiting the approximation of the filtering function composed with non-bijective monomial mappings [63], [87].
- Design of encryption schemes for efficient homomorphic-ciphertext compression: in order to avoid the (extremely) high expansion rate of homomorphic encryption, a solution consists in transmitting to the server the ciphertext c obtained by encrypting m with a symmetric scheme (the corresponding secret key encrypted by the homomorphic cipher is also transmitted). The server then needs to compute m encrypted with the homomorphic scheme from c, i.e. the server needs to homomorphically evaluate the decryption circuit of the symmetric cipher. A. Canteaut, M. Naya-Plasencia together with their coauthors have investigated the constraints on the symmetric cipher imposed by this application and they have proposed some solutions based on additive IV-based stream ciphers [78].

6.1.4. Hash functions and MACS

The international research effort related to the selection of the new hash function standard SHA-3 has led to many important results and to a better understanding of the security offered by hash functions. However, hash functions are used in a huge number of applications with different security requirements, and also form the building-blocks of some other primitives, like MACs. In this context, we have investigated the security of some of these constructions, in order to determine whether some particular constructions for hash functions may affect the security of the associated MACs.

Recent results:

- Improved generic attacks against hash-based MAC [30], [31]
- Cryptanalysis of 7 (out of 8) rounds of the Chaskey MAC [32]. This work has led the designers of Chaskey to increase the number of rounds [80].
- Attack against the XOR of two hash functions, using complex structures build from collisions [54]. This work by G. Leurent and L. Wang shows that, surprisingly, the construction $H_1(M) \oplus H_2(M)$ with common hash functions H_1 and H_2 (e.g. SHA-256 and BLAKE-256) is actually be less secure than each function on their own.

6.1.5. Security of Internet protocols

Hash functions are used to in key-exchange protocols such as TLS, IKE and SSH, to verify the integrity of the exchange. Most practitioners believe that the hash function only need to resist preimage attacks for this use. However, K. Bhargavan and G. Leurent have shown that collisions in the hash function are sufficient to break the integrity of these protocols, and to impersonate some of the parties [41]. Since many protocols still allow the use of MD5 or SHA-1 (for which collision attacks are known), this result in some practical attacks, and extends the real-world impact of the collision attacks against MD5 and SHA-1. This work has already influenced the latest TLS 1.3 draft, and the main TLS libraries are removing support of MD5 signatures

6.1.6. Cryptographic properties and construction of appropriate building blocks

The construction of building blocks which guarantee a high resistance against the known attacks is a major topic within our project-team, for stream ciphers, block ciphers and hash functions. The use of such optimal objects actually leads to some mathematical structures which may be at the origin of new attacks. This work involves fundamental aspects related to discrete mathematics, cryptanalysis and implementation aspects. Actually, characterizing the structures of the building blocks which are optimal regarding to some attacks is very important for finding appropriate constructions and also for determining whether the underlying structure induces some weaknesses or not. For these reasons, we have investigated several families of filtering functions and of S-boxes which are well-suited for their cryptographic properties or for their implementation characteristics.

Recent results:

- Definition of an extended criterion for estimating the resistance of a block cipher to differential attacks. This work emphasizes the role played by the affine permutation of the set of 8-bit words which follows the inverse function in the AES [45], [25], [26], [64], [24] (see Section 5.1.1).
- Construction of new Sboxes for lightweight ciphers: A. Canteaut, S. Duval and G. Leurent have investigated several constructions for obtaining good cryptographic Sboxes (especially 8-bit Sboxes) with a low implementation cost [43], [62], [84].
- P. Charpin, together with S. Mesnager and S. Sarkar, has provided a rigorous study of involutions over the finite field of order 2^n which are relevant primitives for cryptographic designs [47]. Most notably, they have focused on the class of involutions defined by Dickson polynomials [70], [79].

6.2. Code-based cryptography

Participants: Rodolfo Canto Torres, Julia Chaulet, Adrien Hauteville, Irene Márquez Corbella, Aurélie Phesso, Nicolas Sendrier, Jean-Pierre Tillich.

The first cryptosystem based on error-correcting codes was a public-key encryption scheme proposed by McEliece in 1978; a dual variant was proposed in 1986 by Niederreiter. We proposed the first (and only) digital signature scheme in 2001. Those systems enjoy very interesting features (fast encryption/decryption, short signature, good security reduction) but also have their drawbacks (large public key, encryption overhead, expensive signature generation). Some of the main issues in this field are

- security analysis, implementation and practicality of existing solutions,
- reducing the key size, *e.g.*, by using rank metric instead of Hamming metric, or by using particular families of codes,
- addressing new functionalities, like hashing or symmetric encryption.

Recent results:

- Structural attacks against some variants of the McEliece cryptosystem based on subclasses of alternant/Goppa codes which admit a very compact public matrix, typically quasi-cyclic, quasi-dyadic, or quasi-monoidic matrices [20]. This result is obtained thanks to a new operation on codes called folding that exploits the knowledge of the automorphism group of the code [19].
- Cryptanalysis of a variant of McEliece cryptosystem based on polar codes [40], [59].
- Cryptanalysis of a code-based encryption scheme proposed by Baldi *et al.* in the *Journal of Cryptology* [48].
- Cryptanalysis of a code-based signature scheme proposed at PQCrypto 2013 by Baldi at al. [57].
- Improved algorithm for decoding in the rank metric when some additional information about the targeted codeword is provided [51]; this algorithm used together with a folding technique leads to a feasible attack on the LRPC cryptosystem.
- Design on a new code-based stream cipher, named RankSynd, variant of Synd for the rank metric [50].
- In-depth analysis of the complexity of generic decoding algorithms for linear codes [37]. Most notably, R. Canto Torres and N. Sendrier have investigated the information-set decoding algorithms applied to the case where the number of errors is sub-linear in the code length [46]. This situation appears in the analysis of the McEliece based in quasi-cyclic Moderate Density Parity Check (MDPC) codes.

6.3. Quantum Information

Participants: Kaushik Chakraborty, André Chailloux, Anthony Leverrier, Jean-Pierre Tillich.

6.3.1. Quantum codes

Protecting quantum information from external noise is an issue of paramount importance for building a quantum computer. It also worthwhile to notice that all quantum error-correcting code schemes proposed up to now suffer from the very same problem that the first (classical) error-correcting codes had: there are constructions of good quantum codes, but for the best of them it is not known how to decode them in polynomial time.

Recent results:

- A. Leverrier and JP. Tillich, together with G. Zémor, proposed a new class of quantum LDPC codes, "Quantum expander codes", which feature a simple and very efficient decoding algorithm which can correct arbitrary patterns of errors of size scaling as the square-root of the length of the code. These are the first codes with constant rate for which such an efficient decoding algorithm is known (see Section 5.1.3) [55], [35], [73].
- Error analysis for Boson Sampling, a simplified model for quantum computation [21]

6.3.2. Quantum cryptography

A recent approach to cryptography takes into account that all interactions occur in a physical world described by the laws of quantum physics. These laws put severe constraints on what an adversary can achieve, and allow for instance to design provably secure key distribution protocols. We study such protocols as well as more general cryptographic primitives such as coin flipping with security properties based on quantum theory.

Recent results:

- A. Leverrier gave the first composable security proof for a continuous-variable quantum key distribution protocol with coherent states [22]. This essentially completes the security analysis of continuous-variable protocols with coherent states, which are by far the most practical protocols relying on continuous variables.
- A. Leverrier and E. Diamanti reviewed the state-of-the-art concerning quantum key distribution with continuous variables [18].
- A. Leverrier and M. Tomamichel gave the most complete security proof of the BB84 protocol to date, including all finite-size effects and a full description of the protocol [89].
- K. Chakraborty and A. Leverrier studied a general family of quantum protocols for position verification and present a new class of attacks based on the Clifford hierarchy that outperform previously known attacks [17].

6.3.3. Quantum correlations and nonlocality

Since the seminal work from Bell in the 60's, it has been known that classical correlations obtained via shared randomness cannot reproduce all the correlations obtained by measuring entangled quantum systems. This impossibility is for instance witnessed by the violation of a Bell inequality and is known under the name of "Quantum Nonlocality". In addition to its numerous applications for quantum cryptography, the study of quantum nonlocality and quantum games has become a central topic in quantum information theory, with the hope of bringing new insights to our understanding of quantum theory.

Recent results:

- Development of a general framework for the study of quantum correlations with combinatorial tools [14]

6.3.4. Relativistic cryptography

(see Section 5.1.2).

6.3.5. *Quantum cryptanalysis of symmetric primitives*

Symmetric cryptography seems at first sight much less affected in the post-quantum world than asymmetric cryptography: its main known threat is Grover's algorithm, which allows for an exhaustive key search in the square root of the normal complexity. For this reason, it is usually believed that doubling key lengths suffices to maintain an equivalent security in the post-quantum world. However, a lot of work is certainly required in the field of symmetric cryptography in order to "quantize" the classical families of attacks in an optimized way. G. Leurent, A. Leverrier and M. Naya Plasencia have recently started working in this area in collaboration with M. Kaplan, especially on differential cryptanalysis. Some preliminary results show that counter-intuitive and surprising cases appear: in general, it is not sufficient to consider the best classical attacks and try to "quantize" them if one wants to find the best post-quantum attack [34], [85].

6.4. Reverse-engineering of communication systems

Participants: Nicolas Sendrier, Jean-Pierre Tillich, Audrey Tixier.

Our activity within this domain, whose first aim is to establish the scientific and technical foundations of a discipline which does not exist yet at an academic level, has been supported by some industrial contracts driven by the Ministry of Defense.

Recent results:

- Efficient algorithm for recovering the block interleaver and the convolutional code when several noisy interleaver codewords are given [76], [13].

SPECFUN Project-Team

6. New Results

6.1. Integration of rational functions

Periods of rational integrals are specific integrals, with respect to one or several variables, whose integrand is a rational function and whose domain of integration is closed. This particular class of integrals contains large families of functions naturally occurring in combinatorics and statistical physics, such as diagonals, constant terms and positive part of rational functions. Periods involving one parameter are classically known to satisfy *Picard-Fuchs equations*, a special type of linear differential equations with a very rich analytic and arithmetic structure. As for other special-function manipulations, handling periods through those differential equations is a good way to actually compute them, and this was the topic of Pierre Lairez' PhD thesis defended in 2014 [53] and awarded the "Ecole Polytechnique thesis prize" in 2015.

Computing multivariate integrals is one speciality of the team and our algorithms are known to treat much more general integrals than just periods of rational integrals. However, integration is still slow in practice when the number of variables goes increasing. By looking at periods of rational functions, the hope is to obtain relevant complexity bounds and faster algorithms.

The goal of reaching relevant theoretical complexity bounds had been reached in 2013 [31] but a practically fast algorithm was still missing. This year, we described a new algorithm which is efficient in practice [4], though its complexity is not known. This algorithm allows to compute quickly integrals that are too big to be computed with previous algorithms. As a challenging benchmark, we computed 210 integrals given by Batyrev and Kreuzer in their work on Calabi–Yau varieties. This achievement gave strong visibility to the paper and allowed a quick dissemination of the implementation, which is provided in Magma under a CeCILL B license. The algorithm is now used on a regular basis by several teams. We know of:

- Tom Coates' team (Dpt. of Mathematics, Imperial College, London, UK), which uses the software in their work about mirror symmetry and classification of Fano varieties;
- Duco van Straten (Institute of Mathematics, University of Mainz, Germany), who uses the software in his work in algebraic geometry;
- Gert Alkmvist (Dpt. of Mathematics, University of Lund, Sweden), who uses the software in his work of enumerating the Calabi–Yau differential equations.

6.2. Multiple binomial sums

Multiple binomial sums form a large class of multi-indexed sequences, closed under partial summation, which contains most of the sequences obtained by multiple summation of binomial coefficients and also all the sequences with algebraic generating function. We study in [14] the representation of the generating functions of binomial sums by integrals of rational functions. The outcome is twofold. Firstly, we show that a univariate sequence is a multiple binomial sum if and only if its generating function is the diagonal of a rational function. Secondly we propose algorithms that decide the equality of multiple binomial sums and that compute recurrence relations for them. In conjunction with geometric simplifications of the integral representations, this approach behaves well in practice. The process avoids the computation of certificates and the problem of accurate summation that afflicts discrete creative telescoping, both in theory and in practice.

6.3. Diagonals of rational functions and selected differential Galois groups

Diagonals of rational functions naturally occur in lattice statistical mechanics and enumerative combinatorics. In all the examples emerging from physics, the minimal linear differential operators annihilating these diagonals of rational functions have been shown to actually possess orthogonal or symplectic differential Galois groups. In order to understand the emergence of such orthogonal or symplectic groups, we exhaustively analyze in [1] three (constrained) sets of diagonals of rational functions, corresponding respectively to rational functions of three variables, four variables and six variables. The conclusion is that, even for these sets of examples which, at first sight, have no relation with physics, their differential Galois groups are always orthogonal or symplectic groups. We also discuss conditions on the rational functions such that the operators annihilating their diagonals do not correspond to orthogonal or symplectic differential Galois groups, but rather to generic special linear groups.

6.4. Algebraic Diagonals and Walks

The diagonal of a multivariate power series F is the univariate power series $\mathsf{Diag}F$ generated by the diagonal terms of F. Diagonals form an important class of power series; they occur frequently in number theory, theoretical physics and enumerative combinatorics. In [7] we study algorithmic questions related to diagonals in the case where F is the Taylor expansion of a bivariate rational function. It is classical that in this case $\mathsf{Diag}F$ is an algebraic function. We propose an algorithm that computes an annihilating polynomial for $\mathsf{Diag}F$. Generically, it is its minimal polynomial and is obtained in time quasi-linear in its size. We show that this minimal polynomial has an exponential size with respect to the degree of the input rational function. We then address the related problem of enumerating directed lattice walks. The insight given by our study leads to a new method for expanding the generating power series of bridges, excursions and meanders. We show that their first N terms can be computed in quasi-linear complexity in N, without first computing a very large polynomial equation. An extended version of this work is presented in [13].

6.5. A human proof of the Gessel conjecture

Counting lattice paths obeying various geometric constraints is a classical topic in combinatorics and probability theory. Many recent works deal with the enumeration of 2-dimensional walks with prescribed steps confined to the positive quadrant. A notoriously difficult case concerns the so-called *Gessel walks*: they are planar walks confined to the positive quarter plane, that move by unit steps in any of the following directions: West, North-East, East and South-West. In 2001, Ira Gessel conjectured a closed-form expression for the number of such walks of a given length starting and ending at the origin. In 2008, Kauers, Koutschan and Zeilberger gave a computer-aided proof of this conjecture. The same year, Bostan and Kauers showed, using again computer algebra tools, that the trivariate generating function of Gessel walks is algebraic. We propose in [3] the first “human proofs” of these results. They are derived from a new expression for the generating function of Gessel walks in terms of special functions. This work has been published in the prestigious journal *Transactions of the AMS*.

6.6. Enumeration of 3-dimensional lattice walks confined to the positive octant

Small step walks in 2D are by now quite well understood, but almost everything remains to be done in higher dimensions. We explored in [2] the classification problem for 3-dimensional walks with unit steps confined to the positive octant. The first difficulty is their number: there are 11 074 225 cases (instead of 79 in dimension 2). In our work, we focused on the 35 548 that have at most six steps. We applied to them a combined approach, first experimental and then rigorous. Among the 35 548 cases, we first found 170 cases with a finite group; in the remaining cases, our experiments suggest that the group is infinite. We then rigorously proved D-finiteness of the generating series in all the 170 cases, with the exception of 19 intriguing step sets for which the nature of the generating function still remains unclear. In two challenging cases, no human proof is currently known, and we derived computer-algebra proofs, thus constituting the first proofs for those two step sets.

6.7. Efficient algorithms for rational first integrals

We presented in [29] fast algorithms for computing rational first integrals with degree bounded by N of a planar polynomial vector field of degree $d \leq N$. The main novelty is that such rational first integrals are obtained by computing via systems of linear equations instead of systems of quadratic equations. This leads to a probabilistic algorithm with arithmetic complexity $\tilde{\mathrm{O}}(N^{2\omega})$ and to a deterministic algorithm for solving the problem in $\tilde{\mathrm{O}}(d^2N^{2\omega+1})$ arithmetic operations, where ω is the exponent of linear algebra. By comparison, the best previous algorithm uses at least $d^{\omega+1}N^{4\omega+4}$ arithmetic operations. Our new algorithms are moreover very efficient in practice.

6.8. Quasi-optimal computation of the p-curvature

The p-curvature of a system of linear differential equations in positive characteristic p is a matrix that measures to what extent the system is close to having a fundamental matrix of rational function solutions. This notion, originally introduced in the arithmetic theory of differential equations, has been recently used as an effective tool in computer algebra and in combinatorial applications. We have described in [6] a recent algorithm for computing the p-curvature, whose complexity is almost optimal with respect to the size of the output. The new algorithm performs remarkably well in practice. Its design relies on the existence of a well-suited ring, of so-called Hurwitz series, for which an analogue of the Cauchy–Lipschitz Theorem holds, and on a FFT-like method in which the “evaluation points” are Hurwitz series.

6.9. Axiomatic constraint systems for proof search modulo theories

Goal-directed proof search in first-order logic uses meta-variables to delay the choice of witnesses; substitutions for such variables are produced when closing proof-tree branches, using first-order unification or a theory-specific background reasoner. We have investigated a generalization of such mechanisms whereby theory-specific constraints are produced instead of substitutions. In order to design modular proof-search procedures over such mechanisms, we provide a sequent calculus with meta-variables, which manipulates such constraints abstractly. Proving soundness and completeness of the calculus leads to an acclimatization that identifies the conditions under which abstract constraints can be generated and propagated in the same way unifiers usually are. We then extract from our abstract framework a component interface and a specification for concrete implementations of background reasoners. This is a common work with Damien Rouhling (ENS Lyon), Stéphane Lengrand (CNRS, LIX) and Jean-Marc Notin (CNRS, LIX), based on the PhD contributions of Mahfuza Farooque (unaffiliated). It is described in [8].

6.10. DynaMoW: Dynamic Mathematics on the Web

The interactivity needed by our on-line encyclopedia DDMF is made possible by implementing it over our tool DynaMoW (http://ddmf.msr-inria.inria.fr/DynaMoW/). This Ocaml library simultaneously controls external symbolic calculations and web-page generation and was first developed from 2008 to 2011. With the evolution of Ocaml and web technologies, it became possible to hope for a more reactive and configurable tool, by using light-weight threads and websockets. A new design was elaborated this year by F. Chyzak and M. Guesdon, and DynaMoW was rewritten by the latter. Using this new DynaMoW will require a complete and potentially time-consuming port of DDMF. So we decided that experimenting with the port of a smaller DynaMoW-based application should be done to ascertain the new design of DynaMoW-based before going to scale with DDMF. To this end, we applied DynaMoW to another on-line encyclopedia of our’s, ECS. The code is now stabilizing, and will be released next year, after documentation is written.

6.11. ECS: Encyclopedia of Combinatorial Structures

The Encyclopedia of Combinatorial Structures (ECS, http://algo.inria.fr/encyclopedia/) originates as a project in Project-Team Algorithms, with a first release back in 1998. It is an on-line mathematical encyclopedia with an emphasis on sequences that arise in the context of decomposable combinatorial structures, with the possibility to search by the first terms in the sequence, keyword, generating function, or closed form. As such, ECS ambitions to be seen as a young cousin of Sloane's famous Encyclopedia of Integer Sequences http://www.research.att.com/articles/featured_stories/2012_03/201203_OEIS.html?fbid=cibE46xiHwx. The latter lists more general types of sequences, and points to numerous entries in ECS for specific properties. With regard to our software development, ECS has served as a nice testbed for several evolutions of DynaMoW, in particular in 2009 and 2011. This year, F. Chyzak and M. Guesdon ported ECS to the language of the new DynaMoW. Public release is expected soon in 2016, and will please the many users waiting for this new release after the former website was discontinued for technical reasons.

6.12. Mathematical Components Library

We have released a new version of the Mathematical Components Library (http://www.msr-inria.fr/projects/mathematical-components-2/), including an updated version of the Ssreflect package (http://ssr.msr-inria.inria.fr/). A major refactoring of the archive now allows a more modular distribution, through several thematic packages, also available via the OPAM package manager. We have also opened our development repository and we mirror it on the GitHub platform, in order to better foster the community of users of the library.

VEGAS Project-Team

6. New Results

6.1. Robustness issues in computational geometry

Participants: Olivier Devillers, Monique Teillaud.

6.1.1. Qualitative Symbolic Perturbation: a new geometry-based perturbation framework

In a classical Symbolic Perturbation scheme, degeneracies are handled by substituting some polynomials in ϵ to the input of a predicate. Instead of a single perturbation, we propose to use a sequence of (simpler) perturbations. Moreover, we look at their effects geometrically instead of algebraically; this allows us to tackle cases that were not tractable with the classical algebraic approach [25].

This work was done in collaboration with Menelaos Karavelas (Univ. of Crete).

6.2. Probabilistic analysis of geometric data structures and algorithms

Participant: Olivier Devillers.

6.2.1. The worst visibility walk in a random Delaunay triangulation is $O(\sqrt{n})$

We show that the memoryless routing algorithms Greedy Walk, Compass Walk, and all variants of visibility walk based on orientation predicates are asymptotically optimal in the average case on the Delaunay triangulation. More specifically, we consider the Delaunay triangulation of an unbounded Poisson point process of unit rate and demonstrate that the worst-case path between any two vertices inside a domain of area n has a number of steps that is not asymptotically more than the shortest path which exists between those two vertices with probability converging to one (as long as the vertices are sufficiently far apart.) As a corollary, it follows that the worst-case path has $O(\sqrt{n})$ steps in the limiting case, under the same conditions. Our results have applications in routing in mobile networks and also settle a long-standing conjecture in point location using walking algorithms. Our proofs use techniques from percolation theory and stochastic geometry [24].

This work was done in collaboration with Ross Hemsley (formerly in Inria Geometrica).

6.2.2. Smooth analysis of convex hulls

We establish an upper bound on the smoothed complexity of convex hulls in $\mathbb{R}^d$ under uniform Euclidean (ℓ^2) noise. Specifically, let $\{p_1^*, p_2^*, ..., p_n^*\}$ be an arbitrary set of n points in the unit ball in $\mathbb{R}^d$ and let $p_i = p_i^* + x_i$, where $x_1, x_2, ..., x_n$ are chosen independently from the unit ball of radius δ. We show that the expected complexity, measured as the number of faces of all dimensions, of the convex hull of $\{p_1, p_2, ..., p_n\}$ is $O\left(n^{2-\frac{4}{d+1}}\left(1+1/\delta\right)^{d-1}\right)$; the magnitude δ of the noise may vary with n. For $d=2$ this bound improves to $O\left(n^{\frac{2}{3}}(1+\delta^{-\frac{2}{3}})\right)$.

We also analyze the expected complexity of the convex hull of ℓ^2 and Gaussian perturbations of a nice sample of a sphere, giving a lower-bound for the smoothed complexity. We identify the different regimes in terms of the scale, as a function of n, and show that as the magnitude of the noise increases, that complexity varies monotonically for Gaussian noise but non-monotonically for ℓ^2 noise [13].

This work was done in collaboration with Xavier Goaoc (Univ. Marne la Vallée), Marc Glisse and Remy Thomasse (Inria Geometrica).

6.3. Non-linear computational geometry

Participants: Guillaume Moroz, Sylvain Lazard, Marc Pouget, Laurent Dupont, Rémi Imbach.

6.3.1. Solving bivariate systems and topology of plane algebraic curves

In the context of our algorithm Isotop for computing the topology of plane algebraic curves (see Section 5.1), we work on the problem of solving a system of two bivariate polynomials. We are interested in certified numerical approximations or, more precisely, isolating boxes of the solutions. But we are also interested in computing, as intermediate symbolic objects, a Rational Univariate Representation (RUR) that is, roughly speaking, a univariate polynomial and two rational functions that map the roots of the univariate polynomial to the two coordinates of the solutions of the system. RURs are relevant symbolic objects because they allow to turn many queries on the system into queries on univariate polynomials. However, such representations require the computation of a separating form for the system, that is a linear combination of the variables that takes different values when evaluated at the distinct solutions of the system.

We published this year [11] results showing that, given two polynomials of degree at most d with integer coefficients of bitsize at most τ, (i) a separating form, (ii) the associated RUR, and (iii) isolating boxes of the solutions can be computed in, respectively, $\widetilde{O}_B(d^8+d^7\tau)$, $\widetilde{O}_B(d^7+d^6\tau)$ and $\widetilde{O}_B(d^8+d^7\tau)$ bit operations in the worst case, where $\widetilde{O}$ refers to the complexity where polylogarithmic factors are omitted and O_B refers to the bit complexity.

However, during the publishing process, we have substentially improved these results. We have presented for these three sub-problems new algorithms that have worst-case bit complexity $\widetilde{O}_B(d^6+d^5\tau)$. We have also presented probabilistic Las Vegas variants of our two first algorithms, which have expected bit complexity $\widetilde{O}_B(d^5+d^4\tau)$. We also show that it is likely difficult to improve these complexities as it would essentially require to improve bounds on other fundamental problems (e.g., computing resultants, checking squarefreeness and root isolation of univariate polynomials) that have hold for decades.

This work was done in collaboration with Yacine Bouzidi (Inria Saclay), Michael Sagraloff (MPII Sarrebruken, Germany) and Fabrice Rouillier (Inria Rocquencourt). It is published in the research report [22] and submitted to a journal.

A key ingredient of the above work is the classical triangular decomposition algorithm via subresultants [31] on which we obtain two results of independent interest. First, we improved by a factor d the state-of-the-art worst-case bit complexity of this algorithm [22]. One constraint on this algorithm is that it requires that the curves defined by the input polynomials have no common vertical asymptotes. Our second result is a generalization of this algorithm, which removes that restriction while preserving the same worst-case bit complexity of $\widetilde{O}_B(d^6+d^5\tau)$. Furthermore, we actually present a refined bit complexity in $\widetilde{O}_B(d_x^3d_y^3+(d_x^2d_y^3+d_xd_y^4)\tau)$ where d_x and d_y bound the degrees of the input polynomials in x and y, respectively. We also prove that the total bitsize of the decomposition is in $\widetilde{O}((d_x^2d_y^3+d_xd_y^4)\tau)$.

This work was done in collaboration with Fabrice Rouillier (Inria Rocquencourt). It is published in the research report [27] and submitted to a journal.

6.3.2. Numeric and Certified Isolation of the Singularities of the Projection of a Smooth Space Curve

Let a smooth real analytic curve embedded in $\mathbb{R}^3$ be defined as the solution of real analytic equations of the form $P(x,y,z)=Q(x,y,z)=0$ or $P(x,y,z)=\frac{\partial P}{\partial z}=0$. Our main objective is to describe its projection $\mathcal{C}$ onto the (x,y)-plane. In general, the curve $\mathcal{C}$ is not a regular submanifold of $\mathbb{R}^2$ and describing it requires to isolate the points of its singularity locus Σ. After describing the types of singularities that can arise under some assumptions on P and Q, we present a new method to isolate the points of Σ. We experimented our method on pairs of independent random polynomials (P,Q) and on pairs of random polynomials of the form $(P,\frac{\partial P}{\partial z})$ and got promising results [14].

On the same topic but with a different approach, we improved our research report [26] by including experimental data using SubdivisionSolver (see Section 5.2) and submitted this work to a journal.

6.3.3. Mechanical design of parallel robots

In collaboration with F. Rouillier, D. Chablat and our PhD student Ranjan Jha, we analyzed the singularities and the workspace of different families of robots.

The first result is a certified description of the workspace and the singularities of a Delta like family robot [16]. Workspace and joint space analysis are essential steps in describing the task and designing the control loop of the robot, respectively. This paper presents the descriptive analysis of a family of delta-like parallel robots by using algebraic tools to induce an estimation about the complexity in representing the singularities in the workspace and the joint space. A Gröbner based elimination is used to compute the singularities of the manipulator and a Cylindrical Algebraic Decomposition algorithm is used to study the workspace and the joint space. From these algebraic objects, we propose some certified three dimensional plotting describing the shape of workspace and of the joint space which will help the engineers or researchers to decide the most suited configuration of the manipulator they should use for a given task. Also, the different parameters associated with the complexity of the serial and parallel singularities are tabulated, which further enhance the selection of the different configurations of the manipulator by comparing the complexity of the singularity equations.

The second result is an algebraic method to check the singularity-free paths for parallel robots [15]. Trajectory planning is a critical step while programming the parallel manipulators in a robotic cell. The main problem arises when there exists a singular configuration between the two poses of the end-effectors while discretizing the path with a classical approach. This paper presents an algebraic method to check the feasibility of any given trajectories in the workspace. The solutions of the polynomial equations associated with the trajectories are projected in the joint space using Gröbner based elimination methods and the remaining equations are expressed in a parametric form where the articular variables are functions of time t unlike any numerical or discretization method. These formal computations allow to write the Jacobian of the manipulator as a function of time and to check if its determinant can vanish between two poses. Another benefit of this approach is to use a largest workspace with a more complex shape than a cube, cylinder or sphere. For the Orthoglide, a three degrees of freedom parallel robot, three different trajectories are used to illustrate this method.

6.3.4. Reflection through quadric mirror surfaces

We addressed the problem of finding the reflection point on quadric mirror surfaces, especially ellipsoid, paraboloid or hyperboloid of two sheets, of a light ray emanating from a 3D point source P_1 and going through another 3D point P_2, the camera center of projection. We previously proposed a new algorithm for this problem, using a characterization of the reflection point as the tangential intersection point between the mirror and an ellipsoid with foci P_1 and P_2. The computation of this tangential intersection point is based on our algorithm for the computation of the intersection of quadrics [5], [28]. Unfortunately, our new algorithm is not yet efficient in practice. This year, we made several improvements on this algorithm. First, we decreased from 11 to 4 the degree of a critical polynomial that we need to solve and whose solutions induce the coefficients in some other polynomials appearing later in the computations. Second, we implemented Descarte's algorithm for isolating the real roots of univariate polynomials in the case where the coefficients belong to extensions of $\mathbb{Q}$ generated by at most two square roots. Furthermore, we are currently implementing the generalization of that algorithm when the coefficients belong to extensions of $\mathbb{Q}$ generated by one root of an arbitrary polynomial. These undergoing improvements should allow us to compute more directly the wanted reflexion point, thus avoiding problematic approximations and making the overall algorithm tractable.

ALF Project-Team

7. New Results

7.1. Processor Architecture

Participants: Pierre Michaud, Bharath Narasimha Swamy, Sylvain Collange, Erven Rohou, André Seznec, Arthur Perais, Surya Khizakanchery Natarajan, Sajith Kalathingal, Tao Sun, Andrea Mondelli, Aswinkumar Sridharan, Biswabandan Panda, Fernando Endo.

Processor, cache, locality, memory hierarchy, branch prediction, multicore, power, temperature

Multicore processors have now become mainstream for both general-purpose and embedded computing. Instead of working on improving the architecture of the next generation multicore, with the DAL project, we deliberately anticipate the next few generations of multicores. While multicores featuring 1000s of cores might become feasible around 2020, there are strong indications that sequential programming style will continue to be dominant. Even future mainstream parallel applications will exhibit large sequential sections. Amdahl's law indicates that high performance on these sequential sections is needed to enable overall high performance on the whole application. On many (most) applications, the effective performance of future computer systems using a 1000-core processor chip will significantly depend on their performance on both sequential code sections and single threads.

We envision that, around 2020, the processor chips will feature a few complex cores and many (maybe 1000's) simpler, more silicon and power effective cores.

In the DAL research project, https://team.inria.fr/alf/members/andre-seznec/defying-amdahls-law-dal/, we explore the microarchitecture techniques that will be needed to enable high performance on such heterogeneous processor chips. Very high performance will be required on both sequential sections, -legacy sequential codes, sequential sections of parallel applications-, and critical threads on parallel applications, -e.g. the main thread controlling the application. Our research focuses essentially on enhancing single process performance.

7.1.1. Microarchitecture

7.1.1.1. Branch prediction

Participant: André Seznec.

This research was done in collaboration with Joshua San Miguel and Jorge Albericio from University of Toronto

The most efficient branch predictors proposed in academic literature exploit both global branch history and local branch history. However, local history branch predictor components introduce major design challenges, particularly for the management of speculative histories. Therefore, most effective hardware designs use only global history components and very limited forms of local histories such as a loop predictor. The wormhole (WH) branch predictor was recently introduced to exploit branch outcome correlation in multidimensional loops. For some branches encapsulated in a multidimensional loop, their outcomes are correlated with those of the same branch in neighbor iterations, but in the previous outer loop iteration. Unfortunately, the practical implementation of the WH predictor is even more challenging than the implementation of local history predictors.

In [36], we introduce practical predictor components to exploit this branch outcome correlation in multidimensional loops: the IMLI-based predictor components. The iteration index of the inner most loop in an application can be efficiently monitored at instruction fetch time using the Inner Most Loop Iteration (IMLI) counter. The outcomes of some branches are strongly correlated with the value of this IMLI counter. A single PC+IMLI counter indexed table, the IMLI-SIC table, added to a neural component of any recent predictor (TAGE-based or perceptron-inspired) captures this correlation. Moreover, using the IMLI counter, one can efficiently manage the very long local histories of branches that are targeted by the WH predictor. A second IMLI-based component, IMLI-OH, allows for tracking the same set of hard-to-predict branches as WH. Managing the speculative states of the IMLI-based predictor components is quite simple. Our experiments show that augmenting a state-of-the-art global history predictor with IMLI components outperforms previous state-of-the-art academic predictors leveraging local and global history at much lower hardware complexity (i.e., smaller storage budget , smaller number of tables and simpler management of speculative states).

7.1.1.2. Revisiting Value Prediction

Participants: Arthur Perais, André Seznec.

Value prediction was proposed in the mid 90's to enhance the performance of high-end microprocessors. The research on Value Prediction techniques almost vanished in the early 2000's as it was more effective to increase the number of cores than to dedicate some silicon area to Value Prediction. However high end processor chips currently feature 8-16 high-end cores and the technology will allow to implement 50-100 of such cores on a single die in a foreseeable future. Amdahl's law suggests that the performance of most workloads will not scale to that level. Therefore, dedicating more silicon area to value prediction in high-end cores might be considered as worthwhile for future multicores.

At a first step, we showed that all predictors are amenable to very high accuracy at the cost of some loss on prediction coverage [7]. This greatly diminishes the number of value mispredictions and allows to delay validation until commit-time. As such, no complexity is added in the out-of-order engine because of VP (save for ports on the register file) and pipeline squashing at commit-time can be used to recover.

This allows to leverage the possibility of validating predictions at commit to introduce a new microarchitecture, EOLE [19]. EOLE features *Early Execution* to execute simple instructions whose operands are ready in parallel with Rename and *Late Execution* to execute simple predicted instructions and high confidence branches just before Commit. EOLE depends on Value Prediction to provide operands for *Early Execution* and predicted instructions for *Late Execution*. However, Value Prediction requires EOLE to become truly practical. That is, EOLE allows to reduce the out-of-order issue-width by 33% without impeding performance. As such, the number of ports on the register file diminishes. Furthermore, optimizations of the register file such as *banking* further reduce the number of required ports. Overall EOLE possesses a register file whose complexity is on-par with that of a regular wider-issue superscalar while the out-of-order components (scheduler, bypass) are greatly simplified. Moreover, thanks to Value Prediction, speedup is obtained on many benchmarks of the SPEC'00/'06 suite.

However complexity in the value predictor infrastructure itself is also problematic. First, multiple predictions must be generated each cycle, but multi-ported structures should be avoided. Second, the predictor should be small enough to be considered for implementation, yet coverage must remain high enough to increase performance. In [32], to address these remaining concerns, we first propose a block-based value prediction scheme mimicking current instruction fetch mechanisms, BeBoP. It associates the predicted values with a fetch block rather than distinct instructions. Second, to remedy the storage issue, we present the Differential VTAGE predictor. This new tightly coupled hybrid predictor covers instructions predictable by both VTAGE and Stride-based value predictors, and its hardware cost and complexity can be made similar to those of a modern branch predictor. Third, we show that block-based value prediction allows to implement the checkpointing mechanism needed to provide D-VTAGE with last computed/predicted values at moderate cost. Overall, we establish that EOLE with a 32.8KB block-based D-VTAGE predictor and a 4-issue OoO engine can significantly outperform a baseline 6-issue superscalar processor, by up to 62.2 % and 11.2 % on average (gmean), on our benchmark set.

The overall study on value prediction is presented in Arthur Perais's PhD [14].

7.1.1.3. Cost-Effective Speculative Scheduling in High Performance Processors

Participants: André Seznec, Arthur Perais, Pierre Michaud.

This study was done in collaboration with Andreas Sembrant and Erik Hagersten from Upsala University

To maximize performance, out-of-order execution processors sometimes issue instructions without having the guarantee that operands will be available in time; e.g. loads are typically assumed to hit in the L1 cache and dependent instructions are issued assuming a L1 hit. This form of speculation ?that we refer to as speculative scheduling? has been used for two decades in real processors, but has received little attention from the research community. In particular, as pipeline depth grows and the distance between the Issue and the Execute stages increases, it becomes critical to issue dependents on variable-latency instructions as soon as possible, rather than to wait for the actual cycle at which the result becomes available. Unfortunately, due to the uncertain nature of speculative scheduling, the scheduler may wrongly issue an instruction that will not have its source(s) on the bypass network when it reaches the Execute stage. Therefore, this instruction must be canceled and replayed, which can potentially impair performance and increase energy consumption.

In [31] we focus on ways to reduce the number of replays that are agnostic of the replay scheme. First, we propose an easily implementable, low-cost solution to reduce the number of replays caused by L1 bank conflicts. Schedule Shifting always assumes that, given a dual-load issue capacity, the second load issued in a given cycle will be delayed because of a bank conflict. Its dependents are thus always issued with a corresponding delay. Second, we also improve on existing L1 hit/miss prediction schemes by taking into account instruction criticality. That is, for some criterion of criticality and for loads whose hit/miss behavior is hard to predict, we show that it is more cost-effective to stall dependents if the load is not predicted critical. In total, in our experiments assuming a 4-cycle issue-to-execute delay, we found that the vast majority of instructions replays due to L1 data cache banks conflicts and L1 hit mispredictions can be avoided, thus leading to a 3.4% performance gain and a 13.4% decrease in the number of issued instructions, over a baseline speculative scheduling scheme.

7.1.1.4. Criticality-aware Resource Allocation in OOO Processors

Participants: André Seznec, Arthur Perais, Pierre Michaud.

This study was done in collaboration with Andreas Sembrant, Erik Hagersten, David Black-Schaffer and Trevor Carlson from Upsala University.

Modern processors employ large structures (IQ, LSQ, register file, etc.) to expose instruction-level parallelism (ILP) and memory-level parallelism (MLP). These resources are typically allocated to instructions in program order. This wastes resources by allocating resources to instructions that are not yet ready to be executed and by eagerly allocating resources to instructions that are not part of the application's critical path. In [35], we explore the possibility of allocating pipeline resources only when needed to expose MLP, and thereby enabling a processor design with significantly smaller structures, without sacrificing performance. First we identify the classes of instructions that should not reserve resources in program order and evaluate the potential performance gains we could achieve by delaying their allocations. We then use this information to "park" such instructions in a simpler, and therefore more efficient, Long Term Parking (LTP) structure. The LTP stores instructions until they are ready to execute, without allocating pipeline resources, and thereby keeps the pipeline available for instructions that can generate further MLP. LTP can accurately and rapidly identify which instructions to park, park them before they execute, wake them when needed to preserve performance, and do so using a simple queue instead of a complex IQ. We show that even a very simple queue-based LTP design allows us to significantly reduce IQ (64 → 32) and register file (128 → 96) sizes while retaining MLP performance and improving energy efficiency.

7.1.1.5. Efficient Execution on Guarded Instruction Sets

Participant: André Seznec.

ARM ISA based processors are no longer low complexity processors. Nowadays, ARM ISA based processor manufacturers are struggling to implement medium-end to high-end processor cores which implies implementing a state-of-the-art out-of-order execution engine. Unfortunately providing efficient out-of-order execution on legacy ARM codes may be quite challenging due to guarded instructions.

Predicting the guarded instructions addresses the main serialization impact associated with guarded instructions execution and the multiple definition problem. Moreover, guard prediction allows to use a global branch-and-guard history predictor to predict both branches and guards, often improving branch prediction accuracy. Unfortunately such a global branch-and-guard history predictor requires the systematic use of guard predictions. In that case, poor guard prediction accuracy would lead to poor overall performance on some applications.

Building on top of recent advances in branch prediction and confidence estimation, we propose a hybrid branch and guard predictor, combining a global branch history component and global branch-and-guard history component. The potential gain or loss due to the systematic use of guard prediction is dynamically evaluated at run-time. Two computing modes are enabled: systematic guard prediction and high confidence only guard prediction. Our experiments show that on most applications, an overwhelming majority of guarded instructions are predicted. Therefore a relatively inefficient but simple hardware solution can be used to execute the few unpredicted guarded instructions. Significant performance benefits are observed on most applications while applications with poorly predictable guards do not suffer from performance loss [8].

This study was accepted to ACM Transactions on Architecture and Compiler Optimizations (Dec. 2014) and presented at the HIPEAC conference in January 2015.

7.1.1.6. Clustered microarchitecture

Participants: Andrea Mondelli, Pierre Michaud, André Seznec.

In the last 10 years, the clock frequency of high-end superscalar processors did not increase significantly. Performance keeps being increased mainly by integrating more cores on the same chip and by introducing new instruction set extensions. However, this benefits only to some applications and requires rewriting and/or recompiling these applications. A more general way to increase performance is to increase the IPC, the number of instructions executed per cycle.

In [18], we argue that some of the benefits of technology scaling should be used to increase the IPC of future superscalar cores. Starting from microarchitecture parameters similar to recent commercial high-end cores, we show that an effective way to increase the IPC is to increase the issue width. But this must be done without impacting the clock cycle. We propose to combine two known techniques: clustering and register write specialization. The objective of past work on clustered microarchitecture was to allow a higher clock frequency while minimizing the IPC loss. This led researchers to consider narrow-issue clusters. Our objective, instead, is to increase the IPC without impacting the clock cycle, which means wide-issue clusters. We show that, on a wide-issue dual cluster, a very simple steering policy that sends 64 consecutive instructions to the same cluster, the next 64 instructions to the other cluster, and so on, permits tolerating an inter-cluster delay of several cycles. We also propose a method for decreasing the energy cost of sending results of one cluster to the other cluster.

7.1.1.7. Adaptive Intelligent Memory Systems

Participants: André Seznec, Aswinkumar Sridharan.

Multi-core processors employ shared Last Level Caches (LLC). This trend will continue in the future with large multi-core processors (16 cores and beyond) as well. At the same time, the associativity of this LLC tends to remain in the order of sixteen. Consequently, with large multicore processors, the number of cores that share the LLC becomes larger than the associativity of the cache itself. LLC management policies have been extensively studied for small scale multi-cores (4 to 8 cores) and associativity degree in the 16 range. However, the impact of LLC management on large multi-cores is essentially unknown, in particular when the associativity degree is smaller than the number of cores.

In [43], we introduce Adaptive Discrete and deprioritized Application PrioriTization (ADAPT), an LLC management policy addressing the large multi-cores where the LLC associativity degree is smaller than the number of cores. ADAPT builds on the use of the Footprint-number metric. Footprint-number is defined as the number of unique accesses (block addresses) that an application generates to a cache set in an interval of time. We propose a monitoring mechanism that dynamically samples cache sets to estimate the Footprint-number of applications and classifies them into discrete (distinct and more than two) priority buckets. The cache replacement policy leverages this classification and assigns priorities to cache lines of applications during cache replacement operations. Footprint-number is computed periodically to account the dynamic changes in applications behavior. We further find that de- prioritizing certain applications during cache replacement is beneficial to the overall performance. We evaluate our proposal on 16, 20 and 24-core multi-programmed workloads and discuss other aspects in detail.

[43] has been accepted for publication at the IPDPS 2016 conference.

7.1.1.8. Hardware data prefetching

Participant: Pierre Michaud.

Hardware prefetching is an important feature of modern high-performance processors. When an application's working set is too large to fit in on-chip caches, disabling hardware prefetchers may result in severe performance reduction. We propose a new hardware data prefetcher, the Best-Offset (BO) prefetcher. The BO prefetcher is an offset prefetcher using a new method for selecting the best prefetch offset taking into account prefetch timeliness. The hardware required for implementing the BO prefetcher is very simple. The BO prefetcher won the last Data Prefetching Championship [27].

A paper describing and studying the BO prefetcher has been accepted for publication at the HPCA 2016 conference.

7.1.1.9. Prediction-based superpage-friendly TLB designs

Participant: André Seznec.

This research was done in collaboration with Misel-Myrto Papadopoulou, Xin Tong and Andreas Moshovos from University of Toronto

In [30], we demonstrate that a set of commercial and scale-out applications exhibit significant use of superpages and thus suffer from the fixed and small superpage TLB structures of some modern core designs. Other processors better cope with superpages at the expense of using power-hungry and slow fully-associative TLBs. We consider alternate designs that allow all pages to freely share a single, power-efficient and fast set-associative TLB. We propose a prediction-guided multi-grain TLB design that uses a superpage prediction mechanism to avoid multiple lookups in the common case. In addition, we evaluate the previously proposed skewed TLB which builds on principles similar to those used in skewed associative caches . We enhance the original skewed TLB design by using page size prediction to increase its effective associativity. Our prediction-based multi-grain TLB design delivers more hits and is more power efficient than existing alternatives. The predictor uses a 32-byte prediction table indexed by base register values.

7.1.2. Microarchitecture Performance Modeling

7.1.2.1. Symbiotic scheduling on SMT cores and symmetric multicores

Participant: Pierre Michaud.

This research was done in collaboration with Stijn Eyerman and Wouter Rogiest from Ghent University.

When several independent tasks execute concurrently on a simultaneous multithreaded (SMT) core or on a multicore, they share hardware resources. Hence the execution rate of a task is influenced by the other tasks running at the same time. Based on this observation, Snavely and Tullsen proposed *symbiotic* scheduling, i.e., the idea that performance can be increased by co-scheduling tasks that do not stress the same shared resources [63]. They claim that, when the number of concurrent tasks exceeds the number of logical cores, symbiotic scheduling increases performance substantially. A more recent study by Eyerman and Eeckhout reached similar conclusions [54].

We have revisited symbiotic scheduling for SMT cores and symmetric multicores [22], and we obtained very modest throughput gains, which seemingly contradicts the above mentioned studies. We analyzed the reasons for this discrepancy and found that previous studies did not measure throughput but average response time. Response time reductions can be magnified by setting the job arrival rate very close to the maximum throughput, which turns a tiny throughput increase into a large response time reduction. Also, the proposed scheduling policies are approximately equivalent to scheduling the shortest jobs first, which mechanically reduces the average response time independently of any symbiosis effect.

We identified three typical situations where symbiotic scheduling yields little to no throughput gain: (1) most of the time is spent executing a single type of job, or (2) jobs' execution rates barely depend on which other jobs are running concurrently, or (3) jobs' execution rates are proportional to the fraction they get of a certain shared resource (e.g., instruction decode bandwidth in an SMT core). In our experiments, most workloads were close to one of the three situations above.

7.1.2.2. Modeling multi-threaded programs execution time in the many-core era

Participants: Surya Khizakanchery Natarajan, Bharath Narasimha Swamy, André Seznec.

Estimating the potential performance of parallel applications on the yet-to-be-designed future many cores is very speculative. The simple models proposed by Amdahl's law (fixed input problem size) or Gustafson's law (fixed number of cores) do not completely capture the scaling behaviour of a multi-threaded (MT) application leading to over estimation of performance in the many-core era. On the other hand, modeling many-core by simulation is too slow to study the applications performance. In [28], [13], we propose a more refined but still tractable, high level empirical performance model for multi-threaded applications, the Serial/Parallel Scaling (SPS) Model to study the scalability and performance of application in many-core era. SPS model learns the application behavior on a given architecture and provides realistic estimates of the performance in future many-cores. Considering both input problem size and the number of cores in modeling, SPS model can help in making high level decisions on the design choice of future many-core applications and architecture. We validate the model on the Many-Integrated Cores (MIC) xeon-phi with 240 logical cores.

7.1.2.3. Optimal cache replacement

Participant: Pierre Michaud.

This research was done in collaboration with Mun-Kyu Lee, Jeong Seop Sim and DaeHun Nyang from Inha University.

The replacement policy for a cache is the algorithm, implemented in hardware, selecting a block to evict for making room for an incoming block. This research topic has been revitalized in recent years. The MIN replacement policy, which evicts the block referenced furthest in the future, was introduced by Belady [51] and was later shown to be optimal by Mattson et al. [60]. The MIN policy is an offline policy that cannot be implemented in real processors, as it needs the knowledge of future memory accesses. Still, a possible way to improve online replacement policies would be to emulate the MIN policy, trying to use past references to predict future ones. However, the MIN policy is not intuitive, and Mattson et al.'s proof of optimality is quite involved. We believe that new intuition about the MIN policy will help microarchitects improve cache replacement policies. As a first step toward this goal, we produced a new, intuitive proof of optimality of the MIN policy [17].

7.1.3. Hardware/Software Approaches

7.1.3.1. Helper threads

Participants: Bharath Narasimha Swamy, André Seznec.

Heterogeneous Many Cores (HMC) architectures that mix many simple/small cores with a few complex/large cores are emerging as a design alternative that can provide both fast sequential performance for single threaded workloads and power-efficient execution for throughput oriented parallel workloads. The availability of many small cores in a HMC presents an opportunity to utilize them as low-power helper cores to accelerate memory-intensive sequential programs mapped to a large core. However, the latency overhead of accessing small cores in a loosely coupled system limits their utility as helper cores. Also, it is not clear if small cores can execute helper threads sufficiently in advance to benefit applications running on a larger, much powerful, core.

In [12] we present a hardware/software framework called core-tethering to support efficient helper threading on heterogeneous many-cores. Core-tethering provides a co-processor like interface to the small cores that (a) enables a large core to directly initiate and control helper execution on the helper core and (b) allows efficient transfer of execution context between the cores, thereby reducing the performance overhead of accessing small cores for helper execution. Our evaluation on a set of memory intensive programs chosen from the standard benchmark suites show that, helper threads using moderately sized small cores can significantly accelerate a larger core compared to using a hardware prefetcher alone. We also find that a small core provides a good trade-off against using an equivalent large core to run helper threads in a HMC.

In summary, despite the latency overheads of accessing prefetched cache lines from the shared L3 cache, helper thread based prefetching on small cores looks as a promising way to improve single thread performance on memory intensive workloads in HMC architectures.

This research was partially done in collaboration with Alain Ketterlin from the Inria Camus project-team in Strasbourg.

7.1.3.2. Branch Prediction and Performance of Interpreter

Participants: Erven Rohou, André Seznec, Bharath Narasimha Swamy.

Interpreters have been used in many contexts. They provide portability and ease of development at the expense of performance. The literature of the past decade covers analysis of why interpreters are slow, and many software techniques to improve them. A large proportion of these works focuses on the dispatch loop, and in particular on the implementation of the switch statement: typically an indirect branch instruction. Folklore attributes a significant penalty to this branch, due to its high misprediction rate. In [34], we revisit this assumption, considering state-of-the-art branch predictors and the three most recent Intel processor generations on current interpreters. Using both hardware counters on Haswell, the latest Intel processor generation, and simulation of the ITTAGE predictor [10], we show that the accuracy of indirect branch prediction is no longer critical for interpreters. We further compare the characteristics of these interpreters and analyze why the indirect branch is less important than before.

7.1.3.3. Augmenting superscalar architecture for efficient many-thread parallel execution

Participants: Sylvain Collange, André Seznec, Sajith Kalathingal.

Threads of Single-Program Multiple-Data (SPMD) applications often exhibit very similar control flows, i.e. they execute the same instructions on different data. In [42] we propose the Dynamic Inter-Thread Vectorization Architecture (DITVA) to leverage this implicit Data Level Parallelism on SPMD applications to create dynamic vector instructions at runtime. DITVA extends an in-order SMT processor with SIMD units with an inter-thread vectorization execution mode. In this mode, identical instructions of several threads running in lockstep are aggregated into a single SIMD instruction. DITVA leverages existing SIMD units and maintains binary compatibility with existing CPU architectures. To balance TLP and DLP, threads are statically grouped into fixed-size warps, inside which threads run in lockstep. At instruction fetch time, if the instruction streams of several threads within a warp are synchronized, then DITVA aggregates the instructions of the threads as dynamic vectors. To maximize vectorization opportunities, we use resource sharing arbitration policies that favor thread synchronization within warps. The policies do not require any compiler hints or modified algorithms for the existing SPMD applications and allow to run unmodified CPU binaries. A dynamic vector instruction is executed as a single unit. This allows to execute m identical instructions from m different threads on m parallel execution lanes while activating the I-fetch, the decode, and the overall pipeline control only once.

Our evaluation on the SPMD applications from the PARSEC and SPLASH benchmarks shows that a 4-warp 4-lane 4-issue DITVA architecture with a realistic bank-interleaved cache achieves 44% higher performance than a 4-thread 4-issue SMT architecture with AVX instructions while fetching and issuing 40 % fewer instrructions, achieving an overall 22% energy reduction.

7.2. Compiler, vectorization, interpretation

Participants: Erven Rohou, Emmanuel Riou, Bharath Narasimha Swamy, Arjun Suresh, André Seznec, Nabil Hallou, Sylvain Collange.

7.2.1. Improving sequential performance through memoization

Participants: Erven Rohou, Emmanuel Riou, Bharath Narasimha Swamy, André Seznec, Arjun Suresh.

Many applications perform repetitive computations, even when properly programmed and optimized. Performance can be improved by caching results of pure functions, and retrieving them instead of recomputing a result (a technique called memoization).

We propose [20] a simple technique for enabling software memoization of any dynamically linked pure function and we illustrate our framework using a set of computationally expensive pure functions – the transcendental functions.

Our technique does not need the availability of source code and thus can be applied even to commercial applications as well as applications with legacy codes. As far as users are concerned, enabling memoization is as simple as setting an environment variable.

Our framework does not make any specific assumptions about the underlying architecture or compiler tool-chains, and can work with a variety of current architectures.

We present experimental results for x86-64 platform using both gcc and icc compiler tool-chains, and for ARM cortex-A9 platform using gcc. Our experiments include a mix of real world programs and standard benchmark suites: SPEC and Splash2x. On standard benchmark applications that extensively call the transcendental functions we report memoization benefits of upto 16 %, while much higher gains were realized for programs that call the expensive Bessel functions. Memoization was also able to regain a performance loss of 76 % in *bwaves* due to a known performance bug in the gcc libm implementation of *pow* function.

This work has been published in ACM TACO 2015 [20] and accepted for presentation at the International Conference HiPEAC 2016.

7.2.2. Code Obfuscation

Participant: Erven Rohou.

This research is done in collaboration with the group of Prof. Ahmed El-Mahdy at E-JUST, Alexandria, Egypt.

We propose [24] to leverage JIT compilation to make software tamper-proof. The idea is to constantly generate different versions of an application, even while it runs, to make reverse engineering hopeless. More precisely a JIT engine is used to generate new versions of a function each time it is invoked, applying different optimizations, heuristics and parameters to generate diverse binary code. A strong random number generator will guarantee that generated code is not reproducible, though the functionality is the same.

This work was presented in January 2015 at the International Workshop on Dynamic Compilation Everywhere (DCE-2015) [24].

7.2.3. Dynamic Binary Re-vectorization

Participants: Erven Rohou, Nabil Hallou, Emmanuel Riou.

This work is done in collaboration with Philippe Clauss and Alain Ketterlin (Inria CAMUS).

Applications are often under-optimized for the hardware on which they run. Several reasons contribute to this unsatisfying situation, including the use of legacy code, commercial code distributed in binary form, or deployment on compute farms. In fact, backward compatibility of instruction sets guarantees only the functionality, not the best exploitation of the hardware. In particular SIMD instruction sets are always evolving.

We proposed [23] a runtime re-vectorization platform that dynamically adapts applications to execution hardware. The platform is built on top of Padrone. Programs distributed in binary forms are re-vectorized at runtime for the underlying execution hardware. Focusing on the x86 SIMD extensions, we are able to automatically convert loops vectorized for SSE into the more recent and powerful AVX. A lightweight mechanism leverages the sophisticated technology put in a static vectorizer and adjusts, at minimal cost, the width of vectorized loops. We achieve speedups in line with a native compiler targeting AVX. Our re-vectorizer is implemented inside a dynamic optimization platform; its usage is completely transparent to the user and requires neither access to source code nor rewriting binaries.

7.2.4. Dynamic Parallelization of Binary Executables

Participants: Erven Rohou, Nabil Hallou, Emmanuel Riou.

We address runtime automatic parallelization of binary executables, assuming no previous knowledge on the executable code. The Padrone platform is used to identify candidate functions and loops. Then we disassemble the loops and convert them to the intermediate representation of the LLVM compiler (thanks to the external tool McSema). This allows us to leverage the power of the polyhedral model for auto-parallelizing loops. Once optimized, new native code is generated just-in-time in the address space of the target process.

Our approach enables user transparent auto-parallelization of legacy and/or commercial applications with auto-parallelization.

This work is done in collaboration with Philippe Clauss (Inria CAMUS).

7.2.5. Hardware Accelerated JIT Compilation for Embedded VLIW Processors

Participant: Erven Rohou.

Just-in-time (JIT) compilation is widely used in current embedded systems (mainly because of Java Virtual Machine). When targeting Very Long Instruction Word (VLIW) processors, JIT compilation back-ends grow more complex because of the instruction scheduling phase. This tends to reduce the benefits of JIT compilation for such systems. We propose a hybrid JIT compiler where JIT management is handled in software and the back-end is performed by specialized hardware. Experimental studies show that this approach leads to a compilation up to 15 times faster and 18 times more energy efficient than a pure software compilation.

This work is done in collaboration with the CAIRN team (Steven Derrien and Simon Rokicki).

7.2.6. Performance Assessment of Sequential Code

Participant: Erven Rohou.

The advent of multicore and manycore processors, including GPUs, in the customer market encouraged developers to focus on extraction of parallelism. While it is certainly true that parallelism can deliver performance boosts, parallelization is also a very complex and error-prone task, and many applications are still dominated by sequential sections. Micro-architectures have become extremely complex, and they usually do a very good job at executing fast a given sequence of instructions. When they occasionally fail, however, the penalty is severe. Pathological behaviors often have their roots in very low-level details of the micro-architecture, hardly available to the programmer. In [33], we argue that the impact of these low-level features on performance has been overlooked, often relegated to experts. We show that a few metrics can be easily defined to help assess the overall performance of an application, and quickly diagnose a problem. Finally, we illustrate our claim with a simple prototype, along with use cases.

7.2.7. Compilers for emerging throughput architectures

Participant: Sylvain Collange.

This work is done in collaboration with Douglas de Couto and Fernando Pereira from UFMG.

The increasing popularity of Graphics Processing Units (GPUs) has brought renewed attention to old problems related to the Single Instruction, Multiple Data execution model. One of these problems is the reconvergence of divergent threads. A divergence happens at a conditional branch when different threads disagree on the path to follow upon reaching this split point. Divergences may impose a heavy burden on the performance of parallel programs.

We have proposed a compiler-level optimization to mitigate the performance loss due to branch divergence on GPUs. This optimization consists in merging function call sites located at different paths that sprout from the same branch. We show that our optimization adds negligible overhead on the compiler. When not applicable, it does not slow down programs and it accelerates substantially those in which it is applicable. As an example, we have been able to speed up the well known SPLASH Fast Fourier Transform benchmark by 11 %.

7.2.8. Deterministic floating-point primitives for high-performance computing

Participant: Sylvain Collange.

This work is done in collaboration with David Defour (UPVD), Stef Graillat and Roman Iakymchuk (LIP6).

Parallel algorithms such as reduction are ubiquitous in parallel programming, and especially high-performance computing. Although these algorithms rely on associativity, they are used on floating-point data, on which operations are not associative. As a result, computations become non-deterministic, and the result may change according to static and dynamic parameters such as machine configuration or task scheduling.

We introduced a solution to compute deterministic sums of floating-point numbers efficiently and with the best possible accuracy. A multi-level algorithm incorporating a filtering stage that uses fast vectorized floating-point expansions and an accumulation stage based on superaccumulators in a high-radix carry-save representation guarantees accuracy to the last bit even on degenerate cases while maintaining high performance in the common cases [16]. Leveraging these algorithms, we build a reproducible BLAS library [49] and extend the approach to triangular solvers [25].

7.3. WCET estimation and optimization

Participants: Hanbing Li, Isabelle Puaut, Erven Rohou, Damien Hardy, Viet Anh Nguyen, Benjamin Rouxel.

7.3.1. WCET estimation for architectures with faulty caches

Participants: Damien Hardy, Isabelle Puaut.

This is joint work with Yannakis Sazeides from University of Cyprus

Fine-grained disabling and reconfiguration of hardware elements (functional units, cache blocks) will become economically necessary to recover from permanent failures, whose rate is expected to increase dramatically in the near future. This fine-grained disabling will lead to degraded performance as compared to a fault-free execution.

Until recently, all static worst-case execution time (WCET) estimation methods were assuming fault-free processors, resulting in unsafe estimates in the presence of faults. The first static WCET estimation technique dealing with the presence of permanent faults in instruction caches was proposed in [4]. This study probabilistically quantified the impact of permanent faults on WCET estimates. It demonstrated that the probabilistic WCET (pWCET) estimates of tasks increase rapidly with the probability of faults as compared to fault-free WCET estimates.

New results show that very simple reliability mechanisms allow mitigating the impact of faulty cache blocks on pWCETs. Two mechanisms, that make part of the cache resilient to faults are analyzed. Experiments show that the gain in pWCET for these two mechanisms are on average 48% and 40% as compared to an architecture with no reliability mechanism.

This work will appear at DATE 2016.

7.3.2. Speeding up Static Probabilistic Timing Analysis

Participants: Damien Hardy, Isabelle Puaut.

This is joint work with Suzana Milutinovic, Jaume Abella, Eduardo Quinones and Francisco J. Cazorla from Barcelona Supercomputing Center.

Probabilistic Timing Analysis (PTA) has emerged recently to derive trustworthy and tight WCET estimates. For its static variant, called SPTA, we identify one of the main elements that jeopardizes its scalability to real-size programs: its high computation time cost. This SPTA's high computational costs are due to convolution, a mathematical operator used by SPTA and also deployed in many domains including signal and image processing.

In [40], we show how convolution is applied in SPTA, and qualitatively and quantitatively evaluate optimizations developed in other domains to reduce convolution time cost when applied to SPTA, and SPTA-specific optimizations. We show that SPTA-specific optimizations provide larger execution time reductions than generic cores.

7.3.3. Traceability of flow information for WCET estimation

Participants: Hanbing Li, Isabelle Puaut, Erven Rohou.

This research is part of the ANR W-SEPT project.

Control-flow information is mandatory for WCET estimation, to guarantee that programs terminate (e.g. provision of bounds for the number of loop iterations) but also to obtain tight estimates (e.g. identification of infeasible or mutually exclusive paths). Such flow information is expressed through annotations, that may be calculated automatically by program/model analysis, or provided manually.

The objective of this work is to address the challenging issue of the mapping and transformation of the flow information from high level down to machine code. In our recent work, we have proposed a framework to systematically transform flow information from source code to machine code. The framework [11] defines a set of formulas to transform flow information for standard compiler optimizations. Transforming the flow information is done within the compiler, in parallel with transforming the code. There thus is no guessing what flow information have become, it is transformed along with the code.

Our most recent results in this framework were to add support for vectorization [26]. We implemented our approach in the LLVM compiler. In addition, we show through measurements on single-path programs that vectorization improves not only average-case performance but also WCETs. The WCET improvement ratio ranges from 1.18x to 1.41x depending on the target architecture on a benchmark suite designed for vectorizing compilers (TSVC).

This work is part of a more general traceability framework, designed and implemented within the ANR W-SEPT project and described in paper [21]. In this paper, we introduce a complete semantic-aware WCET estimation workflow. We introduce some program analysis to find infeasible paths: they can be performed at design, C or binary level, and may take into account information provided by the user. We design an annotation-aware compilation process that enables to trace the infeasible path properties through the program transformations performed by the compilers. Finally, we adapt the WCET estimation tool to take into account the kind of annotations produced by the workflow.

7.3.4. WCET estimation for many core processors

Participants: Viet Anh Nguyen, Damien Hardy, Isabelle Puaut.

This research is part of the PIA Capacités project.

The overall goal of this research is to defined WCET estimation methods for parallel applications running on many-core architectures, such as the Kalray MPPA machine.

Some approaches to reach this goal have been proposed, but they assume the mapping of parallel applications on cores already done. Unfortunately, on architectures with caches, task mapping requires a priori known WCETs for tasks, which in turn requires knowing task mapping (i.e., co-located tasks, co-running tasks) to have tight WCET bounds. Therefore, scheduling parallel applications and estimating their WCET introduce a chicken and egg situation.

In [41], we address this issue by developing an optimal integer linear programming formulation for solving the scheduling problem, whose objective is to minimize the WCET of a parallel application. Our proposed static partitioned non-preemptive mapping strategy addresses the effect of local caches to tighten the estimated WCET of the parallel application. We report preliminary results obtained on synthetic parallel applications.

ATEAMS Project-Team

6. New Results

6.1. Faster Immutable Data Structures for the JVM

Immutable data structures involve copying when updating. Efficient implementations use persistent data-structures, so that most of the unchanged data is shared between the copies. Existing libraries for such data structures in the context of the Java virtual machine (JVM), such as the data structures in Clojure and Scala, are based on Hash Array-Mapped Tries (HAMTs), which provide efficient insertion and concatenation operations for persistent maps and sets. In [37] Steindorfer and Vinju presented additional optimisation which allow such operations to be up to 28 times faster than in the Clojure and Scala libraries. Furthermore, the cost of equality checking of such data structures is lower as well. All this, without incurring additional memory.

6.2. Automated Measurement and Analysis of Open Source Software

> Deciding whether an open source software (OSS) meets the required standards for adoption in terms of quality, maturity, activity of development and user support is not a straightforward process. It involves analysing various sources of information, including the project's source code repositories, communication channels, and bug tracking systems. OSSMETER extends state-of-the-art techniques in the field of automated analysis and measurement of open-source software (OSS), and develops a platform that supports decision makers in the process of discovering, comparing, assessing and monitoring the health, quality, impact and activity of opensource software. To achieve this, OSSMETER computes trustworthy quality indicators by performing advanced analysis and integration of information from diverse sources including the project metadata, source code repositories, communication channels and bug tracking systems of OSS projects [29], [26]

This result comes from intensive collaboration in the FP7 STREP project “OSSMETER”. The ATEAMS contribution is focused around source code metrics and activity analysis for Java and PHP.

6.3. Modular Interpreters for the Masses

Object Algebras [46] are new design pattern for increased modularity and extensibility of tree based, abstract data types. By modelling the abstract syntax of a language as a generic factory interface, implementations of this interface provide multiple semantics of the data. For instance, one can define evaluation, type checking and pretty printing of the abstract syntax fully modularly. Additionally, the pattern allows syntax extension: adding a new constructor to the datatype, and modularly extending any existing interpretations to deal with the construct. The same interpretation of different constructs, however, might involve different kinds of context information. For instance, evaluation of arithmetic expressions does not require any context information, but evaluation of variables and binders requires and environment. In [34], Inostroza and Van der Storm introduce a simple, modular, and type safe technique to allow such interpretations to be composed anyway. It is based on lifting one interpreter to implicitly propagate the context information it does not require, so that the signatures of the interpreters become compatible. As a result, semantic definitions of language modules do not have to anticipate all kinds of context information that might be required by other modules with which it might be composed. The technique is simple, does not sacrifice separate compilation, is easy automate, and works in mainstream languages. It provides a first step towards a foundation for defining language by assembling modular building blocks.

6.4. One Parser to Rule Them All

Parsing realistic languages requires much more than just a parsing algorithm. Different kinds of language require advanced disambiguation, operator priorities, off-side rule checking, whitespace dependence or data dependence. In [25], Afroozeh and Izmaylova showed how most of these concerns are actually instances of data dependent parsing: the parsing process depends on the value of previously parsed input. They provided an encoding of indentation sensitive parsing, operator precedence and parsing in the presence of preprocessor directives, to a simple, data dependent core language which is executed using the general parsing algorithm GLL. By exposing the data dependent machinery at the level of the grammar formalism , this opens up a range of possibilities for custom parsing aspects, and provides a clear semantics for existing concerns like disambiguation.

6.5. A Pattern-Based Game Mechanics Design Assistant

Video game designers iteratively improve player experience by play testing game software and adjusting its design. Deciding how to improve gameplay, however, is difficult and time-consuming: designers lack an effective means for exploring decision alternatives and modifying a game's mechanics. In [35], Van Rozen presented the Mechanics Pattern Language (MPL) for encoding common game economy structures and design intent, and a Mechanics Design Assistant (MeDeA) for analyzing, explaining, understanding existing mechanics, and generating, filtering, exploring and applying design alternatives for modifying mechanics. As a result, game designers' productivity and game quality is increased by providing feedback and design alternatives early in the development cycle. Furthermore, the game economy modifications are applied at runtime using the MicroMachinations library, so that the effect of changes can be immediately experienced.

CAIRN Project-Team

7. New Results

7.1. Reconfigurable Architecture Design

7.1.1. Design Flow and Run-Time Management for Compressed FPGA Configurations

Participants: Olivier Sentieys, Christophe Huriaux.

Almost since the creation of the first SRAM-based FPGAs there has been a desire to explore the benefits of partially reconfiguring a portion of an FPGA at run-time while the remainder of design functionality continues to operate uninterrupted. Currently, the use of partial reconfiguration imposes significant limitations on the FPGA design: reconfiguration regions must be constrained to certain shapes and sizes and, in many cases, bitstreams must be precompiled before application execution depending on the precise region of the placement in the fabric. We developed an FPGA architecture that allows for seamless translation of partially-reconfigurable regions, even if the relative placement of fixed-function blocks within the region is changed.

In [42] we proposed a design flow for generating compressed configuration bit-streams abstracted from their final position on the logic fabric. Those configurations can then be decoded and finalized in real-time and at run-time by a dedicated reconfiguration controller to be placed at a given physical location. The VTR framework has been expanded to include bit-stream generation features. A bit-stream format is proposed to take part of our approach and the associated decoding architecture was designed. We analyzed the compression induced by our coding method and proved that compression ratios of at least $2.5\times$ can be achieved on the 20 largest MCNC benchmarks. The introduction of clustering which aggregates multiple routing resources together showed compression ratio up to a factor of $10\times$, at the cost of a more complex decoding step at runtime. The VBS approach can provide increased online relocation capabilities using a decoding algorithm capable of decoding the VBS on-the-fly during the task migration.

7.1.2. Run-Time Approximation under Performance Constraints in OFDM Wireless Receivers

Participants: Olivier Sentieys, Fernado Cladera.

Mobile wireless channels are characterized by time-varying multipath propagation, noise, and interference effects. To cope with these rapid variations of channel parameters, wireless receivers are designed with a significant performance margin to be able to reach a given link quality (BER - Bit Error Rate), even for the worst-case channel conditions. Indeed, one of the steps during the design phase is the choice of the architecture bit-width, and the smallest wordlength that ensures the correct behaviour of the receiver is usually chosen. In [39], an adaptive precision OFDM receiver is proposed. Significant energy savings come from varying at run time processing bit-width, based on estimation of channel conditions, without compromising BER constraints. To validate the energy savings, the energy consumption of basic operators has been obtained from real measurements for different bit-widths on a FPGA and a processor using soft SIMD. Results show that up to 62% of the dynamic energy consumption can be saved using this adaptive technique. The algorithms proposed for the low complexity selector used to choose the processing word-length at run time, without modifying the standard OFDM frame, are detailed in [38].

7.1.3. Optical Interconnections for 3D Multiprocessor Architectures

Participants: Jiating Luo, Pham Van Dung, Cédric Killian, Daniel Chillet, Olivier Sentieys.

To address the issue of interconnection bottleneck in multiprocessor on a single chip, we study how an Optical Network-on-Chip (ONoC) can leverage 3D technology by stacking a specific photonics die. The objectives of this study target: i) the definition of a generic architecture including both electrical and optical components, ii) the interface between electrical and optical domains, iii) the definition of strategies (communication protocol) to manage this communication medium, and iv) new techniques to manage and reduce the power consumption of optical communications. The first point is required to ensure that electrical and optical components can be used together to define a global architecture. Indeed, optical components are generally larger than electrical components, so a trade-off must be found between the size of optical and electrical parts. For example, if the need in terms of communications is high, several waveguides and wavelengths must be necessary, and can lead to an optical area larger than the footprint of a single processor. In this case, a solution is to connect (through the optical NoC) clusters of processors rather than each single processor. For the second point, we study how the interface can be designed to take applications needs into account. From the different possible interface designs, we extract a high-level performance model of optical communications from losses induced by all optical components to efficiently manage Laser parameters. Then, the third point concerns the definition of high-level mechanisms which can handle the allocation of the communication medium for each data transfer between tasks. This part consists in defining the protocol of wavelength allocation. Indeed, the optical wavelengths are a shared resource between all the electrical computing clusters and are allocated at run time according to application needs and quality of service. The last point concerns the definition of techniques allowing to reduce the power consumption of on-chip optical communications. The power of each Laser can be dynamically tuned in the optical/electrical interface at run time for a given targeted bit-error-rate. Due to the relatively high power consumption of such integrated Laser, we study how to define adequate policies able to adapt the laser power to the signal losses.

In [44], we proposed a wavelength reservation protocol handled by an Optical Network Interface (ONI) Manager for reconfigurable ONoC based on shared waveguide. It allows to efficiently allocate, at runtime, the optical communication channels for a manycore architecture. We described the ONI manager architecture and reservation protocol. Synthesis results in a 28nm FDSOI technology demonstrated that our interface can support a clock frequency up to 550 MHz with 6 wavelengths managed. From these results, we can be optimistic about the scaling of the ONoC and its capacity to manage a large number of processors and more wavelengths.

In [55], we explored the trade-off among channel bandwidth alternatives, performance, area and power. We showed that the channel size has a strong impact on the system performance and cost. We employed synthetic and real application traffic executed on the GEM5 simulator. As a result, we show that different channel bandwidths can improve the execution time of an application up to 75%, while including low area and power penalties.

7.1.4. Arithmetic Operators for Cryptography and Fault-Tolerance

Participants: Arnaud Tisserand, Emmanuel Casseau, Nicolas Veyrat-Charvillon, Karim Bigou, Franck Bucheron, Jérémie Métairie, Gabriel Gallin.

Arithmetic Operators for Fast and Secure Cryptography.

Our paper [36], presented at CHES, describes a new RNS modular multiplication algorithm for efficient implementations of ECC over GF(p). Thanks to the proposition of RNS-friendly Mersenne-like primes, the proposed RNS algorithm requires 2 times less moduli than the state-of-art ones, leading to 4 times less precomputations and about 2 times less operations. FPGA implementations of our algorithm are presented, with area reduced up to 46 %, for a time overhead less than 10 %. Other RNS algorithms and implementations have been presented at RAIM [66].

Scalar recoding is popular to speed up ECC (elliptic curve cryptography) scalar multiplication: non-adjacent form, double-base number system, multi-base number system (MBNS). Ensuring uniform computation profiles is an efficient protection against some side channel attacks (SCA) in embedded systems. Typical ECC scalar multiplication methods use two point operations (addition and doubling) scheduled according to secret scalar digits. Euclidean addition chains (EAC) offer a natural SCA protection since only one point operation is used. Computing short EACs is considered as a very costly operation and no hardware implementation has been reported yet. We designed an hardware recoding unit for short EACs which works concurrently to scalar multiplication. It has been integrated in an in-house ECC processor on various FPGAs. The implementation results show similar computation times compared to non-protected solutions, and faster ones compared to typical protected solutions (e. g. 18 % speed-up over 192 b Montgomery ladder). A paper [62] has been presented at Compas conference.

In a collaboration with University College Cork (Ireland), we worked on the design of secure multipliers for asymmetric cryptography using asynchronous circuits. A common paper has been published at ASYNC Conference [37]. In this paper, a specially adjusted Latch-less Asynchronous Charge Sharing Logic (LACSL) is developed to inherently defend such architecture against DPA attacks. The proposed logic provides input data independent low-power/energy consumption which is attributed to interleaved charge sharing stages with non-static elements involved in the data path. A 32-bit LACSL Montgomery Multiplier (case study) is extensively tested through HSPICE simulations and great consistency in power/energy consumption is achieved. The normalized energy deviation and normalized standard deviation are only 0.048 and 0.011, respectively. Compared with the original ACSL implementation, besides the impressive energy coherence, 42% energy saving is demonstrated plus that the leakage power is 3.5 times smaller. Furthermore, the scalability of the proposed multiplier is explored where 64- bit, 128-bit and 256-bit designs are implemented. Again, great energy consistency is found with the highest deviation being 0.5%.

In collaboration with D. Pamula, we worked on fast and secure finite field multipliers for GF(2^m) arithmetic, a paper has been presented at DSD conference [53]. It presents details on fast and secure GF(2^m) multipliers dedicated to elliptic curve cryptography applications. Presented design approach aims at high efficiency and security against side channel attacks of a hardware multi- plier. The security concern in the design process of a GF(2^m) multiplier is quite a novel concept. Basing on the results obtained in course of conducted research it is argued that, as well as efficiency of the multiplier impacts the efficiency of the cryptoprocessor, the security level of the multiplier impacts the security level of the whole cryptoprocessor. Thus the goal is to find a tradeoff, to compromise efficiency, in terms of speed and area, and security of the multiplier. We intend to secure the multiplier by masking the operation, either by uniformization or by randomization of the power consumption of the device during its work. The design methodology is half automated. The analyzed field sizes are the standard ones, which ensure that a cryptographic system is mathematically safe. The described architecture is based on principles of Mastrovito multiplication method. It is very flexible and enables to improve the resistance against side channel attacks without degrading the multiplier efficiency.

In a collaboration with G. Abozaid (EJUST University Egypt), we worked on the FPGA implementation of arithmetic operators for very large numbers (millions of bits) in fully homomorphic encryption (FHE) applications. A journal paper has been published in IEEE Embedded Systems Letters [18].

ECC Crypto-Processor with Protections Against SCA.

A dedicated processor for elliptic curve cryptography (ECC) is under development. Functional units for arithmetic operations in GF(2^m) and GF(p) finite fields and 160-600-bit operands have been developed for FPGA implementation. Several protection methods against side channel attacks (SCA) have been studied. The use of some number systems, especially very redundant ones, allows one to change the way some computations are performed and then their effects on side channel traces. This work is done in the PAVOIS project. An ASIC version of the processor is under development and should be sent for fabrication in the beginning of 2016.

A. Tisserand has been invited speaker at the conference on elliptic curve cryptography (ECC): "Hardware Accelerators for ECC and HECC" [29].

Arithmetic Operators and Crypto-Processor for HECC.

In the HAH project, we study and prototype efficient arithmetic algorithms for hyperelliptic curve cryptography for hardware implementations (on FPGA circuits). We study new advanced arithmetic algorithms and representations of numbers for efficient and secure implementations of HECC in hardware. First results have been published in Compas conference [60] and RAIM workshop [68].

Arithmetic Operators for Fault Tolerance.

In the ARDyT and Reliasic projects, we work on computation algorithms, representations of numbers and hardware implementations of arithmetic operators with integrated fault detection (and/or fault tolerance) capabilities. The target arithmetic operators are: adders, subtracters, multipliers (and variants of multiplications by constants, square, FMA, MAC), division, square-root, approximations of the elementary functions. We study two approaches: residue codes and specific bit-level coding in some redundant number systems for fault detection/tolerance integration at the arithmetic operator/unit level. FPGA prototypes are under development.

7.2. Compilation and Synthesis for Reconfigurable Platform

7.2.1. Adaptive dynamic compilation for low power embedded systems

Participants: Steven Derrien, Simon Rokicki.

Just-in-time (JIT) compilers have been introduced in the 1960s and became popular in the mid-1990s with the Java virtual machine. The use of JIT techniques for bytecode languages brings both portability and performance, making it an attractive solution for embedded systems, as evidenced by the Dalvik framework used by Android.

When targeting embedded systems, JIT compilation is even more challenging. First, because embedded systems are often based on architectures with an explicit use of Instruction- Level Parallelism (ILP), such as Very Long Instruction Word (VLIW) processors. Those architectures are highly dependent of the quality of the compilation, mainly because of the instruction scheduling phase performed by the compiler. The other challenge lies in the high constraints of the embedded system: the energy and execution time overhead due to the JIT compilation must be carefully kept under control. This is even more true if the JIT system is to be used in the context of a heterogeneous multi-core system with support dynamic task migration for heterogeneous ISA cores and/or support dynamically reconfigurable machines.

To address these challenges, we are currently studying how it is possible to take advantage of custom hardware to speed-up (and reduce the energy cost of) the JIT compilation stage. In this framework, basic optimizations and JIT management are performed in software, while the compilation back-end is implemented by means of specialized hardware. This back-end involves both instruction scheduling and register allocation, which are known to be the most time consuming stages of such a compiler. The first results are very encouraging, and we are finalizing an FPGA-based demonstration of the system.

7.2.2. Design Tools for Reconfigurable Video Coding

Participants: Emmanuel Casseau, Yaset Oliva.

In the field of multimedia coding, standardization recommendations are always evolving. To reduce design time taking benefit of available SW and HW designs, Reconfigurable Video Coding (RVC) standard allows defining new codec algorithms. The application is represented by a network of interconnected components (so called actors) defined in a modular library and the behaviour of each actor is described in the specific RVC-CAL language. Dataflow programming, such as RVC applications, express explicit parallelism within an application. However general purpose processors cannot cope with both high performance and low power consumption requirements embedded systems have to face. We have investigated the mapping of RVC applications onto a dedicated multiprocessor platform. Actually, our goal is to propose an automated co-design flow based on the RVC framework. The design flow starts with the Dynamic Dataflow and CAL descriptions of an application and goes up to the deployment of the system onto the hardware platform. We also propose a framework to explore dynamic mapping algorithms for multiprocessors systems. Such an algorithm should be capable of computing a more efficient workload repartition based on the current configuration and performances of the system. The targeted platform is composed of several Processing Elements (PE). They follow a hierarchical organization: one PE plays the role of master and the others are slaves. The master assigns tasks (actors) to the slaves. The slaves execute the application tasks. The system has been implemented on a Zynq platform. The mapping is computed at runtime on the ARM processor while two clusters of 8 Microblazes each play the role of slaves. The DDR memory is split into two sections: one is reserved to the

Master and the other one is shared with the slaves. This later contains the actor's code. On the FPGA, the Microblazes are connected to private memories through the Local Memory Bus (LMB) that store the runtime copy. A common shared memory is used for the data exchanges between the processors. It contains the FIFOs for token exchanges between actors. The dynamic mapping algorithm aims at increasing data throughput. It starts by gathering the performance metrics of the system. It then identifies the processor with the highest workload. The algorithm evaluates the gain when moving the actor to one of the other processors. The migration is only valuable if the overhead of moving the actor is less that the gain. The actor that would lead to the highest gain is selected for migration. As a use case, we implement an MPEG-4 decoder algorithm onto a multi-core heterogeneous system deployed onto the Zynq platform from Xilinx [61] [69]. This work is done in collaboration with Lab-STICC Lorient.

7.2.3. High-Level Synthesis Based Rapid Prototyping of Software Radio Waveforms

Participants: Emmanuel Casseau, Mai Thanh Tran.

Software Defined Radio (SDR) is now becoming a ubiquitous concept to describe and implement Physical Layers (PHYs) of wireless systems. In this context, FPGA (Field Programmable Gate Array) technology is expected to play a key role. To this aim, leveraging the nascent High-Level Synthesis (HLS) tools, a design flow from high-level specifications to Register-Transfer Level (RTL) description can be thought to generate processing blocks that can be reconfigured at run-time. We thus propose a methodology for the implementation of run-time reconfiguration in the context of FPGA-based SDR. The design flow allows the exploration between dynamic partial reconfiguration and control signal based multi-mode design. This architectural tradeoff relies upon HLS and its associated design optimizations. We apply the methodology to the architectural exploration of a Fast Fourier Transform (FFT) for Long Term Evolution (LTE) standard as a use case.

7.2.4. Optimization of loop kernels using software and memory information

Participant: Angeliki Kritikakou.

Compilers optimize the compilation sub-problems one after the other, following an order which leads to less efficient solutions because the different sub-problems are independently optimized taking into account only a part of the information available in the algorithms and the architecture. In [19], we have presented an approach which applies loop transformations in order to increase the performance of loop kernels. The proposed approach focuses on reducing the L1, L2 data cache and main memory accesses and the addressing instructions. Our approach exploits the software information, such as the array subscript equations, and the memory architecture, such as the memory sizes. Then, it applies source-to-source transformations taking as input the C code of the loop kernels and producing a new C code which is compiled by the target compiler. We have applied our approach to five well-known loop kernels for both embedded processors and general purpose processors. From the obtained experimental results we observed speedup gains from 2 up to 18. [21] presents a new methodology for computing the Dense Matrix Vector Multiplication, for both embedded (processors without SIMD unit) and general purpose processors (single and multi-core processors with SIMD unit). The proposed methodology fully exploits the combination of the software (e.g., data reuse) and hardware parameters (e.g., data cache associativity) which are considered simultaneously giving a smaller search space and high-quality solutions. The proposed methodology produces a different schedule for different values of the (i) number of the levels of data cache; (ii) data cache sizes; (iii) data cache associativities; (iv) data cache and main memory latencies; (v) data array layout of the matrix and (vi) number of cores. With our experimental results we show that the proposed approach achieves increased performance than ATLAS state-of-the-art library with a speedup from 1.2 up to 1.45.

7.2.5. Leveraging Power Spectral Density for Scalable System-Level Accuracy Evaluation

Participants: Benjamin Barrois, Olivier Sentieys.

The choice of fixed-point word-lengths critically impacts the system performance by impacting the quality of computation, its energy, speed and area. Making a good choice of fixed-point word-length generally requires solving an NP-hard problem by exploring a vast search space. Therefore, the entire fixed-point refinement process becomes critically dependent on evaluating the effects of accuracy degradation. In [34], a novel technique for the system-level evaluation of fixed-point systems, which is more scalable and that renders better accuracy, was proposed. This technique makes use of the information hidden in the power-spectral density of quantization noises. It is shown to be very effective in systems consisting of more than one frequency sensitive components. Compared to state-of-the-art hierarchical methods that are agnostic to the quantization noise spectrum, we show that the proposed approach is $5\times$ to $500\times$ more accurate on some representative signal processing kernels.

7.3. Interaction between Algorithms and Architectures

7.3.1. Sensor-Aided Non-Intrusive Load Monitoring

Participants: Xuan-Chien Le, Olivier Sentieys.

Non-Intrusive Load Monitoring (NILM) plays an important role in energy management and energy reduction in buildings and homes. An NILM system does not need a large amount of deployed power meters to monitor the power usage of home devices. Instead, only one meter on the main power line is necessary to detect and identify the operating devices. There are many approaches to solve the problem of device determination in NILM. The features applied in low-frequency based approach essentially include the step-change (or edge) and the steady state. In [47] we introduced three algorithms to solve the $l1$-norm minimization problem in NILM and results on power measurements obtained from a real appliance deployment. With a small number of devices, the obtained precision varies from 75% to 99%, depending on the tolerance criterion to determine the steady state of a given device.

7.3.2. Posture and Gesture Recognition using Wireless Body Sensor Networks

Participants: Arnaud Carer, Alexis Aulery, Olivier Sentieys.

The BoWi project (Body Wold Interactions) aims at designing a Wireless Body Sensor Network (WBSN) for accurate Gesture and Body Movement estimation with extremely severe constraints in terms of footprint and power consumption. Advantages of such system mainly come from its possible use in indoor or outdoor environments without any additional equipment. The 3D geolocation approach will combine radio communication distance measurement and inertial sensors and it will also strongly benefit from cooperative techniques based on multiple observations and distributed computation. Different types of applications, as health care, activity monitoring and environment control, are considered and evaluated along with a human-machine interface expertise.

In [32] we presented three different use cases of WBSN for posture and gesture recognition developed by increasing demands in terms of accuracy: posture recognition, gesture recognition and motion capture. This work is based on a simulator designed to explore algorithmic solutions for posture and gesture identification. Simulation results were performed with a set of different algorithm and sensor proposals for three usages including a Principal Component Analysis (PCA) for posture classification. We show how sensor and algorithm can be carefully chosen according to application scenarios while minimising implementation complexity.

For applications based on predefined postures such as environment control and physical rehabilitation, we show in [31] that low cost and fully distributed solutions, that minimize radio communications, can be efficiently implemented. Considering that radio links provide distance information, we also demonstrate that the matrix of estimated inter-node distances offers complementary information that allows for the reduction of communication load. Our results are based on a simulator that can handle various measured input data, different algorithms and various noise models. Simulation results are useful and used for the development of real-life prototype.

7.3.3. Energy Harvesting and Power Management

Participants: Olivier Sentieys, Arnaud Carer, Trong-Nhan Le.

To design autonomous Wireless Sensor Networks (WSNs) with a theoretical infinite lifetime, energy harvesting (EH) techniques have been recently considered as promising approaches. Ambient sources can provide everlasting additional energy for WSN nodes and exclude their dependence on battery.

In [24], an efficient energy harvesting system which is compatible with various environmental sources, such as light, heat, or wind energy, was proposed. Our platform takes advantage of double-level capacitors not only to prolong system lifetime but also to enable robust booting from the exhausting energy of the system. Simulations and experiments show that our multiple-energy-sources converter (MESC) can achieve booting time in order of seconds. Although capacitors have virtual recharge cycles, they suffer higher leakage compared to rechargeable batteries. Increasing their size can decrease the system performance due to leakage energy. Therefore, an energy-neutral design framework providing a methodology to determine the minimum size of those storage devices satisfying energy-neutral operation (ENO) and maximizing system quality-of-service (QoS) in EH nodes, when using a given energy source, was also proposed. Experiments validating this framework are performed on a real WSN platform with both photovoltaic cells and thermal generators in an indoor environment. Moreover, simulations on OMNET++ showed that the energy storage optimized from our design framework is used up to 93.86%.

A Power Manager (PM) is usually embedded in EH wireless nodes to adapt the computation load by changing their wake-up interval according to the harvested energy. In order to prolong the network lifetime, the PM must ensure that every node satisfies the Energy Neutral Operation (ENO) condition. However, when a multi-hop network is considered, changing the wake-up interval regularly may cripple the synchronization among nodes and therefore, degrade the global system Quality of Service (QoS). In [25], a Wake-up Variation Reduction Power Manager (WVR-PM) was proposed to solve this issue. This PM is applied for wireless nodes powered by a periodic energy source (e.g. light energy in an office) over a constant cycle of 24 hours. Not only following the ENO condition, our power manager also reduces the wake-up interval variations of WSN nodes. Based on this PM, an energy-efficient protocol, named Synchronized Wake-up Interval MAC (SyWiM), was also proposed. OMNET++ simulation results using three different harvested profiles show that the data rate of a WSN node can be increased up to 65% and the latency reduced down to 57% compared to state-of-the-art PMs. Validations on a real WSN platform have also been performed and confirmed the efficiency of our approach.

7.3.4. Signal Processing for High-Rate Optical Communications

Participants: Trung-Hien Nguyen, Olivier Sentieys, Arnaud Carer.

Mary quadrature amplitude modulation (m-QAM) combined with coherent detection and digital signal processing (DSP) is a promising candidate for the implementation of next generation optical transmission systems. However, as the number of modulation levels increases, the sensitivity to system imperfections such as phase noise of the transmitter and the local oscillator lasers or fiber nonlinearities is exacerbated. Moreover, the amplitude and phase imbalances between the in-phase (I) and quadrature (Q) channels in the transmitter (Tx) and the front-end of the receiver (Rx), which is often referred to as IQ imbalance, is also troublesome if not compensated

In [52], we proposed a novel simple blind adaptive IQ imbalance compensation based on a decision-directed least-mean-square (DD-LMS) algorithm integrated to a modified butterfly FIR filter configuration. Since only 2 FIR filter coefficients-sets are used, instead of 4 in the conventional configuration, the time for updating the coefficients and the hardware resources (such as coefficient memories and number of look-up tables) in real time field-programmable gate array (FPGA) platforms is then reduced using this method. A reduction in hardware complexity by a factor of about 3 is achieved by the proposed joint method. The proposed structure is experimentally validated with a 40Gbit/s 16-QAM signal. A 7dB power penalty reduction is experimentally achieved at a bit error rate (BER) of 10^{-3} in the presence of a 10 degree phase imbalance, confirming the effectiveness of the proposed algorithm. The equalization capability remains even in the presence of group velocity dispersion along the link, which is numerically confirmed with optical fiber transmission up to 1200 km and 20 phase imbalance.

In [50], circular harmonic expansion-based carrier frequency offset estimation was investigated for optical m-QAM communication systems. The proposed method, combined with a gradient-descent algorithm, shows better performance compared to already proposed VVMFOE and 4th-power methods.

CAMUS Team

7. New Results

7.1. Formal Proofs for an Ordering Relation in Explicitly Parallel Programs

Participants: Alain Ketterlin, Éric Violard.

This project is a collaborative work with the COMPSYS Inria Team, in Lyon. Participants are: Paul Feautrier, Tomofumi Yuki.

The growing need to make use of available parallelism has led to new explicitly parallel language constructs. These constructs are usually grouped under the term *Task Parallelism*, because they aim to go beyond “simple” *Data Parallelism* (i.e., loop and array-based parallelism). Prominent examples of languages integrating task parallelism are X10 (http://x10-lang.org) and variants, Cilk (http://supertech.csail.mit.edu/cilk/), and recent versions of OpenMP (http://www.openmp.org). Most of the work on such languages has focused on efficient run-time support for *tasks*, in contrast with *threads*, i.e., for programs generating potentially large numbers of distinct tasks with explicit (but arbitrary) ordering between the tasks. However, little attention has been given to the static analysis and optimization of explicitly parallel programs, probably because their properties are much harder to formalize, compared to their sequential counterpart. Starting with the work of our colleagues Paul Feautrier and Tomofumi Yuki, from the Compsys team in Lyon, we have advanced the formalization and formally proved several properties of some fundamental building blocks for the analysis of certain classes of explicitly parallel programs.

Task parallelism is usually based on a few syntactic constructs to represent tasks and their synchronization. We use X10’s terminology (and syntax, with simplifications), but the corresponding constructs of other languages is usually obvious. Across all languages one finds a construct to start (or *spawn*) an asynchronous task, named `async` in X10, and a “container” construct, named `finish` in X10, whose role is to wait for the completion of all task spawned during the execution of its body. Given that these constructs allow the parallel execution of pieces of the program, a first question arises: is there a static (i.e., compile-time) way to decide whether two given statements are ordered, i.e., that the first necessarily executes before the other. Feautrier and Yuki (with colleagues) have defined such a criterion for programs made of `async` and `finish` [33], along with arbitrary statements and for-loops, defining the so-called *polyhedral fragment* of X10. The resulting (partial) relation, called *happens-before*, opens the door to various static analyses, like data-dependence analysis, which are at the heart of a range of optimization techniques. Here is a quick example:

```
finish
  for i in ...
      async
    for j in ...
        S(i,j)
```

$\mathtt{S}(i,j)$ happens before $\mathtt{S}(i',j')$ iff $i = i' \wedge j < j'$

The resulting condition, $i = i' \wedge j < j'$, defines exactly the situation in which two statement executions are ordered, and can be seen as an appropriate extension of the lexicographic order to explicitly parallel programs.

Our work on this basis has been to take the formal definition of happens-before (HB), and implement it in Coq (https://coq.inria.fr). The goal was first to prove various properties of the relation, like transitivity, and second to provide a formal proof of both correctness and completeness of HB itself. The first part has been fairly immediate, due to the high representative power of Coq. The second part took more time, and involved several new contributions. The major part of the work went into defining a formal semantics for the fragment of X10 needed by the definition of HB. Given the semantics, it was possible to obtain the relation between a program and its trace(s), and then to prove that HB is correct (i.e., if HB states that one statement executes before another, then these statements appear in order in all possible traces of the program), and that HB is complete (i.e., that statements that are always ordered in traces are actually recognized as such by HB). The complete proof scripts are available on the Inria forge (gforge.inria.fr), under the x10-coq project.

Further work has also started on extending *happens-before* to X10 programs using synchronization primitives called *clocks*, which are basically *barriers*, where distinct tasks can wait for each other. Since an unrestricted use of synchronization barriers can lead to deadlocks, X10 introduces “implicit clocks”, which are introduced (and scoped) by a finish construct, on which a task can “register”, and whose scoping rules ensure that any program point can only use the single “nearest” clock. These restrictions offer termination guarantees, which in turn enables a sound *happens-before* relation between statement instances. The “clock-less” HB relation can then be modified to take into account the additional ordering imposed by clocks. We have started work to update the semantics to the case of implicit clocks, and to formalize this extension in Coq.

7.2. Validity Conditions for Transformations of Non-Affine Programs

Participants: Alain Ketterlin, Philippe Clauss.

This project is a collaborative work with the CORSE Inria Team, in Grenoble. Participant is: Fabrice Rastello.

Representing loop nests with the help of the polyhedral model has been a powerful and fruitful strategy to enable automatic optimization and parallelization. However, this model places strong requirements on the input program, and in many cases these requirements are hard to meet. Because they are based on linear programming, polyhedral techniques require every constraint to be affine in loop counters and parameters. While this is easily verified for loop bounds in a large majority of programs, the same constraint imposed to memory access functions is often too strong. There are several reasons for this. First, programmers often linearize multi-dimensional arrays, turning straightforward accesses like t[i][j] into t1[i*n+j], with the unfortunate effect of placing their program outside the scope of the polyhedral model. Second, optimization often happens late in the compilation process (or even during just-in-time compilation at run-time), where multi-dimensional array accesses have been transformed by the compiler itself, for the needs of its earlier passes. Third, complex data storage strategies for certain classes of arrays, e.g., band or triangular matrices, may introduce non-linear access functions, and this non-linearity must be taken into account, e.g., for locality optimization. And fourth, some access functions are almost completely unspecified, like in the case of indirect accesses (t[s[i]]) or abstract mappings (t[f(i)]).

Our goal is to extend polyhedral analysis techniques to cover at least some of these cases, and see how far we can push the limits of the fundamental algorithms beyond pure linearity. We have started by considering the case of multi-dimensional array linearization, where the code doesn’t provide access functions for all (original) dimensions, but rather a single access function, which is linear in loop counters but contains parametric coefficients. Here is an example illustrating our initial target, which is taken from the gemver program in the polybench suite:

```
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
      // Was: A[i][j] = A[i][j] + u1[i] * v1[j] + ...;
S1:   *(n*i+A+j) = *(n*i+A+j) + *(u1+i) * *(v1+j) + ...;
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
      // Was: x[i] = x[i] + beta * A[j][i] * y[j];
```

```
S2:   *(x+i) = *(x+i) + beta * *(n*j+A+i) * *(y+j);
  // ...
```

The original form of the statements appear in comments, but what finally reaches the compiler is much more convoluted: basically, every array access appears as a pointer access whose effective address is a polynomial function mixing counters (i, j), array base addresses (A, u1, v1, x, y), and size parameters (n). In some other cases, the arrays have been "locally" linearized, i.e., the code still displays different arrays, but their inner dimensions have been linearized. In our example, statement S1 would appear as:

```
      // Was: A[i][j] = A[i][j] + u1[i] * v1[j] + ...;
S1:   A[n*i+j] = A[n*i+j] + u1[i] + v1[j] + ...;
```

This is an important special case in practice, and its particular structure helps a lot, for example, when data dependence analysis is needed.

Extending current polyhedral techniques to deal with non-affine accesses is a formidable endeavor, requiring the adaptation of the many algorithms developed over decades for analysis, scheduling, and code generation. Rather, we have started by studying a specific task, with immediate practical impact: given a non-affine loop nest *and* a specific desired transformation, what are the conditions under which this condition is valid? It is not unreasonable to expect the transformation to be provided by other means than pure analysis, for instance to be suggested by profiling data. In this case, the problem we are left with is the one of testing whether the given transformation is valid. This in turn requires testing the emptyness of a "problematic system". For any given loop nest, this can be written as:

$$
\bigvee_{(A,A')} \quad \begin{array}{lll}
 & \exists(v,v')\text{s.t.} & \\
 & v \in \mathcal{D}_A \wedge v' \in \mathcal{D}_{A'} & \text{(domain)} \\
\wedge & v \prec_{lex} v' & \text{(originalschedule)} \\
\wedge & A(v) = A'(v) & \text{(sameaccesslocation)} \\
\wedge & T_A(v) \neg \prec_{lex} T_{A'}(v') & \text{(transformedschedule)}
\end{array}
$$

where A and A' range over pairs of potentially conflicting accesses, v and v' are iteration vectors, $\mathcal{D}_A$ and $\mathcal{D}_{A'}$ are iteration domains, $A(v)$ and $A'(v')$ are access functions, and T_A and $T'_{A'}$ are schedules. The condition under which the transformation is valid is the projection of this set on parameter dimensions, i.e., the elimination of all variables representing counters. The difficulty of this comes from the non-affine condition expressing the equality of access functions.

Building on previous work, we have devised a projection procedure that eliminates all counters and leaves a (usually complex) condition on parameters. We have also developed several simplification strategies, applied during elimination and also on the final result, that overall produces a test deciding whether the targeted transformation can be applied. For instance, on the fully linearized version of the previous examples, when deciding whether the following transformation is legal:

$$T_{\texttt{S1}}(0,i,j) = (0,i,j) \qquad T_{\texttt{S2}}(1,i,j) = (0,j,i)$$

i.e., interchanging the second loop (around S2) and then applying fusion on both depth-2 loops, our elimination and simplification procedure produces the following run-time test:

```
  if ( ((y+n >= x+2) && (x+n >= y+2))
  || ((n >= 2) && (n*n+A >= x+1) && (x >= A+1))
  || ((n >= 2) && (u1+n >= x+1) && (x+n >= u1+2))
  || ((n >= 2) && (n+v1 >= x+1) && (x+n >= v1+1))
  || ((n*n+A >= y+1) && (y >= A+1) && (n >= 2)) || ...)
    // Transformation invalid: run the original version...
  else
    // Transformation valid: run the transformed version...
```

The reader may want to verify that this test actually corresponds to verifying that “arrays” do not overlap, but only as far as the given transformation requires it.

A systematic evaluation of our procedure on a benchmark suite has shown that the resulting tests are both accurate and incur very little run-time overhead. The overall mechanism compares favorably with alternative techniques aiming at dealing with non-affine access functions, which consist in statically reconstructing array dimensions [30]. This part of our work is ready for publication. However, to be completely competitive with alternative approaches, we need to find ways to complete the polyhedral compilation chain, with a prior effective scheduling algorithm and a posterior code generation algorithm.

7.3. Automatic Parallelization of Nonlinear Loops

Participants: Aravind Sukumaran-Rajam, Philippe Clauss.

Runtime code optimization and speculative execution are becoming increasingly prominent to leverage performance in the current multi- and many-core era. However, a wider and more efficient use of such techniques is mainly hampered by the prohibitive time overhead induced by centralized data race detection, dynamic code behavior modeling, and code generation. Most of the existing Thread Level Speculation (TLS) systems rely on naively slicing the target loops into chunks and trying to execute the chunks in parallel with the help of a centralized performance-penalizing verification module that takes care of data races. Due to the lack of a data dependence model, these speculative systems are not capable of doing advanced transformations, and, more importantly, the chances of rollback are high. The polyhedral model is a well- known mathematical model to analyze and optimize loop nests. The current state-of-art tools limit the application of the polyhedral model to static control codes. Thus, none of these tools can generally handle codes with while loops, indirect memory accesses, or pointers. Apollo (Automatic POLyhedral Loop Optimizer) is a framework that goes one step beyond and applies the polyhedral model dynamically by using TLS. Apollo can predict, at runtime, whether the codes are behaving linearly or not, and it applies polyhedral transformations on-the-fly.

Apollo has been extended to handle codes whose memory accesses and loop bounds are not necessarily linear [23], [14]. The proposed extension consists of modeling memory addresses that are accessed either as "tubes" obtained through linear regression, or as ranges. More generally, this approach expands the applicability of the polyhedral model at runtime to a wider class of codes. Plugging together both linear and nonlinear accesses to the dependence prediction model enables the application of polyhedral loop optimizing transformations even for nonlinear code kernels while also allowing a low-cost speculation verification.

This work takes part of Aravind Sukumaran-Rajam’s PhD thesis that has been defended November the 5th, 2015 [13].

7.4. Dynamic Code Generation for Speculative Polyhedral Optimization

Participants: Juan Manuel Martinez Caamano, Philippe Clauss.

We have developped a new runtime code generation technique for speculative loop optimization and parallelization, that allows to generate on-the-fly codes resulting from any polyhedral optimizing transformation of loop nests, such as tiling, skewing, loop fission, loop fusion or loop interchange, without introducing a penalizing time overhead. The proposed strategy is based on the generation of code bones at compile-time, which are parametrized code snippets either dedicated to speculation management or to computations of the original target program. These code bones are then instantiated and assembled at runtime to constitute the speculatively-optimized code, as soon as an optimizing polyhedral transformation has been determined. Their granularity threshold is sufficient to apply any polyhedral transformation, while still enabling fast runtime code generation. This strategy has been implemented in the speculative loop parallelizing framework Apollo.

7.5. The XFOR Programming Structure

Participants: Imen Fassi, Philippe Clauss, Cédric Bastoul.

We have proposed a new programming control structure called “xfor” or “multifor”, providing users a way to schedule explicitly the statements of a loop nest, and take advantage of optimization and parallelization opportunities that are not easily attainable using the standard programming structures, or using automatic optimizing compilers [19]. This is the PhD work of Imen Fassi, who started her work in 2013 and who defended her thesis November the 27th, 2015 [12].

It has been shown that xfor programs often reach better performance than programs optimized by fully automatic polyhedral compilers like Pluto [29]. It has also been shown that different versions of codes may perform very differently, although their memory behaviors are very similar. By analyzing further the origins of such performance differences, we noticed five important gaps in the currently adopted and well-established code optimization strategies [18], [19]: insufficient data locality optimization, excess of conditional branches in the generated code, too verbose code with too many machine instructions, data locality optimization resulting in processor stalls, and finally missed vectorization opportunities.

To ease and extend the usage of the XFOR structure, we have developed:

- Xfor-Wizard, which is a programming environment for XFOR programs, assisting users in writing XFOR codes and applying optimizing transformations. Automatic dependence analysis and comparisons against a referential code (XFOR-loops or classic for-loops) are achieved to order to help the user in ensuring semantic correctness of the written code.
- XFORGEN, which is a tool to automatically generate an XFOR code that is equivalent to for-loops that have been automatically transformed using a static polyhedral compiler. The generated XFOR code exhibits the parameters of the transformations that have been applied and thus can be modified for further optimizations.

7.6. Dynamic Optimization of Binary Code

Participants: Philippe Clauss, Alain Ketterlin.

This project is a collaborative work with the ALF Inria Team, in Rennes. Participants are: Erven Rohou and Nabil Hallou.

Automatic code optimizations have traditionally focused on source-to-source transformation tools and compiler IR-level techniques. Sophisticated techniques have been developed for some classes of programs, and rapid progress is made in the field. However, there is a persistent hiatus between software vendors having to distribute generic programs, and end-users running them on a variety of hardware platforms, with varying levels of optimization opportunities. The next decade may well see an increasing variety of hardware, as it has already started to appear particularly in the embedded systems market. At the same time, one can expect more and more architecture-specific automatic optimization techniques.

Unfortunately, many “old” executables are still being used although they have been originally compiled for now outdated processor chips. Several reasons contribute to this situation:

- commercial software is typically sold without source code (hence no possibility to recompile) and targets slightly old hardware to guarantee a large base of compatible machines;
- though not commercial, the same applies to most Linux distributions [0] – for example Fedora 16 (released Nov 2011) is supported by Pentium III (May 1999) [0];
- with the widespread cloud computing and compute servers, users have no guarantee as to where their code runs, forcing them to target the oldest compatible hardware in the pool of available machines.

All this argues in favor of binary-to-binary optimizing transformations. Such transformations can be applied either statically, i.e., before executing the target code, or dynamically, i.e., while the target code is running.

[0] with the exception of Gentoo that recompiles every installed package

[0] http://docs.fedoraproject.org/en-US/Fedora/16/html/Release_Notes/sect-Release_Notes-Welcome_to_Fedora_16.html

Dynamic optimization is mostly addressing adaptability to various architectures and execution environments. If practical, dynamic optimization should be preferred because it eliminates several difficulties associated with static optimization. For instance, when deploying an application in the cloud, the executable file may be handled by various processor architectures providing varying levels of optimization opportunities. Providing numerous different adapted binary versions cannot be a general solution. Another point is related to interactions between applications running simultaneously on shared hardware, where adaptation may be required to adjust to the varying availability of the resources. Finally, most code optimizations have a basic cost that has to be recouped by the gain they provide. Depending on the input data processed by the target code, an optimizing transformation may or may not be profitable.

We distinguish two classes of binary transformations:

1. code transformations that can be handled directly by analyzing and modifying the original binary code. We call such transformations *low-level binary transformations*;
2. code transformations that require a higher level of abstraction of the code in order to generate a very different, but semantically equivalent, optimized code. We call such transformations *high-level binary transformations*.

While we target both classes of transformations, the first was addressed by focusing on SSE to AVX transformations of vectorized codes [20].

In this work, we focus on SIMD ISA extensions, and in particular on the x86 SSE and AVX capabilities. Compared to SSE, AVX provides wider registers, new instructions, and new addressing formats. AVX has been first supported in 2011 by the Intel Sandy Bridge and AMD Bulldozer architectures. However, most existing applications take advantage only of SSE and miss significant opportunities. We show that it is possible to automatically convert SSE to AVX whenever profitable. The key characteristics of our approach are:

- we apply the transformation at run-time, i.e. when the hardware is known;
- we only transform hot loops (detected through very lightweight profiling), thus minimizing the overhead;
- we do *not* implement a vectorization algorithm in a dynamic optimizer, instead we recognize already statically vectorized loops, and convert them to a more powerful ISA at low cost.

For high-level binary transformations, we also focus on hot loops and loop nests appearing in executable codes. There is an important literature addressing automatic loop optimization and parallelization techniques. Such optimizations include combinations of loop interchange, loop fusion and fission, loop skewing, loop shifting and loop tiling. However, they are mostly applied at compile-time, either on the source code, or on an intermediate representation form of the code. The most advanced techniques are related to the polyhedral model.

Applying such advanced loop optimizing transformations at runtime, on a currently running binary code, without any previous knowledge, is our challenging goal. The same goal has been addressed in [8], but not at runtime. In this work, the binary code is analyzed and transformed without any constraint regarding the related time overhead. Candidate loops are identified regarding their compliance to the polyhedral model: the loop bounds and memory references must be convertible into linear functions of the loop indices. Then, compliant loop nests are translated into an equivalent program in C source code, in order to be used as input for the source-to-source polyhedral compiler Pluto [29]. The resulting optimized code is then compiled and re-injected into the original binary code.

While a similar approach should be considered to reach the same goal at runtime, it must be handled differently regarding three main issues:

1. At runtime, the time overhead of the employed analysis and optimization techniques must be small. Thus, any translation to source code, that would require costly steps for the de-compilation/re-compilation phases, must be avoided.
2. Static approaches, as the one presented in [8], can only handle loops that are syntactically compliant with the polyhedral model. However, it has been shown, with the Apollo framework, that loops

may exhibit a compliant behavior at runtime. Since we target runtime optimizations, we also can take advantage of the information that is only available at runtime, and maybe also use speculative techniques.

3. Binary codes may hide some interesting properties of the embedded loops, and may need very complex analysis techniques for discovering such properties. In short, a whole compiler for binary codes would be required.

To address these issues, we are currently investigating the strategy consisting first of translating, at runtime, any selected loop nest into the LLVM [0] intermediate representation form (LLVM-IR). This representation offers several advantages:

- Analysis and transformation passes of the LLVM compiler can be used on-the-fly, in order to discover and compute relevant information, and to safely transform the code;
- The LLVM just-in-time compiler can be used to compile the optimized code, which is in LLVM-IR, as an executable;
- Existing tools for loop optimization can be used, as Polly [0], for static polyhedral-compliant loops, or Apollo, for dynamic polyhedral-compliant loops.

Hence, this strategy requires a fast binary-to-LLVM-IR translator. For this purpose, we are currently using and extending McSema [0], which is a library for translating the semantics of native code to LLVM-IR. McSema supports translation of x86 machine code, including integer, floating point, and SSE instructions. Control flow recovery is separated from translation, permitting the use of custom control flow recovery front-ends.

For McSema to be able to handle mostly any code, we had to parametrize carefully its translation rules, and also to add some x86 SSE instructions that were not handled. McSema was recently plugged to the Padrone platform. Thus, any hot loop nest is now automatically converted into LLVM-IR, as illustrated in Figure 2 .

Instead of taking as input a binary file, McSema takes as input a code extract containing a hot loop nest, thanks to the code address provided by Padrone. Then, McSema builds the control flow graph of the input code and generates a corresponding LLVM-IR. The next step is to plug the polyhedral LLVM compiler Polly (phases *Canonicalication* to *CodeGeneration* in Figure 2), in order to generate automatically an optimized version of the target loop nest, that will be then compiled using the LLVM just-in-time compiler and re-injected in the running code.

7.7. Combining Locking and Data Management Interfaces

Participants: Jens Gustedt, Mariem Saied, Daniel Salas.

Handling data consistency in parallel and distributed settings is a challenging task, in particular if we want to allow for an easy to handle asynchronism between tasks. Our publication [5] shows how to produce deadlock-free iterative programs that implement strong overlapping between communication, IO and computation. The collaboration with Soumeya Hernane has continued after her thesis defence in 2013. It extends distributed lock mechanisms and combines them with implicit data management, and resulted in a journal submission, see [26].

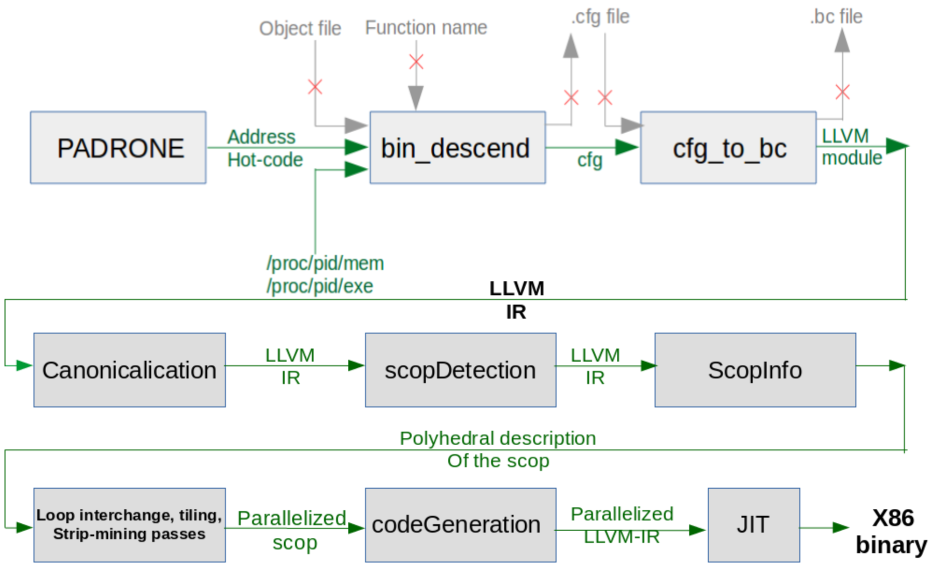
A new implementation (ORWL) of our ideas of combining control and data management in C has been undertaken, see 6.9 . In previous work it has demonstrated its efficiency for a large variety of platforms. In 2015, work on the ORWL model and library has gained vigor with the thesis of Mariem Saied (Inria) and Daniel Salas (INSERM). We also now collaborate on that subject with the TADAAM project team from Inria Bordeaux, where a postdoc has been hired through Inria funding.

In 2015, a new domain specific language (DSL) has been developed that largely eases the implementation of applications with ORWL. In its first version it provides an interface for stencil codes, but extensions towards other types of applications are on their way. In addition, work has been started to encapsulate imaging applications that use certain pipeline patterns to describe dependencies between computational task.

[0] http://llvm.org

[0] http://polly.llvm.org

[0] https://github.com/trailofbits/mcsema



- **Canonicaication passes:** set of llvm and polly passes that transform llvm-ir into a form that polly can handle.
- **SCOPs:** Static Control Part is a subgraph of CFG that has statically known control flow and can therefore be described within the polyhedral model.
- **Code generation:** takes a SCOP described within polyhedral model and translates it back to llvm-ir.

Figure 2. High-level Binary Loop Optimization through LLVM-IR

7.8. Efficient Execution of Polyhedral Codes on GPU and CPU+GPU Systems

Participants: Jean-François Dollinger, Vincent Loechner.

This is the main result of Jean-François Dollinger's PhD, started in 2012 and defended on July the 1st, 2015 [11].

Recent architectures complexity makes it difficult to statically predict the performance of a program. We have developped a reliable and accurate parallel loop nests execution time prediction method on GPUs for polyhedral codes. It is entirely automatic, and it based on three stages: static code generation, offline profiling on the target architecture, and online prediction.

In addition, we derived two techniques to fully exploit the computing resources at disposal on a computer. The first technique consists in jointly using all CPU cores and GPUs for executing a code. In order to achieve good performance, it is mandatory to consider load balance, in particular by predicting the execution time of a loop nest distributed on all those processing units. The runtime scheduler uses the profiling results to predict the execution times and adjust the parallel loop bounds to ensure load balance. The second technique puts CPU and GPU in a competition: instances of the considered code are simultaneously executed on CPU and GPU. The winner of the competition notifies its completion to the other instance, implying its termination.

7.9. Interactive Code Restructuring

Participants: Cédric Bastoul, Oleksandr Zinenko, Stéphane Huot.

This work falls within the exploration and development of semi-automatic programs optimization techniques. It consists in designing and evaluating new visualization and interaction techniques for code restructuring, by defining and taking advantage of visual representations of the underlying mathematical model. The main goal is to assist programmers during program optimization tasks in a safe and efficient way, even if they neither have expertise into code restructuring nor knowledge of the underlying theories. This project is an important step for the efficient use and wider acceptance of semi-automatic optimization techniques, which are still tedious to use and incomprehensible for most programmers. More generally, this research is also investigating new presentation and manipulation techniques for code, algorithms and programs, which could lead to many practical applications: collaboration, tracking and verification of changes, visual search in large amount of code, teaching, etc.

This is a rather new research direction which strengthens CAMUS's static parallelization and optimization issue. It is a joint work with two Inria teams specialized in interaction: EX-SITU at Inria Saclay (contact: Oleksandr Zinenko) and MJOLNIR at Inria Lille (contact: Stéphane Huot).

In 2015, we presented our interactive tool, *Clint*, that maps direct manipulation of the visual representation to polyhedral program transformations with real-time semantics preservation feedback. We conducted two user studies showing that Clint's visualization can be accurately understood by both experts and non-expert programmers, and that the parallelism can be extracted better from Clint's representation than from the source code in many cases [21]. We are planing a first release of that tool in the coming year.

7.10. Automatic Generation of Adaptive Simulation Codes

Participants: Cédric Bastoul, César Sabater.

Compiler automatic optimization and parallelization techniques are well suited for some classes of simulation or signal processing applications, however they usually don't take into account neither domain-specific knowledge nor the possibility to change or to remove some computations to achieve "good enough" results. Quite differently, production simulation and signal processing codes have adaptive capabilities: they are designed to compute precise results only where it matters if the complete problem is not tractable or if the computation time must be short. In this research, we design a new way to provide adaptive capabilities to compute-intensive codes automatically, inspired by Adaptive Mesh Refinement a classical numerical analysis technique to achieve precise computation only in pertinent areas. It relies on domain-specific knowledge provided through special pragmas by the programmer in the input code and on polyhedral compilation techniques, to continuously regenerate at runtime a code that performs heavy computations only where it matters at every moment. A case study on a fluid simulation application shows that our strategy enables dramatic computation savings in the optimized portion of the application while maintaining good precision, with a minimal effort from the programmer.

This research direction started in 2015 and complements our other efforts on dynamic optimization. We are in the process of a collaboration with Inria Nancy Grand Est team TONUS, specialized on applied mathematics (contact: Philippe Helluy), to bring models and techniques from this field to compilers. First results, investigated during the Inria Internship Program of César Sabater, have been presented to the SimRace international conference dedicated on industrial fluid simulation applications [16].

7.11. Polyhedral Compiler White-Boxing

Participants: Cédric Bastoul, Lénaïc Bagnères, Oleksandr Zinenko, Stéphane Huot.

While compilers offer a fair trade-off between productivity and executable performance in single-threaded execution, their optimizations remain fragile when addressing compute-intensive code for parallel architectures with deep memory hierarchies. Moreover, these optimizations operate as black boxes, impenetrable for the user, leaving them with no alternative to time-consuming and error-prone manual optimization in cases where an imprecise cost model or a weak analysis resulted in a bad optimization decision. To address this issue, we researched and designed a technique allowing to automatically translate an arbitrary polyhedral optimization, used internally by loop-level optimization frameworks of several modern compilers, into a sequence of comprehensible syntactic transformations as long as this optimization focuses on scheduling loop iterations. With our approach, we open the black box of the polyhedral frameworks enabling users to examine, refine, replay and even design complex optimizations semi-automatically in partnership with the compiler.

This research started in 2014 and we found the first solution in 2015. It has been conducted as a joint work between teams in compiler technologies (CAMUS and Inria Saclay's POSTALE team) and teams in interaction (EX-SITU at Inria Saclay and MJOLNIR at Inria Lille). The first paper on this has been accepted in 2015 to be presented in one of the top conferences on optimization techniques: CGO 2016 [15]. Subsequent work and a first release of the tool implementing the technique is planned during 2016.

COMPSYS Project-Team

7. New Results

7.1. Studying Optimal Spilling in the Light of SSA

Participants: Florian Brandner [ENSTA ParisTech, previously Compsys], Quentin Colombet [Apple, previously Compsys], Alain Darte.

Recent developments in register allocation, mostly linked to static single assignment (SSA) form, have shown the benefits of decoupling the problem in two phases: a first spilling phase places load and store instructions so that the register pressure at all program points is small enough, and a second assignment and coalescing phase maps the variables to physical registers and reduces the number of move instructions among registers. We focused on the first phase, for which many open questions remain: in particular, we studied the notion of optimal spilling (what can be expressed?) and the impact of SSA form (does it help?).

To identify the important features for optimal spilling on load-store architectures, we developed a new integer linear programming formulation, more accurate and expressive than previous approaches. Among other features, we can express SSA ϕ-functions, memory-to-memory copies, and the fact that a value can be stored simultaneously in a register and in memory. Based on this formulation, we presented a thorough analysis of the results obtained for the SPECINT 2000 and EEMBC 1.1 benchmarks, from which we have drawn, among others, the following conclusions: (1) rematerialization is extremely important; (2) SSA complicates the formulation of optimal spilling, especially because of memory coalescing when the code is not in conventional SSA (CSSA); (3) micro-architectural features are significant and thus have to be accounted for; and (4) significant savings can be obtained in terms of static spill costs, cache miss rates, and dynamic instruction counts.

Parts of this work were published at CASES 2011 [18]. The journal publication [1] contains more detailed discussions, more examples illustrating new concepts and existing approaches, and additional experiments covering the observed worst-case behavior, a new post-latency heuristic, and empiric evidence showing why static spill costs are a poor metric. Three configurations were added: Appel and George under SSA, Koes and Goldstein, and the heuristic of Braun and Hack.

7.2. Symbolic Range of Pointers in C programs

Participants: Vitor Paisante [Univ. Mineas Gerais, Brazil], Maroua Maalej, Leonardo Barbosa [Univ. Mineas Gerais, Brazil], Laure Gonnord, Fernando Pereira [Univ. Mineas Gerais, Brazil].

Alias analysis is one of the most fundamental techniques that compilers use to optimize languages with pointers. However, in spite of all the attention that this topic has received, the current state-of-the-art approaches inside compilers still face challenges regarding precision and speed. In particular, pointer arithmetic, a key feature in C and C++, is yet to be handled satisfactorily. We designed a new alias analysis algorithm to solve this problem. The key insight of our approach is to combine alias analysis with symbolic range analysis. This combination lets us disambiguate fields within arrays and structs, effectively achieving more precision than traditional algorithms. To validate our technique, we have implemented it on top of the LLVM compiler. Tests on a vast suite of benchmarks show that we can disambiguate several kinds of C idioms that current state-of-the-art analyses cannot deal with. In particular, we can disambiguate 1.35x more queries than the alias analysis currently available in LLVM. Furthermore, our analysis is very fast: we can go over one million assembly instructions in 10 seconds.

This work has been accepted at CGO'16 [30]. An extended version of the related work is available as an Inria research report [27] and will be the basis of a journal submission.

7.3. Analyzing C Programs with Arrays

Participants: Laure Gonnord, David Monniaux [CNRS/VERIMAG].

Automatically verifying safety properties of programs is hard, and it is even harder if the program acts upon arrays or other forms of maps. Many approaches exist for verifying programs operating upon Boolean and integer values (e.g., abstract interpretation, counter-examples guided abstraction refinement using interpolants), but transposing them to array properties has been fraught with difficulties.

In contrast to most preceding approaches, we do not introduce a new abstract domain or a new interpolation procedure for arrays. Instead, we generate an abstraction as a scalar problem and feed it to a preexisting solver. The intuition is that if there is a proof of safety of the program, it is likely that it can be expressed by elementary steps between properties involving only a small (tunable) number N of cells from the array.

Our transformed problem is expressed using Horn clauses over scalar variables, a common format with clear and unambiguous logical semantics, for which there exist several solvers. In contrast, solvers directly operating over Horn clauses with arrays are still very immature.

An important characteristic of our encoding is that it creates a non-linear Horn problem, with tree unfoldings, contrary to the linear problems obtained by flatly encoding the control-graph structure. Our encoding thus cannot be expressed by encoding into another control-flow graph problem, and truly leverages the Horn clause format.

Experiments with our prototype VAPHOR (see Section 6.9) show that this approach can prove automatically the functional correctness of several classical examples of the literature, including *selection sort*, *bubble sort*, *insertion sort*, as well as examples from previous articles on array analysis.

This work is presented in a research report [28] and is currently under submission.

7.4. Termination of C Programs

Participants: Laure Gonnord, David Monniaux [CNRS/VERIMAG], Gabriel Radanne [Univ Paris 7/ PPS].

The work of Compsys on the generation of multi-dimensional ranking functions [15], through a mix of polyhedral and abstract interpretation techniques, and its implementation in the tool RanK [16], was continued by Laure Gonnord in collaboration with D. Monniaux. A complete method for synthesizing lexicographic linear ranking functions (and thus proving termination), supported by inductive invariants, was designed in the case where the transition relation of the program includes disjunctions and existentials (large block encoding of control flow).

Previous work would either synthesize a ranking function at every basic block head, not just loop headers, which reduces the scope of programs that may be proved to be terminating, or expand large block transitions including tests into (exponentially many) elementary transitions, prior to computing the ranking function, resulting in a very large global constraint system. In contrast, the new algorithm incrementally refines a global linear constraint system according to extremal counterexamples: only constraints that exclude spurious solutions are included.

Experiments with our tool Termite 6.8 show marked performance and scalability improvements compared to other systems.

This work has been published at the PLDI'15 conference [7].

7.5. Data-aware Process Networks

Participants: Christophe Alias, Alexandru Plesco [XtremLogic SAS].

High-level circuit synthesis (HLS, high-level synthesis) consists in compiling a program described in a high-level programming language (as C) to a circuit. The circuit must be as efficient as possible while using properly the resources (power consumption, silicon area, FPGA elementary units, memory accesses, etc). Although a lot of progress was achieved on the back-end (low-level) aspects (pipeline generation, place/route), the front-end aspects (parallelism, I/O) are still rudimentary compared to the techniques developed by the HPC community, notably the analysis stemming from the *polyhedral model*.

We introduced data-aware process networks (DPN), a parallel execution model adapted to the hardware constraints of high-level synthesis, where the data transfers are made explicit. We have shown that the DPN model is consistent in the sense that any translation of a sequential program produces an equivalent DPN without deadlocks. Finally, we show how to compile a sequential program to a DPN and how to optimize the input/output and the parallelism.

This work has been published as an Inria research report [9] and will be submitted to a journal.

7.6. Mono-parametric Tiling

Participants: Guillaume Iooss, Sanjay Rajopadhye [Colorado State University], Christophe Alias, Yun Zou [Colorado State University].

Tiling is a crucial program transformation with many benefits. It improves locality, exposes parallelism, allows for adjusting the ops-to-bytes balance of codes, and can be applied at multiple levels. Allowing tile sizes to be symbolic parameters at compile time has many benefits, including efficient auto-tuning, and run-time adaptability to system variations. For polyhedral programs, parametric tiling in its full generality is known to be non-linear, breaking the mathematical closure properties of the polyhedral model. Most compilation tools therefore either avoid it by only performing fixed size tiling, or apply it only in the final, code generation step. Both strategies have limitations.

We first introduced mono-parametric partitioning, a restricted parametric, tiling-like transformation that can be used to express a tiling. We showed that, despite being parametric, it is a polyhedral transformation. We first proved that applying mono-parametric partitioning (i) to a polyhedron yields a union of polyhedra, and (ii) to an affine function produces a piecewise-affine function. We then used these properties to show how to partition an entire polyhedral program, including one with reductions. Next, we generalized this transformation to tiles with arbitrary tile shapes that can tessellate the iteration space (e.g., hexagonal, trapezoidal, etc). We showed how mono-parametric tiling can be applied at multiple levels, and how it enables a wide range of polyhedral analyses and transformations to be applied.

This work has been published as an Inria research report [14] and will be submitted to a journal. It is the extended version of the work published at IMPACT'14 [26].

7.7. Exact and Approximated Data-Reuse Optimizations for Tiling with Parametric Sizes

Participants: Alain Darte, Alexandre Isoard.

As mentioned in Section 7.6 , loop tiling is a loop transformation widely used to improve spatial and temporal data locality, to increase computation granularity, and to enable blocking algorithms, which are particularly useful when offloading kernels on computing units with smaller memories. When caches are not available or used, data transfers and local storage must be software-managed, and some useless remote communications can be avoided by exploiting data reuse between tiles. An important parameter of tiling is the sizes of the tiles, which impact the size of the required local memory. However, for most analyses involving several tiles, which is the case for inter-tile data reuse, the tile sizes induce non-linear constraints, unless they are numerical constants. This complicates or prevents a parametric analysis with polyhedral optimization techniques.

We showed that, when tiles are executed in sequence along tile axes, the parametric (with respect to tile sizes) analysis for inter-tile data reuse is nevertheless possible, i.e., one can determine, at compile-time and in a parametric fashion, the copy-in and copy-out data sets for all tiles, with inter-tile reuse, as well as sizes for the induced local memories (this is also connected to the liveness analysis described in Section 7.12). When approximations of transfers are performed, the situation is much more complex, and involves a careful analysis to guarantee correctness when data are both read and written. We provide the mathematical foundations to make such approximations possible, thanks to the introduction of the concept of *pointwise functions*. Combined with hierarchical tiling, this result opens perspectives for the automatic generation of blocking algorithms, guided by parametric cost models, where blocks can be pipelined and/or can contain parallelism. Previous work on FPGAs and GPUs already showed the interest and feasibility of such automation with tiling, but in a non-parametric fashion.

Our method is currently implemented with the `iscc` calculator of `ISL`, a library for the manipulation of integer sets defined with Presburger arithmetic, a complete implementation within the PPCG compiler is in progress (see also Section 6.7).

We believe that our approximation technique can be used for other applications linked to the extension of the polyhedral model as it turns out to be fairly powerful. Our future work will be to derive efficient approximation techniques, either because the program cannot be fully analyzable, or because approximations can speed-up or simplify the results of the analysis without losing much in terms of memory transfers and/or memory sizes.

A preliminary version of this work has been presented at the IMPACT'14 workshop [19]. A revised version was published at the International Conference on Compiler Construction (CC'15) [3].

7.8. Analysis of X10 Programs

Participants: Paul Feautrier, Alain Ketterlin [Inria/CAMUS], Sanjay Rajopadhye [Colorado State University], Vijay Saraswat [IBM Research], Eric Violard [Inria/CAMUS], Tomofumi Yuki.

While, historically, Compsys has applied polyhedral analysis to sequential programs, it was recently realized that it also applies to parallel programs or specifications, with the aim of checking their correctness or improving their performance. The prospect of having to program exascale architectures, with their millions of cores, has led to the development of new programming languages, whose objective is to increase the programmer productivity. Compsys has first applied polyhedral techniques to synchronous languages [24], [25] and pipelined specifications (see Section 7.7), before concentrating on IBM's high-productivity language X10 (see this section as well as Section 7.9) and on the OpenStream language (see Section 7.10).

X10 is based on the creation of independent *activities* (light-weight threads), which can synchronize either by a generalization of the fork/join scheme, or with *clocks*, an improved version of the familiar barriers. X10 is deadlock-free by construction but it is the programmer responsibility to insure determinism by a proper use of synchronizations. Non-determinism bugs may have a very low occurrence probability thus be very difficult to detect by testing, hence the interest for detecting races at compile time. In collaboration with CSU (S. Rajopadhye, T. Yuki) and IBM (V. Saraswat), we first extended array dataflow analysis to polyhedral clock-free X10 programs [34]. We have been working on clocked programs too. Race detection becomes undecidable [35], but realistic problems may still be solved by heuristics.

In cooperation with Eric Violard and Alain Ketterlin (Inria Team Camus, Strasbourg), and in order to obtain a more secure and precise analysis, we are currently attempting to formalize the "happens before" analysis used in these two previous papers [34], [35], using the proof assistant Coq.

7.9. Revisiting Loop Transformations with X10 Clocks

Participant: Tomofumi Yuki.

Loop transformations are known to be important for performance of compute-intensive programs, and are often used to expose parallelism. However, many transformations involving loops often obfuscate the code, and are cumbersome to apply by hand. In this work, we explored alternative methods for expressing parallelism that are more friendly to the programmer. In particular, we seek to expose parallelism without significantly changing the original loop structure. We illustrated how clocks in X10 can be used to express some of the traditional loop transformations, in the presence of parallelism, in a manner that we believe to be less invasive. Specifically, expressing parallelism corresponding to one-dimensional affine schedules can be achieved without modifying the original loop structure and/or statements.

This work was published at the international workshop on X10 [8].

7.10. Static Analysis of OpenStream Programs

Participants: Albert Cohen [Inria Parkas team], Alain Darte, Paul Feautrier.

In the context of the ManycoreLabs project (see Section 8.1), we also studied the applicability of polyhedral techniques to the parallel language OpenStream [31]. When applicable, polyhedral techniques are indeed invaluable for compile-time debugging and for generating efficient code well suited to a target architecture. OpenStream is a two-level language in which a control program directs the initialization of parallel task instances that communicate through *streams*, with possibly multiple writers and readers. It has a fairly complex semantics in its most general setting, but we restricted ourselves to the case where the control program is sequential, which is representative of the majority of the OpenStream applications.

In contrast to X10, this restriction offers deterministic concurrency by construction, but deadlocks are still possible. We showed that, if the control program is polyhedral, one may statically compute, for each task instance, the read and write indices to each of its streams, and thus reason statically about the dependences among task instances (the only scheduling constraints in this polyhedral subset). If the control program has nested loops, communications use one-dimensional channels in a form of linearization, and these indices may be polynomials of arbitrary degree, thus requiring to extend to polynomials the standard polyhedral techniques for dependence analysis, scheduling, and deadlock detection. Modern SMT allow to solve polynomial problems, albeit with no guarantee of success; the approach previously developed by P. Feautrier [6] may offer an alternative solution.

The usual way of disproving deadlocks is by exhibiting a schedule for the program operations, a well-known problem for polyhedral programs where dependences can be described by affine constraints. In the case of OpenStream, we established two important results related to deadlocks: 1) a characterization of deadlocks in terms of dependence paths, which implies that streams can be safely bounded as soon as a schedule exists with such sizes, 2) the proof that deadlock detection is undecidable, even for polyhedral OpenStream.

Details of this work are available in a research report [10]. It will be presented at the international workshop IMPACT'16 [2]. Some further developments are in progress for scheduling OpenStream programs using polynomial techniques, see Section 6.4 .

7.11. Handling Polynomials for Program Analysis and Transformation

Participant: Paul Feautrier.

As shown in Section 7.10 , many problems in parallel programs analysis and verification can be reduced to proving or disproving properties of polynomials in the variables of the program. For instance, the so-called "linearizations" (replacing a multi-dimensional object by a one-dimensional one) generate polynomial access functions. These polynomials then reappear in dependence testing, scheduling, and invariant construction. It may also happen that polynomials are absent from the source program, but are created either by an enabling analysis, as for OpenStream, or are imposed by complexity consideration. The usual solution is to construct a multi-dimensional function (e.g., a schedule for parallelization or a ranking function for termination [15]), which can then be converted into polynomials by counting. However, a direct approach is preferable, especially when the resulting schedule is to be used for further analysis, e.g., in real-time situations or WCET evaluation.

What is needed here is a replacement for the familiar emptiness tests and for Farkas lemma (deciding whether an affine form is positive inside a polyhedron). Recent mathematical results by Handelman and Schweighofer on the *Positivstellensatz* allow one to devise algorithms that are able to solve these problems. The difference is that one gets only sufficient conditions, and that complexity is much higher than in the affine cases. A paper presenting applications of these ideas to three use cases – dependence testing, scheduling, and transitive closure approximation – was presented at the 5th International Workshop on Polyhedral Compilation Techniques (IMPACT'15) [6] in Amsterdam in January 2015. A tool implementing polyhedral schedules complements this work, see Section 6.6 .

7.12. Liveness Analysis in Explicitly-Parallel Programs

Participants: Alain Darte, Alexandre Isoard, Tomofumi Yuki.

In the light of the parallel specifications encountered in our other works (from Section 7.7 to Section 7.11), we revisited scalar and array element-wise liveness analysis for programs with parallel specifications. In earlier work on memory allocation/contraction (register allocation or intra- and inter-array reuse in the polyhedral model), a notion of "time" or a total order among the iteration points was used to compute the liveness of values. In general, the execution of parallel programs is not a total order, and hence the notion of time is not applicable.

We first revised how conflicts are computed by using ideas from liveness analysis for register allocation, studying the structure of the corresponding conflict/interference graphs. Instead of considering the conflict between two pairs of live ranges, we only consider the conflict between a live range and a write. This simplifies the formulation from having four instances involved in the test down to three, and also improves the precision of the analysis in the general case.

Then we extended the liveness analysis to work with partial orders so that it can be applied to many different parallel languages/specifications with different forms of parallelism. An important result is that the complement of the conflict graph with partial orders is directly connected to memory reuse, even in presence of races. However, programs with conditionals do not even have a partial order, and our next step will be to handle such cases with more accuracy.

Details of this work are available in a research report [13]. It will be presented at the international workshop IMPACT'16 [4].

7.13. Extended Lattice-Based Memory Allocation

Participants: Alain Darte, Alexandre Isoard, Tomofumi Yuki.

We extended lattice-based memory allocation [20], an earlier work on memory (array) reuse analysis. The main motivation is to handle in a better way the more general forms of specifications we see today, e.g., with loop tiling, pipelining, and other forms of parallelism available in explicitly parallel languages. Our extension has two complementary aspects. We showed how to handle more general specifications where conflicting constraints (those that describe the array indices that cannot share the same location) are specified as a (non-convex) union of polyhedra. Unlike convex specifications, this also requires to be able to choose suitable directions (or basis) of array reuse. For that, we extended two dual approaches, previously proposed for a fixed basis, into optimization schemes to select suitable basis. Our final approach relies on a combination of the two, also revealing their links with, on one hand, the construction of multi-dimensional schedules for parallelism and tiling (but with a fundamental difference that we identify) and, on the other hand, the construction of universal reuse vectors (UOV), which was only used so far in a specific context, for schedule-independent mapping.

This algorithmic work, connected to the parametric tiling of Section 7.7 and the liveness analysis results of Section 7.12 , is complemented by a set of prototype scripting tools, see Section 6.3 .

Details of this work are available in a research report. It has also been submitted to a conference.

7.14. Stencil Accelerators

Participants: Steven Derrien [University of Rennes 1, Inria/CAIRN], Xinyu Niu [Imperial College London], Sanjay Rajopadhye [Colorado State University], Tomofumi Yuki.

Stencil computations have been known to be an important class of programs for scientific calculations. Recently, various architectures (mostly targeting FPGAs) for stencils are being proposed as hardware accelerators with high throughput and/or high energy efficiency. There are many different challenges for such design: How to maximize compute-I/O ratio? How to partition the problem so that the data fits on the on-chip memory? How to efficiently pipeline? How to control the area usage? We seek to address these challenges by combining techniques from compilers and high-level synthesis tools.

One project in collaboration with the CAIRN team and Colorado State University targets stencils with regular dependence patterns. Although many architectures have been proposed for this type of stencils, most of them use a large number of small processing elements (PE) to achieve high throughput. We are exploring an alternative design that aims for a single, large, deeply-pipelined PE. The hypothesis is that the pipelined parallelism is more area-efficient compared to replicating small PEs. We have published a work-in-progress paper on this topic at IMPACT'16 [5].

Another type of stencil accelerators that we are working on, in collaboration with Xinyu Niu, targets stencil programs with dynamic dependences (i.e., sparse computations). The collaboration is in the context of the EURECA project [0] where the dynamic reconfigurability of modern FPGAs are used to efficiently handle dynamic access patterns.

7.15. PolyApps

Participant: Tomofumi Yuki.

Loop transformation frameworks using the polyhedral model have gained increased attention since the rise of the multi-core era. We now have several research tools that have demonstrated their power on important kernels found in scientific computations. However, there remains a large gap between the typical kernels used to evaluate these tools and the actual applications used by the scientists.

PolyApps is an effort to collect applications from other domains of science to better establish the link between the compiler tools and "real" applications. The applications are modified to bypass some of the front-end issues of research tools, while keeping the ability to produce the original output. The goal is to assess how the state-of-the-art automatic parallelizers perform on full applications, and to identify new opportunities that only arise in larger pieces of code.

We showed that, with a few enhancements, the current tools will be able to reach and/or exceed the performance of existing parallelizations of the applications. One of the most critical element missing in current tools is the ability to modify the memory mappings.

[0] http://www.doc.ic.ac.uk/~nx210/2015/09/01/eureca.html

CORSE Team

6. New Results

6.1. An interval constrained memory allocator for the Givy GAS runtime

Participants: François Gindraud, Fabrice Rastello, Albert Cohen [ENS Ulm], Francois Broquedis.

This work presents a memory allocator for a global address space (GAS) execution environment targeting manycore architectures with distributed memory. Among the family of Multi Processor System on Chip (MPSoC), these devices are composed of multiple nodes linked by an on-chip network; most nodes have multiple processors sharing a small local memory. An MPSoC has an excellent performance-per-Watt ratio, but it is hard to program due to multilevel parallelism, explicit resource and memory management, and hardware constraints (limited memory, network topology).

Practical programming frameworks let the programmer in charge of the hard, target-specific work (e.g., threads or node-local OpenMP plus explicit communications). Automatic, more abstract frameworks exist for specific (scientific) applications, but they target big systems and do not model the hardware constraints of MPSoC. Givy is a runtime system to execute dynamic task graphs on MPSoC. It has a focus on supporting irregular applications, and uses data-flow semantics to coordinate dynamic task scheduling and data transfer. To simplify the programmer's view of memory, both runtime and program data objects live in a GAS. To avoid address collisions when objects are dynamically allocated, and to maintain the consistency of these addresses across explicit data transfers and virtual memory remapping, a GAS-aware memory allocator is required. The allocator proposed in this work has the following properties: (1) it is free of inter-node synchronizations; (2) it is well suited for small memory systems; (3) its performances match that of existing state-of-the-art allocators.

6.2. On Characterizing the Data Access Complexity (IO) of Programs and Using it for Architectural Design Exploration

Participants: Venmugil Elango [OSU], Naser Sedaghati [OSU], Fabrice Rastello, Louis-Noël Pouchet [UCLA], J. Ramanujam [LSU], Radu Teodorescu [OSU], P. Sadayappan [OSU].

Technology trends will cause data movement to account for the majority of energy expenditure and execution time on emerging computers. Therefore, computational complexity will no longer be a sufficient metric for comparing algorithms, and a fundamental characterization of data access complexity will be increasingly important. The problem of developing lower bounds for data access complexity has been modeled using the formalism of Hong & Kung's red/blue pebble game for computational directed acyclic graphs (CDAGs). However, previously developed approaches to lower bounds analysis for the red/blue pebble game are very limited in effectiveness when applied to CDAGs of real programs, with computations comprised of multiple sub-computations with differing DAG structure. We address this problem by developing an approach for effectively composing lower bounds based on graph decomposition. We also develop a static analysis algorithm to derive the asymptotic data-access lower bounds of programs, as a function of the problem size and cache size.

The roofline model is a popular approach to "bounds and bottleneck" performance analysis. It focuses on the limits to performance of processors because of limited bandwidth to off-chip memory. It models upper bounds on performance as a function of operational intensity, the ratio of computational operations per byte of data moved from/to memory. While operational intensity can be directly measured for a specific implementation of an algorithm on a particular target platform, it is of interest to obtain broader insights on bottlenecks, where various semantically equivalent implementations of an algorithm are considered, along with analysis for variations in architectural parameters. This is currently very cumbersome and requires performance modeling and analysis of many variants.

We alleviate this problem by using the roofline model in conjunction with upper bounds on the operational intensity of computations as a function of cache capacity, derived using lower bounds on data movement. This enables bottleneck analysis that holds across all dependence-preserving semantically equivalent implementations of an algorithm. We demonstrate the utility of the approach in in assessing fundamental limits to performance and energy efficiency for several benchmark algorithms across a design space of architectural variations.

This work is the fruit of the collaboration 8.4 with OSU. The first contribution (static analysis for lower bound) will be presented at ACM POPL'15 [10]. The second contribution (architectural exploration) is to be published at ACM TACO'15 [3].

6.3. A Tiling Perspective for Register Optimization

Participants: Duco Van Amstel, Lukasz Domagala, P. Sadayappan [OSU], Fabrice Rastello.

Register allocation is a much studied problem. A particularly important context for optimizing register allocation is within loops, since a significant fraction of the execution time of programs is often inside loop code. A variety of algorithms have been proposed in the past for register allocation, but the complexity of the problem has resulted in a decoupling of several important aspects, including loop unrolling, register promotion, and instruction reordering.

In this work, we develop an approach to register allocation and promotion in a unified optimization framework that simultaneously considers the impact of loop unrolling and instruction scheduling. This is done via a novel instruction tiling approach where instructions within a loop are represented along one dimension and innermost loop iterations along the other dimension. By exploiting the regularity along the loop dimension, and imposing essential dependence based constraints on intra-tile execution order, the problem of optimizing register pressure is cast in a constraint programming formalism. Experimental results are provided from thousands of innermost loops extracted from the SPEC benchmarks, demonstrating improvements over the current state-of-the-art.

This work is the fruit of both the collaboration 8.4 with OSU and with Kalray 7.1 7.2 .

6.4. Hybrid Data Dependence Analysis for Loop Transformations

Participants: Diogo Nunes Sampaio, Alain Ketterlin, Fabrice Rastello, Fernando Pereira, Alexandros Labrineas, Péricles Alves, Fabian Gruber.

Loop optimizations such as tiling, vectorization, or parallel task extraction are extremely important to achieve high performance. All such transformations rely on accurate memory dependence information to assess their validity. There are many practical situations, though, where dependence analysis fails to provide precise enough information. In this common scenario, the compiler will conservatively choose not to do any transformation. This happens in particular with low-level IRs (which are more and more common to address performance portability), but also in legacy code with pointers (e.g. C), linearized arrays, etc.

This work addresses the important problem of may-dependence disambiguation through the angle of a combination of static and dynamic analyses (sometimes called a hybrid analysis), similarly to what is already implemented in mainstream compilers, such as GCC, for auto-vectorization. This technique consists of adding a run-time test to disambiguate may-dependencies which static dependence analysis was not able to rule out. We propose two contributions to address this important problem.

The first approach proposes hybrid may-alias disambiguation. It combines two approaches: one that statically computes a symbolic expression of the interval of memory values a pointer may point to and uses dynamic overlap tests on these intervals to prove non-aliasing for each pair of pointers; another that hooks the memory allocator to find the base-pointer of a pointer and thus determine dynamically if a pointer pair belongs to two different allocations (and is thus disjoint) or not. We have applied these ideas on Polly-LLVM, a loop optimizer built on top of the LLVM compilation infrastructure. Our experiments indicate that our method is precise, effective and useful: we can disambiguate every pair of pointer in the loop intensive Polybench benchmark suite. The result of this precision is code quality: the binaries that we generate are 10% faster than those that Polly-LLVM produces without our optimization, at the -O3 optimization level of LLVM.

The second technique extends the non-overlapping intervals approach to may-dependence disambiguation. For this purpose, a powerful quantifier elimination scheme on multivariate-polynomials over integers has been developed. The quality of the presented scheme is important to make this approach realistic. In particular it must be precise (the integer aspect makes this problem very challenging), so that the test succeeds in practical cases, and must lead to negligible overhead. We evaluate preciseness and overhead on a set of 30+ benchmarks using complex loop transformations including loop fusion, skewing, and tiling.

This work is the fruit of the collaboration with UFMG 8.4 , Kalray 7.1 7.2 , STMicroelectronics 7.2 , and with EPI CAMUS in the context of IPL Multicore 8.2 . The first contribution has been presented at ACM OOPSLA'15 [19]. The second has been submitted to PLDI'16.

6.5. Power Efficiency and Computing Performance

Participants: Emilio Francesquini [UNICAMP, Campinas, Brazil], Edson Luiz Padoin [PhD: UFRGS and UNIJUI, Brazil], Marcio Castro [UFSC, Florianapolis, Brazil], Pedro Penna [PUC Minas, Belo Horizonte, Brazil], Henrique Cota de Freitas [PUC Minas, Belo Horizonte, Brazil], Fabrice Dupros [BRGM, Orléans, France], Philippe Navaux [UFRGS, Porto Alegre, Brazil], Jean François Méhaut.

Until the last decade, performance of HPC architectures has been almost exclusively quantified by their processing power. However, energy efficiency is being recently considered as important as raw performance and has become a critical aspect to the development of scalable systems. These strict energy constraints guided the development of a new class of so-called light-weight manycore processors. This study evaluates the computing and energy performance of two well-known irregular NP-hard problems – the Traveling-Salesman Problem (TSP) and K-Means clustering – and a numerical seismic wave propagation simulation kernel – Ondes3D – on multicore, NUMA, and manycore platforms. First, we concentrate on the nontrivial task of adapting these applications to a manycore, specifically the Kalray/MPPA-256 manycore processor. Then, we analyze their performance and energy consumption on those different machines. Our results show that applications able to fully use the resources of a manycore can have better performance and may consume from 3.8 × to 13 × less energy when compared to low-power and general-purpose multicore processors, respectively.

This work is the fruit of collaborations with Brazil and several universities (UFRGS, UFSC, UNICAMP, PUC Minas, USP). This work has been published in the journal of parallel and distributed computing [6] and in the journal of IET Computers and Digtal Techniques [7]. This work was also part of several international projects (LICIA, CNPq/Inria HOSCAR project, Exase).

Emilio Francesquini and Marcio Castro are also former PhD students of University Grenoble Alpes (UGA) and the LIG Laboratory.

6.6. Modeling and Simulating of Dynamic Task-Based Runtime Systems

Participants: Luka Stanisic [PhD, Inria, Mescal], Samuel Thibault [Univ. Bordeaux, Inria, Storm], Brice Videau, Arnaud Legrand [CNRS, Inria, Mescal], Jean François Méhaut.

Multi-core architectures comprising several GPUs have become mainstream in the field of High-Performance Computing. However, obtaining the maximum performance of such heterogeneous machines is challenging as it requires to carefully offload computations and manage data movements between the different processing units. The most promising and successful approaches so far build on task-based runtimes that abstract the machine and rely on opportunistic scheduling algorithms. As a consequence, the problem gets shifted to choosing the task granularity, task graph structure, and optimizing the scheduling strategies. Trying different combinations of these different alternatives is also itself a challenge. Indeed, getting accurate measurements requires reserving the target system for the whole duration of experiments. Furthermore, observations are limited to the few available systems at hand and may be difficult to generalize. In this work, we show how we crafted a coarse-grain hybrid simulation/emulation of StarPU, a dynamic runtime for hybrid architectures, over SimGrid, a versatile simulator for distributed systems. This approach allows to obtain performance predictions of classical dense linear algebra kernels accurate within a few percents and in a matter of seconds, which allows both runtime and application designers to quickly decide which optimization to enable or whether it is worth investing in higher-end GPUs or not. Additionally, it allows to conduct robust and extensive scheduling studies in a controlled environment whose characteristics are very close to real platforms while having reproducible behavior.

This work is part of the Luka Stanisic's thesis. Luka stanisic was coadvised by Arnaud Legrand, Brice Videau and Jean-François Méhaut. This thesis was defended in November 2015. Luka Stanisic currently holds a postdoc position at Inria Bordeaux in the Storm and HiePacs teams. This work was published in the CCPE journal [9].

6.7. Fast and Accurate Simulation of Multithreaded Sparse Linear Algebra Solvers

Participants: Luka Stanisic [PhD, Inria, Mescal], Arnaud Legrand [CNRS, Inria, Mescal], Emmanuel Agullo [Inria, HiePacs], Alfredo Buttari [CNRS, IRIT, Toulouse], Florent Lopez [CNRS, IRIT, Toulouse], Brice Videau.

The ever growing complexity and scale of parallel architectures imposes to rewrite classical monolithic HPC scientific applications and libraries as their portability and performance optimization only comes at a prohibitive cost. There is thus a recent and general trend in using instead a modular approach where numerical algorithms are written at a high level independently of the hardware architecture as Directed Acyclic Graphs (DAG) of tasks. A task-based runtime system then dynamically schedules the resulting DAG on the different computing resources, automatically taking care of data movement and taking into account the possible speed heterogeneity and variability. Evaluating the performance of such complex and dynamic systems is extremely challenging especially for irregular codes. In this article, we explain how we crafted a faithful simulation, both in terms of performance and memory usage, of the behavior of qr_mumps , a fully-featured sparse linear algebra library, on multi-core architectures. In our approach, the target high-end machines are calibrated only once to derive sound performance models. These models can then be used at will to quickly predict and study in a reproducible way the performance of such irregular and resource-demanding applications using solely a commodity laptop.

This work is part of the Luka Stanisic's thesis. Luka stanisic was coadvised by Arnaud Legrand, Brice Videau and Jean-François Méhaut. This thesis was defended in November 2015. Luka Stanisic currently holds a postdoc position at Inria Bordeaux in the Storm and HiePacs teams. This work was published in the ICPADS'2015 conference [18].

6.8. OpenMP Loop Scheduling

Participants: Pedro Penna [Master, PUC Minas, UFSC], Marcio Castro [Professor, UFSC], Henrique Cota de Freitas [Professor, PUC Minas], Francois Broquedis, Jean François Méhaut.

In High Performance Computing, the application's workload must be well balanced among the threads to achieve better performance. In this work, we propose a methodology that enables the design and exploration of new loop scheduling strategies. In this methodology, a simulator is used to evaluate the most relevant existing scheduling strategies, and a genetic algorithm is employed to explore the solution space of the problem itself. The proposed methodology allowed us to design a new loop scheduling strategy, which showed to be up to 32.3x better than the existing policies in terms of load balance.

6.9. BOAST: a Metaprogramming framework for computing kernels

Participants: Brice Videau [Postdoc CNRS, Mont-Blanc], Kevin Pouget [UJF, Nano2017], Luigi Genovese [Researcher, CEA INAC], Thierry Deutsch [Researcher, CEA INAC], Anthony Leonard [CNRS, Polytech Grenoble, Internship, from May 2015 until Aug 2015], Frederic Desprez, Jean François Méhaut.

Porting and tuning HPC applications to new platforms is tedious and costly in terms of human resources. Nonetheless, it is a very important aspect of the Mont-Blanc project. Indeed, for the Mont-Blanc project, more than ten applications were selected to be ported and optimized for the prototype platform.

Unfortunately, portability efforts are often lost when migrating to a new architecture. Worse, code may lose maintainability because several versions of some functionalities coexist, usually with a lot of duplication. Thus productivity of porting and tuning efforts is low as a huge fraction of those developments are never used after the platform they were intended for is decommissioned. Genericity of HPC codes is often limited. One of the reason is that producing generic code in Fortran 90/95 is difficult as the language does not really fit for it. Sometimes, adding genericity degrades performance as optimization opportunities that come from over-specification are lost. Functionality of HPC codes is tied to the previous point. Without genericity, adding new functionalities can be quite costly.

BOAST is a metaprogramming framework to produce portable and efficient computing kernels for HPC application. BOAST offers an embedded domain specific language to describe the kernels and their possible optimization. BOAST also supplies a complete runtime to compile, run, benchmark, and check the validity of the generated kernels. BOAST is being used in two flagship HPC applications BigDFT and SPECFEM3D, to improve performance portability of those codes.

BOAST is developped in the context of Mont-Blanc projects. It will be also used in the C2S@Exa IPL and the H2020/HPC4E project.

6.10. Performance comparison between Java and JNI for optimal implementation of computational micro-kernels

Participants: Nassim Halli [PhD student, CIFRE Aselta Nanographics], Henri-Pierre Charles [CEA LIST, CRI PILSI], Jean François Méhaut.

General purpose CPUs used in high performance computing (HPC) support a vector instruction set and an out-of-order engine dedicated to increase the instruction level parallelism. Hence, related optimizations are currently critical to improve the performance of applications requiring numerical computation. Moreover, the use of a Java run-time environment such as the HotSpot Java Virtual Machine (JVM) in high performance computing is a promising alternative. It benefits from its programming flexibility, productivity and the performance is ensured by the Just-In-Time (JIT) compiler. Though, the JIT compiler suffers from two main drawbacks. First, the JIT is a black box for developers. We have no control over the generated code nor any feedback from its optimization phases like vectorization. Secondly, the time constraint narrows down the degree of optimization compared to static compilers like GCC or LLVM. So, it is compelling to use statically compiled code since it benefits from additional optimization reducing performance bottlenecks. Java enables to call native code from dynamic libraries through the Java Native Interface (JNI). Nevertheless, JNI methods are not inlined and require an additional cost to be invoked compared to Java ones. Therefore, to benefit from better static optimization, this call overhead must be leveraged by the amount of computation performed at each JNI invocation. In this work we tackle this problem and we propose to do this analysis for a set of micro-kernels. Our goal is to select the most efficient implementation considering the amount of computation defined by the calling context. We also investigate the impact on performance of several different optimization schemes which are vectorization, out-of-order optimization, data alignment, method inlining and the use of native memory for JNI methods.

This work was presented in the ADAPT'2015 workshop. It's alsop part of the Nassim Halli's thesis.

6.11. Reducing trace size in multimedia applications endurance tests

Participants: Serge Emteu [PhD ST Microelectronics, LIG/Slide, CORSE], Miguel Santana [ST Microelectrnics], Alexadre Termier [Prof. Univ. Rennes I, IRISA/Inria/Dream], René Quiniou [CR Inria, IRISA/Inria/Dream], Brice Videau [Postdoc CNRS, Inria/Corse], Jean François Méhaut.

The consumer electronics market is dominated by embedded systems due to their ever-increasing processing power and the large number of functionnalities they offer. To provide such features, architectures of embedded systems have increased in complexity : they rely on several heterogeneous processing units, and allow concurrent tasks execution. This complexity degrades the programmability of embedded system architectures and makes application execution difficult to understand on such systems. The most used approach for analyzing application execution on embedded systems consists in capturing execution traces (event sequences, such as system call invocations or context switch, generated during application execution). This approach is used in application testing, debugging or profiling. However in some use cases, execution traces generated can be very large, up to several hundreds of gigabytes. For example endurance tests, which are tests consisting in tracing execution of an application on an embedded system during long periods, from several hours to several days. Current tools and methods for analyzing execution traces are not designed to handle such amounts of data.

We propose an approach for monitoring an application execution by analyzing traces on the fly in order to reduce the volume of recorded traces. Our approach is based on features of multimedia applications which contribute the most to the success of popular devices such as set-top boxes or smartphones. This approach consists in identifying automatically the suspicious periods of an application execution in order to record only the parts of traces which correspond to these periods. The proposed approach consists of two steps : a learning step which discovers regular behaviors of an application from its execution trace, and an anomaly detection step which identifies behaviors deviating from the regular ones.

The many experiments, performed on synthetic and real-life datasets, show that our approach reduces the trace size by an order of magnitude while maintaining a good performance in detecting suspicious behaviors.

This work was presented at the DATE conference in Grenoble. It was also part of the Serge Emteu's thesis wth ST Microelectronics.

6.12. Data Mining Approach to Temporal Debugging of Embedded Streaming Applications

Participants: Oleg Iegorov [PhD ST Microelectronics, LIG/Slide, CORSE], Miguel Santana [ST Microelectrnics], Alexadre Termier [Prof. Univ. Rennes I, IRISA/Inria/Dream], Vincent Leroy [Associate Professor UJF, LIG/Slide], Jean François Méhaut.

One of the greatest challenges in the embedded systems area is to empower software developers with tools that speed up the debugging of QoS properties in applications. Typical streaming applications, such as multimedia (audio/video) decoding, fulfill the QoS properties by respecting the realtime deadlines. A perfectly functional application, when missing these deadlines, may lead to cracks in the sound or perceptible artifacts in the image.

We start from the premise that most of the streaming applications that run on embedded systems can be expressed under a dataflow model of computation, where the application is represented as a directed graph of the data flowing through computational units called actors. It has been shown that in order to meet real-time constraints the actors should be scheduled in a periodic manner. We exploit this property to propose SATM – a novel approach based on data mining techniques that automatically analyzes execution traces of streaming applications, and discovers significant breaks in the periodicity of actors, as well as potential causes of these breaks. We show on a real use case that our debugging approach can uncover important defects and pinpoint their location to the application developer.

This work was presented at the EMSOFT conference in Amsterdam. It was also part of the Oleg Iegorov's thesis wth ST Microelectronics.

6.13. Tiling Bitwise Computations Using Look-up Instructions

Participants: Florent Bouchez - Tichadou, Cyril Six [Inria, Internship, from Feb 2015 until Jun 2015].

The BWLU is an instruction of a Very Long Instruction Word processor (VLIW) that performs a series of bit-independent computations in only one step through the use of a "look-up table" (LUT). The Bit-Wise Look-Up table instruction (BWLU) takes as input four registers as well as a 32-bit integer (the "table"), and is able to output two bit-independent computations based on the input registers into two output registers.

The goal is to make the best use possible of this instruction by replacing during compilation as much as possible groups of bitwise computation using BWLUs so as to reduce the number of instructions required to perform a computation. The problem is represented by a data-flow graph representing a computation, and the goal is use BLWUs as tiles to "match" groups of bitwise instruction.

We proved the problem NP-complete for a general data-flow graph, so it is not practical to try to find the optimal solution.

It is easy to devise a greedy algorithm that will produce a solution, but we wanted a way to check whether the solutions found where far from the optimal. An optimal algorithm is of course exponential in the size of the input graph, however, we devised a complete space exploration algorithm based on dynamic programming that manages to find the optimal solution for data-flow graphs with small width or height.

DREAMPAL Project-Team

6. New Results

6.1. HoMade in 2015

6.1.1. Interruption support

In the last release of HoMade we introduced interruptions. Up to 7 interruptions are supported. The priority is static and each trap is associated to one of the 7 first VCs of the master, they are called trap1 .. trap7. Trap is par nature reflective. When a trap is raised the HoMade master reaches a no-preemptive kernel. Traps have no effect on the slaves, they can continue to work. At the end of trap execution, HoMade master resumes the sequential execution, trap codes should be clean and should restitute the stack as it was when they began. A WAIT instruction and a long IP cannot be interrupted. An example of interrupts is provided in the reconfiguration part later.

6.1.2. New assembly language

HoMade waits for two binary codes: one for the master and one for the slaves. These two codes are loaded via the UART port and triggers a global reset of all the softcores after. Binary codes are a sequence of 16 bits words finishing by a long word filled with 4 NULL. Our post fixed macro assembler generates some binary codes from text files. This assembly language introduces some flow controls like if for repeat. It is also based on PC and VC definitions. Now the particular operator := generates reflective behaviors via WIM instructions. The syntax is so simple than everybody can understand a program. A full new syntax description is available with the assembler on the official HoMade web site : https://sites.google.com/site/homadeguide/assembleur-homade-v6. Here is the code for a mono HoMade to implement a reflective execution of Fibonacci suite. Switches values are put on the top on the stack to indicate the position in the list we want to process. Different input buttons affect the execution: • Button 0 changes to soft fibo execution using some library IPs. SWAP ROT DUP = - + are IPs to change the tops of the stack or to process dyadic integer operators. • Button 1 changes to hard execution using fibo vhdl long IP • Other buttons process the current fibo (hard or soft).

```
:IP fibo $AC54 ; // fibo hard IPcode 54
// XX = 1 YY = 1
program
  : read
    $1f // immediate hexa
    btnpush // IP reads buttons pushed
    switch // IP reads switches
  ;
  : fibo_soft // function declare
    1 1 rot
    3 -
    for
      dup rot +
    next
    swap
    drop
  ;
  VC fibo_dyn := fibo_soft // VC init soft
start
  begin
  read
```

```
    swap dup
    0 = // test button
    if // reflective process
      fibo_dyn := fibo_soft
    endif
    1 =
    if
      fibo_dyn := fibo
    endif
    fibo_dyn // call VC
    7seg // IP to print result
    $1f
    btn // button to pause
    7seg
    again // infinite loop
  endprogram
```

When the VC fibo_dyn is called, you call hard or soft Fibonacci version depending of the sequence of pushed button. The soft code is 7 time slower than the hard code. The extra cost due to reflective facility is 2 cycles by VC call.

6.1.3. Dynamic IP reconfiguration

Xilinx chips are offering capabilities to program some pre-reserved chip areas with different bitsreams and this during the execution itself. It is not instantaneous and even worse the reconfiguration time depends of the length of the bitstream (the size of the area). Do not abuse of partial reconfigurations! But for some applications where context evolves at a “human speed”, our design can benefit of this functionality to adapt the hardware to the current context. It is easy to introduce this notion in HoMade: just insert an IP! This IP has to manage the bitstream memory and the ICAP to load them in the predefined areas. We develop a such IP for the master, without broadcast of bitstream to the slaves for the moment. This IP reconfiguration only needs to know the bitstream address. Effectively for Xilinx, the data inside the bitstream are sufficient to achieve the reconfiguration. We introduced the new keyword ‘in the assembler in order to express IP reconfigurations. The declaration of reconfigurable IPs may also include the bitstream address. Now we can program dynamic partial reconfiguration of IPs using our dedicated IP that we developed. Furthermore we can couple the dynamic reconfiguration with the reflective notion. Here is a simple example with dynamic image filters. The filter processes 1 block of 3x3 pixels. The 9 pixels are stored on the 3 top of the stack by aggregation of 3 pixels per word. External actuators can change from one IP to the other. We used interrupts and traps to apply this migration.

```
  program // bistream addresses between ( )
    :IP IP_median $EC11 ($0);
    :IP IP_Sobel $EC22 ($49E);
    VC filter
    : T1
      IP_median ^^
      filter := IP_median
    ;
    : T2
      IP_moyenne ^^
      filter := IP_moyenne
      trap1 := T1 // interrupt level 1
      trap2 := T2 // Interrupt level 2
      : get3pix // must be defined &
    ;
  start
```

```
    begin
      $7D for
        get3Pix // 3x3 pixels on stack
        get3Pix
        get3Pix
        -rot swap
        $7D for
          filter // current IP
          get3Pix // next 3 pixels
          -rot
        next
      next
    again
  endprogram
```

Concerning dynamic reconfiguration of IPs, we are testing a dedicated IP to manage directly the ICAP of Xilinx. The different bitstreams are stored in DDR3 and this IP finds the starting address from the stack. Of course this is a long IP. Some optimization to broadcast efficiently the same bitstream towards different slave reconfigurable areas are still a big challenge with Xilinx architecture.

6.1.4. IP fusion

To be free from EDA companies, we are deploying IP fusion strategies to manage the dynamic reconfiguration by ourselves. We obtain good results concerning the reconfiguration time, but for large and very different IPs, the fusion works like an aggregation of two IPs and the surface gain is insignificant.

6.1.5. Using hardware parallelism for reducing power consumption in video streaming applications

In the PhD thesis of Karim Ali we exploited using a flexible parallel hardware-based architecture in conjunction with frequency scaling as a technique for reducing power consumption in video streaming applications. In this work, we derived equations to ease the calculation for the level of parallelism and the maximum depth for the FIFOs used for clock domain crossing. Accordingly, a design space was formed including all the design alternatives for the application. The preferable design alternative is selected in aware of how much hardware it costs and what power reduction goal it can satisfy. We used Xilinx Zynq ZC706 evaluation board to implement two video streaming applications: Video downscaler (1:16) and AES encryption algorithm to verify our approach. The experimental results showed up to 19.6% power reduction for the video downscaler and up to 5.4% for the AES encryption. The architecture and experimental results were published in a paper entitled "Using hardware parallelism for reducing power consumption in video streaming applications" at the 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC) in Jun 2015, Bremen, Germany [12].

In collaboration with NAVYA, we started the first steps to implement a stereo vision algorithm over a parallel architecture using FPGA technologies. The algorithm is based on a local approach for calculating the disparity map using sum of absolute difference between the right and the left image. As a first step, we exploited the possible optimization levels that can be applied at the software level. After that by using high level synthesis tool (Vivado HLS from Xilinx) the code was written in C in a way that facilitates its conversion into HDL files. Optimization techniques were applied to reduce both the hardware resources and time required for processing one frame. This design was tested experimentally to show around 50% decrease in the time required for processing one frame if compared to the software one. Currently, we are in the step of exploring more techniques for hardware optimization and decreasing the processing time to meet the industrial requirements of our partner.

6.1.6. A scalable flexible and dynamic reconfigurable architecture for high performance embedded computing

In collaboration with Nolam Embedded Systems (NES) and in the framework of the CIFRE PhD of Venkatasubramanian Viswanathan, we proposed a scalable and customizable reconfigurable computing platform, with a parallel full-duplex switched communication network, and a software execution model to redefine the computation, communication and reconfiguration paradigms in high performance embedded systems. High Performance Embedded Computing (HPEC) applications are becoming highly sophisticated and resource consuming for three reasons. First, they should capture and process real-time data from several I/O sources in parallel. Second, they should adapt their functionalities according to the application or environment variations within given Size Weight and Power (SWaP) constraints. Third, since they process several parallel I/O sources, applications are often distributed on multiple computing nodes making them highly parallel. Due to the hardware parallelism and I/O bandwidth offered by Field Programmable Gate Arrays (FPGAs), application can be duplicated several times to process parallel I/Os, making Single Program Multiple Data (SPMD) the favorite execution model for designers implementing parallel architectures on FPGAs. Furthermore Dynamic Partial Reconfiguration (DPR) feature allows efficient reuse of limited hardware resources, making FPGA a highly attractive solution for such applications. The problem with current HPEC systems is that, they are usually built to meet the needs of a specific application, i.e., lacks flexibility to upgrade the system or reuse existing hardware resources. On the other hand, applications that run on such hardware architectures are constantly being upgraded. Thus there is a real need for flexible and scalable hardware architectures and parallel execution models in order to easily upgrade the system and reuse hardware resources within acceptable time bounds. Thus these applications face challenges such as obsolescence, hardware redesign cost, sequential and slow reconfiguration, and wastage of computing power.

Addressing the challenges described above, we propose an architecture that allows the customization of computing nodes (FPGAs), broadcast of data (I/O, bitstreams) and reconfiguration several or a subset of computing nodes in parallel. The software environment leverages the potential of the hardware switch, to provide support for the SPMD execution model. Finally, in order to demonstrate the benefits of our architecture, we have implemented a scalable distributed secure H.264 encoding application along with several avionic communication protocols for data and control transfers between the nodes. We have used a FMC based high-speed serial Front Panel Data Port (sFPDP) data acquisition protocol to capture, encode and encrypt RAW video streams. The system has been implemented on 3 different FPGAs, respecting the SPMD execution model. In addition, we have also implemented modular I/Os by swapping I/O protocols dynamically when required by the system. We have thus demonstrated a scalable and flexible architecture and a parallel runtime reconfiguration model in order to manage several parallel input video sources. These results represent a conceptual proof of a massively parallel dynamically reconfigurable next generation embedded computers [16] [15]. The PhD of Venkatasubramanian Viswanathan has been defended in the 12th of october 2015 .

6.2. Language-Parametric Formal Methods

The HoMade assembly language is still evolving. Thus, our research in formal methods for programming languages kept the language-parametric nature that we decided upon when we started the project. The techniques and tools developed here will be instantiated on the HoMade assembly when it stabilizes. Our results are also applicable to general programming languages in order to target a broader audience.

In 2015 we have consolidated the results obtained in previous years, by making them more generally available and publishing them in high-end venues.

6.2.1. Language Definitions as Rewrite Theories

In [9] we study the foundations of $\mathbb{K}$, a formal framework for defining operational semantics of programming languages. The $\mathbb{K}$-Maude compiler translates $\mathbb{K}$ language definitions to Maude rewrite theories. The compiler enables program execution by using the Maude rewrite engine with the compiled definitions, and program analysis by using various Maude analysis tools. $\mathbb{K}$ supports symbolic execution in Maude by means of an automatic transformation of language definitions. The transformed definition is called the *symbolic extension* of the original definition. In this paper we investigate the theoretical relationship between $\mathbb{K}$ language definitions and their Maude translations, between symbolic extensions of $\mathbb{K}$ definitions and their Maude translations, and how the relationship between $\mathbb{K}$ definitions and their symbolic extensions is reflected on their respective representations in Maude. In particular, the results show how analysis performed with Maude tools can be formally lifted up to the original language definitions.

6.2.2. A Generic Framework for Symbolic Execution: Theory and Applications

In [10], [17] we propose a language-independent symbolic execution framework. The approach is parameterised by a language definition, which consists of a signature for the language's syntax and execution infrastructure, a model interpreting the signature, and rewrite rules for the language's operational semantics. Then, symbolic execution amounts to computing symbolic paths using a *derivative* operation. We prove that the symbolic execution thus defined has the properties naturally expected from it, meaning that the feasible symbolic executions of a program and the concrete executions of the same program mutually simulate each other. We also show how a coinduction-based extension of symbolic execution can be used for the deductive verification of programs. We show how the proposed symbolic-execution approach, and the coinductive verification technique based on it, can be seamlessly implemented in language definition frameworks based on rewriting such as the $\mathbb{K}$ framework. A prototype implementation of our approach has been developed in $\mathbb{K}$. We illustrate it on the symbolic analysis and deductive verification of nontrivial programs.

6.2.3. Symbolic Execution by Language Transformation

In [2] we propose a language-independent symbolic execution framework for languages endowed with a formal operational semantics based on term rewriting. Starting from a given definition of a language, a new language definition is generated, with the same syntax as the original one, but whose semantical rules are transformed in order to rewrite over logical formulas denoting possibly infinite sets of program states. Then, the symbolic execution of concrete programs is, by definition, the execution of the same programs with the symbolic semantics. We prove that the symbolic execution thus defined has the properties naturally expected from it (with respect to concrete program execution). A prototype implementation of our approach was developed in the K Framework. We demonstrate the tool's genericity by instantiating it on several languages, and illustrate it on the reachability analysis and model checking of several programs.

6.2.4. Program Equivalence by Circular Reasoning

In [7] we propose a logic and a deductive system for stating and automatically proving the equivalence of programs written in languages having a rewriting-based operational semantics. The chosen equivalence is parametric in a so-called observation relation, and it says that two programs satisfying the observation relation will inevitably be, in the future, in the observation relation again. This notion of equivalence generalises several well-known equivalences and is appropriate for deterministic (or, at least, for confluent) programs. The deductive system is circular in nature and is proved sound and weakly complete; together, these results say that, when it terminates, our system correctly solves the given program-equivalence problem. We show that our approach is suitable for proving equivalence for terminating and non-terminating programs as well as for concrete and symbolic programs. The latter are programs in which some statements or expressions are symbolic variables. By proving the equivalence between symbolic programs, one proves the equivalence of (infinitely) many concrete programs obtained by replacing the variables by concrete statements or expressions. The approach is illustrated by proving program equivalence in two languages from different programming paradigms. The examples in the paper, as well as other examples, can be checked using an online tool.

6.2.5. Verifying Reachability-Logic Properties on Rewriting-Logic Specifications

Rewriting Logic is a simply, flexible, and powerful framework for specifying and analysing concurrent systems. Reachability Logic is a recently introduced formalism, which is currently used for defining the operational semantics of programming languages and for stating properties about program executions. Reachability Logic has its roots in a wider-spectrum framework, namely, in Rewriting Logic Semantics. In the invited paper [10] we show how Reachability Logic can be adapted for stating properties of transition systems described by Rewriting-Logic specifications. We propose a procedure for verifying Rewriting-Logic specifications against Reachability-Logic properties. We prove the soundness of the procedure and illustrate it by verifying a communication protocol specified in Maude.

6.2.6. A Theoretical Foundation for Programming Languages Aggregation

This work was published as [11]. Programming languages should be formally specified in order to reason about programs written in them. We show that, given two formally specified programming languages, it is possible to construct the formal semantics of an aggregated language, in which programs consist of pairs of programs from the initial languages. The construction is based on algebraic techniques and it can be used to reduce relational properties (such as equivalence of programs) to reachability properties (in the aggregated language).

6.3. The SCAC Model : a weakly-coupled execution model for MPSoC

Synchronous Communication Asynchronous Computation (SCAC) is an execution model that separates the execution of communication phases from those of computation in order to facilitate their overlapping, thus covering the data transfer time. To allow the simultaneous execution of these two phases, we propose an approach based on three levels : two globally-centralized/locally-distributed hierarchical control levels and a parallel computation level.

G-MPSoC [5] is a SCAC System-on-Chip implementation based on a grid of clusters of Hardware and Software Computation Elements with different size, performance, and complexity. It is composed of parametric IP-reused modules: processor, controller, accelerator, memory, interconnection network, etc. to build different architecture configurations. The generic structure of G-MPSoC facilitates its adaptation to the intensive signal processing applications requirements.

The communication phase in SCAC System-on-Chip should be as fast as possible to avoid compromising parallel computing, using small and low power consumption modules to facilitate the interconnection network extensibility in a scalable system. To meet these criteria and based on a module reuse methodology, we chose to integrate a reconfigurable SCAC-Net [14] interconnection network to communicate data in our system. The SCAC-Net network is composed of communication modules as the number of the nodes used by the system. Using generic parameters, the topology of SCAC-Net network can be easily configured according to the needed communication which give more flexibility to the system.

POSTALE Team

6. New Results

6.1. Parallel light speed labeling: the world's fastest connected component labeling for multicore processors

Participants: Lionel Lacassagne, Laurent Cabaret, Daniel Etiemble.

We have designed a parallel version of the Light Speed Labeling for shared-memory multicore processor. This algorithm outperforms the best algorithm by a factor x10. We are now working on the design of algorithms for GPU and manycore embedded processor and especially the TSAR architecture of LIP6 laboratory. More information is available at

- TSAR architecture: https://www-soc.lip6.fr/trac/tsar
- ALMOS operating system: https://www-soc.lip6.fr/trac/almos
- GIET-VM system: https://www-soc.lip6.fr/trac/giet-vm

The paper [20] introduces the parallel version of the Light Speed Labeling (LSL) and compares it with the parallel versions of the competitors. A benchmark shows that the parallel Light Speed Labeling is at least ×1.9 faster than all the other algo- rithms for random images. This factor reach ×3.6 for structured random images. More important, we show that thanks to its run-based processing (segments), LSL is intrinsically more efficient than all pixel-based algorithms.

6.2. Opening Polyhedral Compiler's Black Box

Participants: Lénaïc Bagnères, Oleksandr Zinenko, Stéphane Huot, Cédric Bastoul.

While compilers offer a fair trade-off between productivity and executable performance in single-threaded execution, their optimizations remain fragile when addressing compute-intensive code for parallel architectures with deep memory hierarchies. Moreover, these optimizations operate as black boxes, impenetrable for the user, leaving them with no alternative to time-consuming and error-prone manual optimization in cases where an imprecise cost model or a weak analysis resulted in a bad optimization decision. To address this issue, we propose a technique allowing to automatically translate an arbitrary polyhedral optimization, used internally by loop-level optimization frameworks of several modern compilers, into a sequence of comprehensible syntactic transformations as long as this optimization focuses on scheduling loop iterations. With our approach, we open the black box of the polyhedral frameworks enabling users to examine, refine, replay and even design complex optimizations semi-automatically in partnership with the compiler. [17]

6.3. Automating Resource Selection and Configuration in Inter-clouds through a Software Product Line Method

Participants: Alexandro Ferreira Leite, Vladimir Castro Alves, Genaina Nunes Rodrigues, Claude Tadonki, Christine Eisenbeis, Alba Cristina Alves de Melo.

Nowadays, cloud users face three important problems: (a) choosing one or more appropriate cloud provider(s) to run their application(s), (b) selecting appropriate cloud resources, which implies having enough information about the available resources, including their characteristics and constraints, and (c) configuring the cloud resources. These problems are mostly due to the wide range of resources. These resources usually have distinct dependencies, and they are offered at various clouds' layers. In this complex scenario, the users often have to handle cloud resources and their dependencies manually. This is an error-prone and time-consuming activity, even for skilled cloud users and system administrators. In this context, this paper proposes a software product line engineering (SPLE) method and a tool to deal with these issues. Our SPL-based engineering method enables a declarative and goal-oriented strategy. Furthermore, it allows resource selection and configuration in inter-cloud environments. In our proposal, the cloud users specify their applications and requirements, and our tool automatically selects and configures a suitable computing environment, taking into account temporal and functional dependencies. Experimental results on Amazon EC2 and Google Compute Engine (GCE) show that our approach enables unskilled users to have access to advanced inter-cloud computing configurations, without being concerned with the characteristics of each cloud. [18]

6.4. A Randomized LU-based Solver Using GPU and Intel Xeon Phi Accelerators

Participants: Marc Baboulin, Amal Khabou, Adrien Rémy de Zotti.

We present a fast hybrid solver for dense linear systems based on LU factorization. To achieve good performance, we avoid pivoting by using random butterfly transformations for which we developed efficient implementations on heterogeneous architectures. We used both Graphics Processing Units and Intel Xeon Phi as accelerators. The performance results show that the pre-processing due to randomization is negligible and that the solver outperforms the corresponding routines based on partial pivoting. [16]

6.5. Metaprogramming dense linear algebra solvers. Applications to multi and many-core architectures

Participants: Ian Masliah, Marc Baboulin, Joël Falcou.

The increasing complexity of new parallel architectures has widened the gap between adaptability and efficiency of the codes. As high performance numerical libraries tend to focus more on performance, we wish to address this issue using a C++ library called NT2. By analyzing the properties of the linear algebra domain that can be extracted from numerical libraries and combining them with architectural features, we developed a generic approach to solve dense linear systems on various architectures including CPU and GPU. We have then extended our work with an example of a least squares solver based on semi-normal equations in mixed precision that cannot be found in current libraries. For the automatically generated solvers, we report performance comparisons with state-of-the-art codes, and show that it is possible to obtain a generic code with a high-level interface (similar to MATLAB) which runs either on CPU or GPU without generating a significant overhead. [21] [23]

6.6. Using Random Butterfly Transformations in Parallel Schur Complement-Based Preconditioning

Participants: Marc Baboulin, Aygul Jamal, Masha Sosonkina.

We propose to use a randomization technique based on Random Butterfly Transformations (RBT) in the Algebraic Recursive Multilevel Solver (ARMS) to improve the preconditioning phase in the iterative solution of sparse linear systems. We integrated the RBT technique into the parallel version of ARMS (pARMS). The preliminary experimental results on some matrices from the Davis' collection show an improvement of the convergence and accuracy of the results when compared with existing implementations of the pARMS preconditioner. [15]

6.7. LU Preconditioning for Overdetermined Sparse Least Squares Problems

Participants: Gary Howell, Marc Baboulin.

We investigate how to use an LU factorization with the classical LSQR routine for solving overdetermined sparse least squares problems. Usually L is much better conditioned than A and iterating with L instead of A results in faster convergence. When a runtime test indicates that L is not sufficiently well-conditioned, a partial orthogonalization of L accelerates the convergence. Numerical experiments illustrate the good behavior of our algorithm in terms of storage and convergence. [19]

6.8. Dense Symmetric Indefinite Factorization on GPU Accelerated Architectures

Participants: Marc Baboulin, Jack Dongarra, Adrien Rémy de Zotti, Stanimire Tomov, Ichitaro Yamazaki.

We study the performance of dense symmetric indefinite factorizations (Bunch-Kaufman and Aasen's algorithms) on multicore CPUs with a Graphics Processing Unit (GPU). Though such algorithms are needed in many scientific and engineering simulations, obtaining high performance of the factorization on the GPU is difficult because the pivoting that is required to ensure the numerical stability of the factorization leads to frequent synchronizations and irregular data accesses. As a result, until recently, there has not been any implementation of these algorithms on hybrid CPU/GPU architectures. To improve their performance on the hybrid architecture, we explore different techniques to reduce the expensive communication and synchronization between the CPU and GPU, or on the GPU. We also study the performance of a symmetric indefinite factorization with no pivoting combined with the preprocessing technique based on Random Butterfly Transformations. Though such transformations only have probabilistic results on the numerical stability, they avoid the pivoting and obtain a great performance on the GPU. [14]

6.9. Computing least squares condition numbers on hybrid multicore/GPU systems

Participants: Marc Baboulin, Jack Dongarra, Rémi Lacroix.

We present an efficient computation for least squares conditioning or estimates of it. We propose performance results using new routines on top of the multicore-GPU library MAGMA. This set of routines is based on an efficient computation of the variance-covariance matrix for which, to our knowledge, there is no implementation in current public domain libraries LAPACK and ScaLAPACK. [22]

6.10. Towards a High-Performance Tensor Algebra Package for Accelerators

Participants: Marc Baboulin, Veselin Dobrev, Jack Dongarra, Christopher Earl, Joël Falcou, Azzam Haidar, Ian Karlin, Tzanio Kolev, Ian Masliah, Stanimire Tomov.

Numerous important applications, e.g., high-order FEM simulations, can be expressed through tensors. Examples are computation of FE matrices and SpMV products expressed as generalized tensor contractions. Contractions by the first index can often be represented as tensor index reordering plus gemm, which is a key factor to achieve high-performance. We present ongoing work on the design of a high-performance package in MAGMA for Tensor algebra that includes techniques to organize tensor contractions, data storage, and parametrization related to batched execution of large number of small tensor contractions. We apply autotuning and code generation techniques to provide an architecture-aware, user-friendly interface. [24]

TASC Project-Team

7. New Results

7.1. IBEX

The development of the Ibex library has continued. The main developments in 2015 are:

- the complete refactoring of the multi-heap internal structure used for search space exploration in the global optimizer
- the creation of a new module for explicit set (or pavings) manipulation/algebra with full documentation and tutorial

7.2. NetWMS2

New advances have been made in the context of packing curved objects. The packing algorithm developed in 2014 have been published in ICJAI'15, along with new features. The calculation of the *penetration depth* (a classical measure of violation cost for overlapping objects) has also been extended to the case of parametric curves (like, e.g., Bezier curves) and new experiments have been conducted with our solver for this new type of objects.

We deal with the problem of packing two-dimensional objects of quite arbitrary shapes including in particular curved shapes (like ellipses) and assemblies of them. This problem arises in industry for the packaging and transport of bulky objects which are not individually packed into boxes, like car spare parts. There has been considerable work on packing curved objects but, most of the time, with specific shapes; one famous example being the circle packing problem. There is much less algorithm for the general case where different shapes can be mixed together. A successful approach has been proposed recently by Martinez et al. and the algorithm we propose here is an extension of their work. Martinez et al. use a stochastic optimization algorithm with a fitness function that gives a violation cost and equals zero when objects are all packed. Their main idea is to define this function as a sum of $\binom{n}{2}$ elementary functions that measure the overlapping between each pair of different objects. However, these functions are ad-hoc formulas. Designing ad-hoc formulas for every possible combination of object shapes can be a very tedious task, which dramatically limits the applicability of their approach. We generalize the approach by replacing the ad-hoc formulas with a numerical algorithm that automatically measures the overlapping between two objects. Then, we come up with a fully black-box packing algorithm that accept any kind of objects.

7.3. Time-Series Constraints

We describe a large family of constraints for structural time series by means of function composition. These constraints are on aggregations of features of patterns that occur in a time series, such as the number of its peaks, or the range of its steepest ascent. The patterns and features are usually linked to physical properties of the time series generator, which are important to capture in a constraint model of the system, i.e. a conjunction of constraints that produces similar time series. We formalise the patterns using finite transducers, whose output alphabet corresponds to semantic values that precisely describe the steps for identifying the occurrences of a pattern. Based on that description, we automatically synthesise automata with accumulators, as well as constraint checkers. The description scheme not only unifies the structure of the existing 30 time-series constraints in the Global Constraint Catalogue, but also leads to over 600 new constraints, with more than 100,000 lines of synthesised code.

7.4. New Global Constraints

This year we introduce new generic global constraints that can be respectively used to reformulate a number of constraints where the formulation become easy once some tuples are sorted, and to express temporal relation between two sequence of intervals.

- Some constraint programming solvers and constraint modelling languages feature the $\mathit{sort}(L, P, S)$ constraint, which holds if S is a nondecreasing rearrangement of the list L, the permutation being made explicit by the optional list P. However, such sortedness constraints do not seem to be used much in practice. We argue that reasons for this neglect are that it is impossible to require the underlying sort to be stable, so that *sort* cannot be guaranteed to be a total-function constraint, and that L cannot contain tuples of variables, some of which form the key for the sort. To overcome these limitations, we introduce the *stable-keysort* constraint, decompose it using existing constraints, and propose a propagator. This new constraint enables a powerful modelling idiom, which we illustrate by elegant and scalable models of two problems that are otherwise hard to encode as constraint programs.
- The constraint was initially motivated by an application where the objective is to generate a video summary, built using intervals extracted from a video source. In this application, the constraints used to select the relevant pieces of intervals are based on Allen's algebra. The best state-of-the-art results are obtained with a small set of ad hoc solution techniques, each specific to one combination of the 13 Allen's relations. Such techniques require some expertise in Constraint Programming. This is a critical issue for video specialists. We design a generic constraint, dedicated to a class of temporal problems that covers this case study, among others. ExistAllen takes as arguments a vector of tasks, a set of disjoint intervals and any of the 213 combinations of Allen's relations. ExistAllen holds if and only if the tasks are ordered according to their indexes and for any task at least one relation is satisfied, between the task and at least one interval. We design a propagator that achieves bound-consistency in O(n+m), where n is the number of tasks and m the number of intervals. This propagator is suited to any combination of Allen's relations, without any specific tuning. Therefore, using our framework does not require a strong expertise in Constraint Programming. The experiments, performed on real data, confirm the relevance of our approach.

7.5. Controlling the Generation of Solutions

The following two results deal with controlling the generation of solutions to a constraint problem.

- The *focus* constraint expresses the notion that solutions are concentrated. In practice, this constraint suffers from the rigidity of its semantics. To tackle this issue, we propose three generalizations of the FOCUS constraint. We provide for each one a complete filtering algorithm. Moreover, we propose mathematical programming (ILP) and constraint programming decompositions.
- There are significant motivations for considering alternate solutions to a problem. As expressed by renowned statistician George Box *The most that can be expected from any model is that it can supply a useful approximation to reality: all models are wrong; some models are useful.*. Multiple solutions alone, however, are not sufficient to guarantee anything of value. If they are nearly identical nothing is gained. While most frameworks in the literature consider diversity between solutions through mathematical distances, this paper proposes alternative distance measures represented by global constraints. It introduces a constraint programming framework for optimization problems, able to generate sets of nearly-optimal solutions that are diverse. With respect to over-constrained problems, the framework can be specialized in order to generate solution sets where constraint violations are diverse.

AOSTE Project-Team

7. New Results

7.1. CCSL as a Logical Clock Calculus Algebra: expressiveness and decidability results

Participants: Robert de Simone, Julien Deantoni, Frédéric Mallet, Qingguo Xu.

CCSL is a language dedicated to the expression of time constraints, based on so-called logical clocks. Its declarative nature is akin to the Lustre or (even closer to) the Signal language, but without values (to clock/event occurrences) and with both synchronous and asynchronous constraints. Solving a set of CCSL constraints amounts to the production of a feasible schedule of the system. While the TimeSquare tool may attempt to generate such a schedule trace by insightful simulation, it is not guaranteed to be complete in its search.So the issue of expressiveness and decidability was left open to this day.

Still, in previous years, we had established the CCSL constraints could be translated into parallel products (extended, transition-labelled) Büchi machines, but some of these machines had to contain integer shift counters, and were thus not fully FSMs. Our (misled) conjecture that CCSL had semilinear, Presburger-arithmetic power was defeated by a new translation expressing (unitary then general) Petri Nets and Vector Addition Systems into CCSL by encoding. The new conjecture that CCSL was then as powerful as Petri Nets was again defeated by a construction interpreting the features of *inhibitor arcs* in CCSL. As such inhibitor arcs extend the expressive power of Petri Nets to become universal (Turing-complete), CCSL enjoys the same universal property (which makes it unfortunately impossible to solve automatically in general).

Despite this negative result we could show that, under natural restrictions such as the assumption that "input" clocks have bounded jitter around a mean rate, and even if those bounds are not exactly known (but may be used as a parameter), then expressiveness remains in the semi-linear, Presburger-arithmetic range.

As a side-effect of this work we provided the translation of CCSL constraints into Büchi components by using a well-defined fragment of the Esterel syntax to express the Buchi automata.

Preliminary results are exposed in a research report. A much more ambitious article is in preparation.

As part of Professor Xu sabbatical in Aoste, we also considered the topic of machine-assisted proof of schedulability using theorem-provers (in our case PVS) [54]).

7.2. Industrial design flow for Embedded System Engineering

Participants: Julien Deantoni, Frédéric Mallet, Marie Agnes Peraldi Frati, Robert de Simone, Ales Mishchenko.

As part of the PIA LEOC Clarity collaborative project we attempt to instill some of our theoretical and methodological ideas into the framework of the (open-source, Polarsys Eclipse) Capella environment. This environment was initially developed inside Thales, under the name ARCADIA/Melody, as a modeling tool flow for System-Level Design in-the-large. As such, several aspects were not fully considered, specially those regarding safe sound simulation semantics at this level, or the role of states and modes in variability regarding both the software applicative and hardware architectural platform models. This research is in part motivated by concrete needs as expressed by end-users such as Airbus, Areva/EDF and Thales.
Results on methodological enhancements are described

7.3. Coordination of heterogeneous Models of Computation as Domain-Specific Languages

Participants: Matias Vara Larsen, Julien Deantoni, Frédéric Mallet.

In the context of the collaborative ANR GEMOC project (9.2.1.2 , we investigated the way the multiview approach generally promoted in Aoste could deal with analysis and simulation of systems specified using multiple heterogeneous languages. Coordinated use of heterogeneous domain specific languages (DSL) led to so-called globalization of modeling language. We wrote a chapter related to these concerns [50], as part of a book dedicated to the challenges of the field, gathering industrial and academic contributors.

This goal was achieved in two steps. First step consisted in specifying a language able to support appropriate information (*i.e.,* the one required for the coordination) in a *Language Behavioral Interface (LBI).* Second step consisted in using the LBI to define coordination patterns from which the coordination of models can be automatically inferred. Design is supported by an heterogeneous simulation engine that has been developed and integrated in the Gemoc studio environment. Gemoc Studio, enhanced with our new research ideas, won the 9$^{\text{th}}$ execution tool contest at ...

We also developed MoCCML (Model of Concurrency and Communication Modeling Language), an imperative extension of the CCSL language in the form of constraint automata [28]. MoCCML defines the concurrent and communication part of the semantics of a language, and is used by the LBI to exhibit internal causalities and synchronizations. Finally, we defined a protocol combining the concurrency aspects and the execution functions (*i.e.,* the rewriting rules) so as to be able to develop, in a modular way, the whole behavioral semantics of a language [30], [31].

Our work this on coordination of heterogeneous languages produced two major results. The first one is the development of BCOoL (Behavioral Coordination Operator Language [33]). BCOoL is a language dedicated to the specification of coordination patterns between heterogeneous languages. It comes with a tool chain allowing the generation of the coordination given a BCOoL operator and specific models. Our second result is the development of an heterogeneous execution engine, integrated to Gemoc studio, to run conjointly different models [44]. Both works were mainly realized by Matias Vara Larsen, as part of his upcoming PhD.

7.4. SoC multiview (meta)modeling for performance, power, and thermal aspects

Participants: Amani Khecharem, Robert de Simone, Emilien Kofman, Julien Deantoni.

In the framework of the ANR HOPE project we progressed the definition of multiview metamodels for the design of Systems-on-Chip) (SoC systems integrating performance, power and thermal aspects. The main concern was to stress regularity and commonality between those views, each developed on "domains" defined as partitions of the original block diagram (clock domains, voltage domains, floorplans,...), and with finite state machine controlers setting the levels of these domains; links between distinct views are originally provided by laws of physics, but then usually identified on discrete allowable values by engineers. The application view, meant to provide typical use-cases to help dimension the SoC platform by abstract simulation, also fits in this framework. This methodological work was presented in the local forum SAME (Sophia-Antipolis MicroElectronices) [53]. It is supposed to work in two ways, both by allowing the appplication of analytic methods to compute an optimized mapping of application tasks onto platform resources, and then to translate these results towards sophisticated simulation environments (such as MCO Platform Architect by Synopsys or ACEplorer by Docea Power/Intel, both partners in the HOPE consortium) which consider non-functional aspects of power and thermal modeling in their simulation environments. The various approaches considered in Aoste to define mapping constraints and solve them algorithmically are presented elsewhere. All this should sonn be rported in Ameni Khacharem PhD document.

7.5. Networks-on-Board: between NoCs and rack connector buses

Participants: Amine Oueslati, Robert de Simone, Albert Savary, Emilien Kofman.

The recent paradigm of Massively Parallel Processor Arrays (MPPA), or more generally manycore Systems-on-Chip, rely on the existence of a high-throughput on-Chip Network (NoC) to interconnect the various cores and processing clusters. Despite its benefits, it requires that all components are put on the same dye, and thus designed monolithically. On the other end, supercomputers are built by assembling racks or blades of processors, connected by fast buses (fast ethernet or infinyband usually), with low predictivity of throughput. A third, intermediate path is explored in the context of the FUI Clistine project, based on a notion of Network-on-Board (or Network-in-Package), aiming at the benefits of NoCs brought to the level of a single PCB board, where the various components can be assembled in a modular fashion. We consider the applicationof our previous expertise on modeling and analysis of NoC-based architecture, with their implications on the optimized mapping of dataflow models of applications onto such interconnects, to adapt them in this new context. The objective is to consider alternative network topologies, and to transate optimal mappings into the concrete network operations on a prototype implementation realized by SynergieCAD, the company heading the project. This topic reflects the PhD thesis of Amine Oueslati, and the engineering work of Albert Savary.

7.6. Solving AAA constraints analytically

Participants: Emilien Kofman, Dumitru Potop Butucaru, Thomas Carle, Raul Gorcitz, Robert de Simone, Mohamed Bergach, Amine Oueslati.

Given two abstract modeling descriptions, one of a dataflow process network for the application, one of a block diagram structure for the computing platform and its interconnects, together with cost functions for the elementary compuations and communications, one is bound to seek optimal mappings pairing the two. Amongst all the possible techniques, an obvious one consists in solving constraint using general solvers (real, integer, or boolean constraint programming, SMT solvers, etc). Given the NP-hard nature of the problem, the issue here is to scale to the dimensions of realistic problems. We conducted extensive experiments on several case studies, with as extra objective the concern of studying how the formulation of constraints, or the exploiation of additional information (in concurency or exclusion of tasks, structural symmetries,...) could impact favorably or negatively the process. Results were compiled in a publication [57].

In the framework of the PhD thesis of Mohamed Bergach, under CIFRE funding with Kontron Toulon, we studied how to adjust a radar application, that typically computes extensively FFT convolutions, on an hybrid CPU/GPU architecture such as IntelCore IvyBridge and Haswell processors. This approach works in two stages: first we considered how to implement a FFT redex as large as possible in exactly one core (either a CPU core or a GPU Execution Unit), so as to make full use of the local register memories and SIMD/vectorial instructions. Not by accident certainly FFT blocks of size exactly 8 and 16 respectively can so be fitted on a GPU (resp. CPU) block. This provides a new "compound" instruction, on which to build modularly and optimization the allocation of larger aplications based on such basic block. This is fully described in Mohamed Bergach PhD document [16].

7.7. Stochastic extension of MARTE/CCSL for CPS modeling

Participant: Frédéric Mallet.

This work was conducted during the sabbatical period of Frédéric Mallet at ECNU Shanghai, in the context of the associated team FM4CPS (9.4.1.1).

As a declarative language, CCSL allows the specification of causal and temporal properties of systems expressed as constraints in a specific syntax. While each constraint reduces the set of possible behaviors, there may still be multiple (schedule) solutions, or none at all. When several solutions remain feasible, our TimeSquare tool allows to set up a resolving policy, to choose whether we want to attempt exploring exhaustively all these solutions, or else narrow the solution space according to an auxiliary criterion.

The extension of CCSL with stochastic features and probabilistic information is meant to help provide such an additional criterion, while modeling temporal constraints on the environment which are not necessarily well-known or controllable, specially in the domain of Cyber-Physical Systems. Then, such features should help reducing the set of possible behaviors, narrowing for instance to the most likely ones (in a formal quantitative meaning).

We are currently relying on UPPAAL SMC (Stochastic Model-Checking) toolset as prototype analyzer for the resulting specifications.

7.8. Coupling SystemC and FMI for co-simulation of Cyber-Physical Systems

Participants: Stefano Centomo, Julien Deantoni, Robert de Simone.

In the context of Stefano Centoma master internship, and in collaboration with his global supervisor Professor Davide Quaglia, from the University of Verona, we considered the possibility to build heterogenous, multi-physics co-simulation schemes for hybrid continuous-discrete Cyber-Physical systems. The first step consisted in extracting relevant interface information from IP component described in the SystemC language; it was naturally inspired from some of our former work. But currently IP-XACT is meant to address easy component assembly at the *structural* (static) level, and is not concerned with dynamical aspects of behavior simulation. This extension, and the proper combination with the FMI standard for its purpose, allowing hybrid and multiform co-simulation of SystemC components (and also others describing the continuous physical environment) are the next-step objective being currently tacked.

7.9. Code generation for time-triggered platforms based on Real-Time Scheduling

Participants: Dumitru Potop Butucaru, Raul Gorcitz, Yves Sorel.

We have continued this year the work on real-time scheduling and code generation for time-triggered platforms. Much of this work was carried out as part of a trilateral collaboration with Airbus DS and the CNES, which have funded an (onerous) TTEthernet-based test platform and partly funded the post-doctorate of Raul Gorcitz. The remainder of Raul Gorcitz' post-doc has been funded by the ITEA3 Assume project.

This year, the objective has been to allow code generation on an industry-grade platform comprising ARINC 653-based computers connected through a TTEthernet network. The novelty with respect to previous years comes from the time-triggered TTEthernet network, whose scheduling properties raise new problems. Unlike in classical field buses, resource reservation in a TTEthernet network is done at the level of directed links (physical wires that connect routers and end stations). Each of these links is controlled by an arbiter that determines the scheduling of both time-triggered data transfers and control messages needed to ensure the global time synchronization. This year we have built a model of the TTEthernet network allowing precise real-time scheduling, and worked on code generation aspects. We expect to have a fully running prototype in the next 2 months, and to demonstrate it to our funders. Relevant publications are [18], [38].

For teaching purposes and to achieve a finer understanding of ARINC 653-based operating systems, we have also developed an implementation of the standard on inexpensive RaspberryPi platforms, and published a scientific vulgarization paper [55].

7.10. Real-time systems compilation

Participants: Dumitru Potop Butucaru, Keryan Didier, Mihail Asavoae.

This research line develops over various results of the team over the years, its aim being to develop fully automatic implementation flows going fully automatically from functional and non-functional specification to correct and efficient running implementation. We advocate for a real-time systems compilation approach that combines aspects of both real-time scheduling and compilation of both classical and synchronous languages. Like a classical compiler such as GCC, a real-time systems compiler should use fast and efficient scheduling and code generation heuristics, to ensure scalability. Similarly, it should provide traceability support under the form of informative error messages enabling an incremental trial-and-error design style, much like that of classical application software. This is more difficult than in a classical compiler, given the complexity of the transformation flow (creation of tasks, allocation, scheduling, synthesis of communication and synchronization code, etc.), and requires a full formal integration along the whole flow, including the crucial issue of correct hardware abstraction. A real-time systems compiler should perform precise, conservative timing accounting along the whole scheduling and code generation flow, allowing it to produce safe and tight real-time guarantees. More generally, and unlike in classical compilers, the allocation and scheduling algorithms must take into account a variety of non-functional requirements, such as real-time constraints, criticality, partitioning, preemptability, allocation constraints, etc. As the accent is put on the respect of requirements (as opposed to optimization of a metric, like in classical compilation), resulting scheduling problems are quite different.

We are currently building such a real-time systems compiler, called Lopht. The construction of the Lopht tool, which takes into account complex functional and non-functional specifications is discussed in the corresponding section and in [17].

This year, we have initiated work on two fundamental topics. The first one is sound architecture abstraction – ensuring that the platform models used for real-time scheduling and code generation are conservative abstractions of the real hardware and basic software, allowing the generation of implementations that are functionally and non-functionally correct. This work is performed in the framework of the LEOC Capacites project, which funds the post-doc of Mihail Asavoae. The second line of work aims at formally proving that the output of Lopht is correct with respect to its input models (including functional specification and platform model). This work is performed in the ITEA3 Assume project, which funds the PhD thesis of Keryan Didier. Together with the Parkas team-project we have also considered the implementation of mixed-criticality systems [26].

7.11. Uniprocessor Real-Time Scheduling

Participants: Mamadou Diallo, Yves Sorel, Walid Talaboulma, Robert Davis.

In the context of the master internship of Mamadou Diallo we implemented the offline time trigger scheduler proposed in his PhD thesis by Falou Ndoye on a development board based on an ARM Cortex M4. We used this ARM version since it is better suited to embedded systems, since more predictable, than the ARM 7 we used last year. Especially, it allows to determine more accurately the cost of the scheduler and of the preemptions we use in our offline schedulability analysis. We remind that the schedulability analysis provides a scheduling table which is exploited by the scheduler during the real-time execution of the tasks. This approach allows a low and fixed cost for the scheduler and the preemptions whereas these costs are variable in the case of classical online schedulers. For several task sets we compared the timing diagrams predicted by the schedulability analysis with the real-time timing diagrams measured on the ARM Cortex M4. It turns out that those timings are very close, as expected.

A new direction opened with the arrival of Rob Davis was to consider by studying the impact of the non-preemptivity constraints on the optimality of the schedulers [37], or by considering fixed priorities while scheduling messages in the context of Control Area Networks [36].

7.12. Multiprocessor Real-Time Scheduling

Participants: Aderraouf Benyahia, Laurent George, Salah Eddine Saidi, Yves Sorel, Robert Davis, Liliana Cucu.

In the context of the PhD thesis of Salah Eddine Saidi we considered the co-simulation of several process models specified in continuous time and several controllers models specified in discrete time according to a real-time hardware in the loop approach. These models specified with different tools such as Simulink, AMEsim, Modelica, etc., cooperate according to the FMI standard. They are translated in a dataflow graph that is compliant with the conditioned repetitive dataflow model of our AAA methodology for functional specification. Each model considers the feed-through function as well as the functions which depend of the state, and the state computation itself. In order to meet the real-time constraints of such complex co-simulation we need to execute them on multicore platforms. We studied the limitations of greedy and local search distributed real-time scheduling heuristics we developped in the past for control applications. The first limitation is related to the FMI standard which requires that the functions belonging to a model are allocated to the same core. We first try to introduce additional semaphores in the real-time code generated automatically to avoid these situations. Unfortunately, this solution decreases significantly the acceleration brought by the multicore. Therefore, we started to investigate graph based techniques that add non directed edges to specify the FMI relation and search solutions where some non oriended edges can be oriented to minimize locally the makespan.

In the context of the master internship of Mamadou Diallo we studied the possibilities to extend the offline time trigger scheduler implemented on a uniprocessor to the multiprocessor case. Since the embedded board based on the ARM Cortex M4 we utilize features an ethernet interface, we conducted several experimentations on ethernet switches to measure the end-to-end communication time between several real-time tasks running on such boards with such schedulers.

We completed the work on the gateway with modeling languages for certified code generation carried out in the P FUI project 9.2.2 which ended in June 2015. Mainly, we tested the P modeling language to SynDEx gateway on four industrial use cases provided by AdaCore, Continental and Aboard Engeneering. We specified these applications with the P language and translated them in the SynDEx format. With SynDEx we analysed the schedulability and automatically generated the corresponding code for an Intel 8 cores Xeon ES-1620v2 3.70Ghz. For these applications ranging from 103 to 1403 bloks we obtained an acceleration factor equal to the number of cores.

Thanks in part to the arrival of Rob Davis, our team has participated to the proposition of a new framework in the context of multicore platforms: *Multicore Response Time Analysis framework* [34]. This proposal was made in close collaboration with academic partners such as the University of Luxembourg, Verimag and ISEP Porto. The framework is extensible to different multicore architectures, with various types and arrangements of local memory, and different arbitration policies for the common interconnects. The MRTA framework provides a general approach to timing verification for multicore systems, parametric in the hardware configuration, and so can be used architectural design stage to compare the guaranteed levels of performance that can be obtained with different hardware configurations. The MRTA framework decouples response time analysis from a reliance on context independent WCET values. Instead, the analysis specifies response times directly according to requirements on different hardware resources.

7.13. Probabilistic and statistical temporal analysis

Participants: Liliana Cucu, Robert Davis, Adriana Gogonel, Walid Talaboulma, Dorin Maxim, Cristian Maxim.

Real-time constraint guarantees require worst-case reasoning to provide sound solutions. We have proposed to define and use worst-case reasoning in different contexts: optimal scheduling algorithms, response time analysis, estimation of worst-case execution times. These results have laid the foundations for certifiable probabilistic solutions to real-time systems.

In particular, we have studied the probabilistic response time analysis for systems with multiple probabilistic parameters, either by using bounds based on real-time calculus, extreme value theory, direct calculation or in a context of component-based systems. Generally, probabilistic methods have high complexity cost; using upper-bounds for the input probability distributions we provide conservative(safe) results faster. Worst-case reasoning is also provided for the statical estimation of a task probabilistic worst-case execution time.

Results were published in [22], [24], [58], [56], [42], [46], [23], [42], [43], [40]

7.14. Parametric and Non-Parametric Statistics for Program Performance Analysis and Comparison

Participant: Sid Touati.

This research activity is a continuation of our joint research effort with Julien Worms, Assistant Professor at University of Versailles Saint-Quentin (UVSQ), dealing with statistical program performance analysis and comparison, in presence of performance variability. In the previous study (called Speedup-Test), we gave a rigorous statistical methodology for analysis of program speedups based on mean or median performance metrics: execution time, energy consumption, etc. However mean or median observed performances do not always reflect the user's feeling of performance, especially when the performances are really unstable. In the current study, we propose additional precise performance metrics, based on performance modeling using gaussian mixtures. We explore the difference between parametric and non parametric statistics applied on program performance analysis. Our additional statistical metrics for analysing and comparing program performances give to the user more precise decision tools to select best code versions, not necessarily based on mean or median numbers. Also, we provide a new metric to estimate performance variability based on gaussian mixture model. Our statistical methods are implemented in R, and distributed as open source code. A research report is under completion, before submission as article.

CONVECS Project-Team

6. New Results

6.1. New Formal Languages and their Implementations

6.1.1. Definition of LNT

Participants: Hubert Garavel, Frédéric Lang, Wendelin Serwe.

LNT is a next generation formal description language for asynchronous concurrent systems, which attempts to combine the best features of imperative programming languages and value-passing process algebras. LNT is increasingly used by CONVECS for industrial case studies and applications (see § 6.5) and serves also in university courses on concurrency, in particular at ENSIMAG (Grenoble) and at Saarland University.

In 2015, the theoretical foundations of LNT have been explored in a journal article [14] that, after examining the various ways sequential composition is handled in mainstream value-passing process calculi, shows that these various approaches are subsumed by the LNT approach, which is easier to learn and leads to more readable and more concise specifications.

The LNT language has also been enhanced in several aspects:

- The "`case`" construct now supports multiple (tuple-like) expressions and patterns.
- Two new parameter-passing modes "`in var`" and "`out var`" have been introduced to allow finer data-flow analyses.
- Exceptions are better handled and a new "`assert`" statement was added to LNT.
- The "`none`" channel is now implicitly predefined.
- Finally, the LNT reference manual has been extended and updated at many places.

6.1.2. Translation from LNT to LOTOS

Participants: Hubert Garavel, Frédéric Lang, Wendelin Serwe.

In 2015, the translator from LNT to LOTOS was further improved. In addition to 22 bug fixes and improved error messages, the following enhancements have been brought:

- The "`-root`" option of LNT2LOTOS now accepts value parameters for LNT processes and supports gate parameters in named style. It also accepts the name of a process not present in the current module.
- Negative number constants of the form "-2^{k-1}", where integer numbers are represented using k bits, are now supported.
- Better warning messages are emitted for "`in`" and "`in out`" (formerly "`inout`") parameters.

6.1.3. Translation from LOTOS to Petri nets and C

Participants: Hubert Garavel, Wendelin Serwe.

The LOTOS compilers CAESAR and CAESAR.ADT, which were once the flagship of CADP, now play a more discrete role since LNT (rather than LOTOS) has become the recommended specification language of CADP. Thus, CAESAR and CAESAR.ADT are mostly used as back-end translators for LOTOS programs automatically generated from LNT or other formalisms such as Fiacre, and are only modified when this appears to be strictly necessary.

In 2015, in addition to a few bug fixes, the "`-root`" option of the CAESAR compiler has been generalized to support processes having value parameters; this makes compositional verification easier by removing the need for introducing extra wrapper processes having no value parameters. The EXEC/CAESAR interface has been extended with two new primitives "`CAESAR_KERNEL_DELAY`" and "`CAESAR_KERNEL_EXIT()`". Also, optimizations have been introduced to generate shorter and simpler C code, and to make sure that this C code compiles without spurious warnings.

A systematic comparison between CAESAR.ADT and available interpreters/compilers for other languages that support rewrite rules or pattern matching has been undertaken. This comparison reuses the benchmarks developed for the three Rewrite Engine Competitions (2006, 2009, and 2010). As a preliminary step, we developed a tenth translators from the REC formalism in which these benchmarks are written to languages such as Haskell, LOTOS, Maude, mCRL, OCAML, Opal, Rascal, Scala, and Tom.

6.1.4. NUPN

Participants: Hubert Garavel, Frédéric Lang.

The CAESAR.BDD tool that analyzes NUPN (*Nested-Unit Petri Nets*) models and serves to prepare the yearly Model Checking Contest [0] has been enhanced in several ways. In addition to 7 bug fixes, 14 new command-line options have been added to CAESAR.BDD ("`-arcs`", "`-bits`", "`-creator`", "`-density`", "`-encodings`", "`-height`", "`-hwb`", "`-multiple-arcs`", "`-multiple-initial-tokens`", "`-places`", "`-redundant-units`", "`-transitions`", "`-units`", and "`-width`"). The output format produced by the "`-exclusive-places`" option has been revised. The "`-mcc`" option now computes the extended free choice property. A new option "`-network nupn`" was also added to EXP.OPEN to produce NUPN models from automata networks.

Particular efforts have been put to increase the scalability of CAESAR.BDD for large models. Reading large NUPN files was made much faster. The "`-exclusive-places`" option of CAESAR.BDD was made faster too. The size of the largest data structure allocated by CAESAR.BDD, has been divided by four. CAESAR.BDD has also been optimized to save memory when handling NUPN models having a simple hierarchical structure. Finally, user-specified timeouts are better supported.

A conference article was published [24], which formally defines the NUPN model and investigates its mathematical properties. Additionally, the assembly of a collection of large NUPN models was undertaken, and various prototype tools to handle NUPN models ("nupn_pack", "nupn_reduce", and "nupn_merge") have been developed.

6.1.5. Translation from GRL to LNT

Participants: Fatma Jebali, Jingyan Jourdan-Lu, Frédéric Lang, Eric Léo, Radu Mateescu.

In the context of the Bluesky project (see § 8.1.2.1), we study the formal modeling of GALS (*Globally Asynchronous, Locally Synchronous*) systems, which are composed of several synchronous subsystems evolving cyclically, each at its own pace, and communicating with each other asynchronously. Designing GALS systems is challenging due to both the high level of (synchronous and asynchronous) concurrency and the heterogeneity of computations (deterministic and nondeterministic). To bring our formal verification techniques and tools closer to the GALS paradigm, we designed a new formal language named GRL (*GALS Representation Language*), as an intermediate format between GALS models and purely asynchronous concurrent models. GRL combines the main features of synchronous dataflow programming and asynchronous process calculi into one unified language, while keeping the syntax homogeneous for better acceptance by industrial GALS designers. GRL allows a modular composition of synchronous systems (blocks), environmental constraints (environments), and asynchronous communication mechanisms (mediums), to be described at a level of abstraction that is appropriate to verification. GRL also supports external C and LNT code. A translator named GRL2LNT has been developed, allowing an LNT program to be generated from a GRL specification automatically. Additionally, an OPEN/CAESAR-compliant compiler named GRL.OPEN (based on GRL2LNT and LNT.OPEN) makes possible the on-the-fly exploration of the LTS underlying a GRL specification using CADP.

[0] http://mcc.lip6.fr/

In 2015, we have revised the GRL syntax to make GRL easier to learn and to understand. Our data base of examples has been updated to take those changes into account. We have also added some language features, such as named constants, and a dedicated construct called *activation signal* to define constraints on the asynchronous activation of blocks. This new construct is easier to use than the previous solution based on ad-hoc data signals, and semantically more elegant as it avoids unexpected deadlocks. Activation signals permit realistic situations such as halting, priorities, scenarios, and pace relations between synchronous components to be modeled at a suitable level of abstraction. They can be smoothly translated into LNT without affecting the rest of the translation.

As regards the specification of properties, to reduce the complexity of using full-fledged temporal logics, we have also proposed a property specification language dedicated to GALS systems, based upon a set of temporal logic patterns, which capture frequently encountered behaviours, encompassing both time-critical and untimed aspects of GALS systems. Those patterns include deadlock, livelock, safety, liveness, and fairness properties. The semantics of the proposed patterns have been defined by translation into the MCL language.

As regards the GRL2LNT tool, nine successive versions have been released, to take into account the syntactic changes in the GRL language, to correct about 20 bugs, to eliminate compilation warnings, and to implement the following new features:

- The generated LNT code has been corrected so as to eliminate compilation warnings and to take into account recent changes in the syntax of LNT (see § 6.1.1).
- GRL system specifications can now be parameterized with data values and instantiated using the new "`-root`" option of GRL2LNT.
- The interface between GRL and external C code has been revised in two ways: (1) external blocks with several outputs are now mapped to a single external function instead of one function per output previously, and (2) conversion functions between GRL and C numeric types have been defined, handling runtime verification of overflows. Those conversion functions have been packaged in a new code library, which is automatically included by GRL2LNT.
- Several verifications on the usage of signals and communication channels have been implemented, leading either to error messages, or to warnings corresponding to potential errors. About 20 new error messages and 10 new warnings have been added.

In addition, three manual pages have been written to document respectively the GRL language, the GRL2LNT translator tool, and the GRL.OPEN shell script. The GRL non-regression test base has been extended and now contains 230 correct examples and 400 incorrect examples.

An article describing the GRL language and its associated tools has been submitted to an international journal.

6.1.6. Translation from BPMN to LNT

Participant: Gwen Salaün.

Business processes support the modeling and the implementation of software as workflows of local and inter-process activities. Taking over structuring and composition, evolution has become a central concern in software development. We believe this should be taken into account as soon as the modeling of business processes, which can thereafter be made executable using process engines or model-to-code transformations. We advocate that business process evolution can be formally analyzed in order to compare different versions of processes, identify precisely the differences between them, and ensure the desired consistency.

To reach this objective, we developed, in collaboration with Pascal Poizat (LIP6, Paris), a model transformation from the BPMN standard notation to the LNT process algebra. We then proposed a set of relations for comparing business processes at the formal model level. With reference to related work, we proposed a richer set of comparison primitives supporting renaming, refinement, property- and context-awareness. Thanks to the implementation of a tool that integrates with the Eclipse IDE and behind-the-scene interaction with the CADP verification toolbox, we put the checking of evolution within the reach of business process designers. Our approach is fully automated and has been applied for evaluation to a large set of BPMN processes.

6.1.7. Other Language Developments

Participants: Hugues Evrard, Hubert Garavel, Frédéric Lang, Eric Léo, Wendelin Serwe.

The ability to compile and verify formal specifications with complex, user-defined operations and data structures is a key feature of the CADP toolbox since its very origins. A long-run effort has been recently undertaken to ensure a uniform treatment of types, values, and functions across all the various CADP tools.

In 2015, the connection to external software development tools has progressed. The support of the LOTOS and LNT languages in the Emacs/XEmacs, jEdit, and Vim editors has improved. More text editors are now supported, including Nano, Notepad++, and all the text editors compliant with GtkSourceView 3.0 (including the Gedit editor of Gnome). Pretty-printers such as a2ps and the LaTeX "listings" package are also supported. Configuration files for three CADP languages (MCL, SVL, and XTL) and three CADP formats (BES, NUPN, and RBC) have been added.

Also, with the help of its principal author Pierre Boullier (Inria, Alpage), we corrected a memory allocation bug in the SYNTAX parser generator, which is used in most of the compilers developed by the CONVECS team.

6.2. Parallel and Distributed Verification

6.2.1. Distributed Code Generation for LNT

Participants: Hugues Evrard, Frédéric Lang.

Rigorous development and prototyping of a distributed algorithm using LNT involves the automatic generation of a distributed implementation. For the latter, a protocol realizing process synchronization is required. As far as possible, this protocol must itself be distributed, so as to avoid the bottleneck that would inevitably arise if a unique process would have to manage all synchronizations in the system. A particularity of such a protocol is its ability to support branching synchronizations, corresponding to situations where a process may offer a choice of synchronizing actions (which themselves may nondeterministically involve several sets of synchronizing processes) instead of a single one. Therefore, a classical barrier protocol is not sufficient and a more elaborate synchronization protocol is needed.

Using a synchronization protocol that we verified formally in 2013, we developed a prototype distributed code generator, named DLC (*Distributed LNT Compiler*), which takes as input the model of a distributed system described as a parallel composition of LNT processes.

In 2015, we finalized the development of DLC: the code was cleaned and the different compiler components were better integrated. A new option was added for the generated executables to dump at runtime an execution trace in the SEQUENCE format of CADP, for further analysis. A complete description of DLC, its synchronization protocol, performance data and usage examples were presented in Hugues Evrard's PhD thesis [9], defended in July 2015. An overview of DLC was presented in an international conference paper [23], and an extended version has been prepared for a journal special issue currently under construction. A tool paper was accepted in an international conference that will take place in 2016 [22].

6.2.2. Verification of Asynchronously Communicating Systems

Participants: Lakhdar Akroun, Gwen Salaün.

Analyzing systems communicating asynchronously via reliable FIFO buffers is an undecidable problem. A typical approach is to check whether the system is bounded, and if not, whether the corresponding state space can be made finite by limiting the presence of communication cycles in behavioral models or by fixing the buffer size. In this work, our focus is on systems that are likely to be unbounded and therefore result in infinite systems. We do not want to restrict the system by imposing any arbitrary bound. We introduced a notion of stability and proved that once the system is stable for a specific buffer bound, it remains stable whatever larger bounds are chosen for buffers. This enables one to check certain properties on the system for that bound and to ensure that the system will preserve them whatever larger bounds are used for buffers. We also proved that computing this bound is undecidable but we showed how we can succeed in computing these bounds for many typical examples using heuristics and equivalence checking.

6.2.3. Analysis of Verification Counterexamples

Participants: Gianluca Barbon, Gwen Salaün.

Model checking is an established technique for automatically verifying that a model, e.g., a Labelled Transition System (LTS), obtained from higher-level specification languages (such as process algebras) satisfies a given temporal property, e.g., the absence of deadlocks. When the model violates the property, the model checker returns a counterexample, which is a sequence of actions leading to a state where the property is not satisfied. Understanding this counterexample for debugging the specification is a complicated task for several reasons: (i) the counterexample can contain hundreds (even thousands) of actions, (ii) the debugging task is mostly achieved manually, and (iii) the counterexample does not give any clue on the state of the system (e.g., parallelism or data expressions) when the error occurs.

In collaboration with the SLIDE team of the LIG laboratory, we work on new solutions for simplifying the comprehension of counterexamples and thus favouring usability of model checking techniques. To do so, we apply pattern mining techniques to a set of correct traces (extracted from the LTS) and incorrect traces (corresponding to counterexamples), to identify specific patterns indicating more precisely the source of the problem.

6.3. Timed, Probabilistic, and Stochastic Extensions

6.3.1. Model Checking for Extended PCTL

Participants: Radu Mateescu, José Ignacio Requeno.

In the context of the SENSATION project (see § 8.2.1.1), we study the specification and verification of quantitative properties of concurrent systems.

In 2015, we developed a probabilistic version of ACTL (Action-based CTL) [41], named PACTL. This logic represents an action-based counterpart for PCTL (*Probabilistic Computation Tree Logic*) [50] and is interpreted naturally over DTMCs with labeled transitions, such as those produced from IPCs (*Interactive Probabilistic Chains*) [40]. The PACTL operators generalize those of ACTL: they characterize sequences of transitions in the DTMC by specifying both the states and the actions labeling the transitions. We implemented PACTL as an XTL library, which allows the designer to combine properties on actions, data, probabilities, and discrete time. We have experimented the PACTL library on several DTMCs imported from the probabilistic model checker PRISM [55] to ensure that both implementations provide the same numerical results.

6.4. Component-Based Architectures for On-the-Fly Verification

6.4.1. Compositional Verification

Participants: Hubert Garavel, Frédéric Lang.

The CADP toolbox contains various tools dedicated to compositional verification, among which EXP.OPEN, BCG_MIN, BCG_CMP, and SVL play a central role. EXP.OPEN explores on the fly the graph corresponding to a network of communicating automata (represented as a set of BCG files). BCG_MIN and BCG_CMP respectively minimize and compare behavior graphs modulo strong or branching bisimulation and their stochastic extensions. SVL (*Script Verification Language*) is both a high-level language for expressing complex verification scenarios and a compiler dedicated to this language.

In 2015, we corrected one bug in BCG_CMP and eight bugs in SVL. We extended the SVL language and compiler as follows:

- A new statement was added to translate a LOTOS file or a process in a LOTOS file to an LNT file automatically.
- LNT processes with data parameters can now be instantiated directly in the SVL script, without requiring a parameterless intermediate process to be defined.

- LNT processes with gate parameters can now be instantiated in the SVL script using the named parameter-passing style of LNT.
- Specification of a diagnostic file is now optional in the "`comparison`", "`deadlock`", and "`livelock`" statements of SVL.
- The "`property`" statement has been extended so that it can now contain any kind of statement, provided it contains at least one verification statement.
- Within SVL properties, it is now possible to define shell lines followed by an "`expected`" clause to specify the expected result of the shell line.
- It is now possible to add a "`result`" clause after a verification statement, so as to store the result of the verification in a shell variable that can be subsequently used in the SVL script.

We improved several demo examples of CADP by using these new SVL constructs, and we added a new demo example on the verification of an airplane-ground communication protocol.

We also improved the PMC tool, by correcting five bugs and adding a new "`-order`" option, which permits the user to define a particular order for quotienting. We improved the presentation of the demo examples released in the PMC distribution. Those examples are now given in LNT and translated automatically into networks of automata in the EXP language, instead of being given directly as networks of automata.

6.4.2. On-the-Fly Test Generation

Participants: Hubert Garavel, Radu Mateescu, Wendelin Serwe.

In the context of the collaboration with STMicroelectronics, we study techniques for testing if a (hardware) implementation is conform to a formal model described in LNT. Our approach is inspired by the theory of conformance testing [63], as implemented for instance in TGV [53] and JTorX [33]. We have developed three prototype tools to support this approach. The first tool implements a dedicated OPEN/CAESAR-compliant compiler for the particular asymmetric synchronous product between the model and the test purpose. The second tool, based on slightly extended generic components for graph manipulation (τ-compression, τ-confluence reduction, determinization) and resolution of Boolean equation systems, generates the complete test graph (CTG), which can be used to extract concrete test cases or to drive the test of the implementation. A third prototype tool takes as input a CTG and extracts either a single test case (randomly chosen or the first encountered one), or the set of *all* test cases. The principal advantage of our approach compared to existing tools is the use of LNT for describing test purposes, which facilitates the manipulation of data values.

In 2015, we corrected the prototype tools to properly handle timers and failure transitions, improved the documentation, and simplified internal data structures.

These prototype tools were used in the case study with STMicroelectronics (see § 6.5.1) and the EnergyBus (see § 6.5.4).

6.4.3. Other Component Developments

Participants: Soraya Arias, Hubert Garavel, Frédéric Lang, Radu Mateescu.

We separated the MCL library defining the operators of ACTL (Action-based CTL) [41] in two parts: the first one defines the operators of ACTL$\setminus$X (the fragment of ACTL without the next-time operators), including optimized definitions of derived temporal operators, and the second one defines the next-time operators, including the definitions of silent next-time operators, which complement the visible next-time operators already present in the library.

We also added an MCL library defining the operators of the Lμ-dsbr fragment of modal μ-calculus [6], which includes the ACTL$\setminus$X library. The Lμ-dsbr library also defines the absence of deadlock property as an MCL formula adequate w.r.t. divergence-sensitive branching bisimulation (divbranching for short) and allowing one to hide all visible actions in the LTS and to reduce it modulo divbranching prior to verification, which may bring significant performance gains.

A new major version 1.2 of the BCG format for storing Labelled Transition Systems was released as part of CADP 2015-a. Following this change, various minor residual bugs have been identified and fixed in 2015, and the type system of XTL has been modified to require fewer explicit type coercions.

In addition to bug fixes in various tools (e.g., CUNCTATOR, EUCALYPTUS, TST, XTL, etc.), the installation procedures of CADP have been revisited and updated; in particular, work is going on and many preliminary changes have been silently brought to ease installation of CADP on Windows.

6.5. Real-Life Applications and Case Studies

6.5.1. ACE Cache Coherency Protocol

Participants: Abderahman Kriouile, Radu Mateescu, Wendelin Serwe.

In the context of a CIFRE convention with STMicroelectronics, we studied system-level cache coherency, a major challenge faced in the current System-on-Chip architectures. Because of their increasing complexity (mainly due to the significant number of computing units), the validation effort using current simulation-based techniques grows exponentially. As an alternative, we study formal verification.

We focused on the ACE (AXI Coherency Extensions) cache coherency protocol, a system-level coherency protocol proposed by ARM [31]. In previous years, we developed a parametric formal model (about $3,400$ lines of LNT) of a system consisting of an ACE-based cache coherent interconnect, processors, and a main memory. We also specified temporal properties expressing cache coherence, data integrity, and successful completion of each transaction. Note that the former property required to transform state-based properties into action-based ones, by adding information about the cache state to the actions executed by the cache.

In 2015, we continued to exploit the formal model to improve the validation of the architecture under design at STMicroelectronics, in particular by integrating tests derived from the formal model into the official test plans. This work led to a publication in an international conference [25], and the defense of the PhD corresponding to the CIFRE convention [10].

6.5.2. Deployment and Reconfiguration Protocols for Cloud Applications

Participants: Rim Sakka Abid, Gwen Salaün.

Cloud applications are complex applications composed of a set of interconnected software components running on different virtual machines, hosted on remote physical servers. Deploying and reconfiguring this kind of applications are very complicated tasks especially when one or multiple virtual machines fail when achieving these tasks. Hence, there is a need for protocols that can dynamically reconfigure and manage running distributed applications.

In 2015, we proposed a novel protocol, which aims at reconfiguring cloud applications. This protocol is able to ensure communication between virtual machines and resolve dependencies by exchanging messages, (dis)connecting, and starting/stopping components in a specific order. The interaction between machines is assured via a publish-subscribe messaging system. Each machine reconfigures itself in a decentralized way. The protocol supports virtual machine failures, and the reconfiguration always terminates successfully even in the presence of a finite number of failures. Due to the high degree of parallelism inherent to these applications, the protocol was specified in LNT and verified using CADP. The use of formal specification languages and tools helped to detect several bugs and to improve the protocol. These results were published in [12].

Another line of work concerns autonomic computing in cloud environments. Managing distributed cloud applications is a challenging problem because manual administration is no longer realistic for these complex distributed systems. Thus, autonomic computing is a promising solution for monitoring and updating these applications automatically. This is achieved through the automation of administration functions and the use of control loops called autonomic managers. An autonomic manager observes the environment, detects changes, and reconfigures dynamically the application. Multiple autonomic managers can be deployed in the same system and must make consistent decisions. Using them without coordination may lead to inconsistencies and error-prone situations.

In 2015, we propose an approach for coordinating stateful autonomic managers, which relies on a simple coordination language, new techniques for asynchronous controller synthesis and Java code generation. We used our approach for coordinating real-world cloud applications. These results were published in [19].

6.5.3. Networks of Programmable Logic Controllers

Participants: Fatma Jebali, Jingyan Jourdan-Lu, Frédéric Lang, Eric Léo, Radu Mateescu.

In the context of the Bluesky project (see § 8.1.2.1), we study the software applications embedded on the PLCs (Programmable Logic Controllers) manufactured by Crouzet Automatismes. One of the objectives of Bluesky is to enable the rigorous design of complex control applications running on several PLCs connected by a network. Such applications are instances of GALS (*Globally Asynchronous, Locally Synchronous*) systems composed of several synchronous automata embedded on individual PLCs, which interact asynchronously by exchanging messages. A formal analysis of these systems can be naturally achieved by using the formal languages and verification techniques developed in the field of asynchronous concurrency.

For describing the applications embedded on individual PLCs, Crouzet provides a dataflow language with graphical syntax and synchronous semantics, equipped with an ergonomic user-interface that facilitates the learning and use of the language by non-experts. To equip the PLC language of Crouzet with functionalities for automated verification, the solution adopted in Bluesky was to translate it into GRL (see § 6.1.5), which enables the connection to testing and verification tools covering the synchronous and asynchronous aspects.

In 2015, we have provided support to Crouzet, who started to integrate GRL in the PLC design process by developing both a library of GRL blocks corresponding to function blocks present in their PLC programming tool, and an automated translation from a PLC program into a GRL block. The GRL2LNT and GRL.OPEN tools (see § 6.1.5) provide a direct connection to all verification functionalities of CADP, in particular model checking and equivalence checking.

We also investigated the equivalence checking for networks of PLCs, with the objective of proposing a general methodology usable in industrial context. We identified several rules (formalized as templates) for describing the asynchronous and synchronous parts of PLC networks, as well as their external behaviour (service), in order to facilitate the equivalence checking modulo branching bisimulation.

6.5.4. EnergyBus Standard for Connecting Electric Components

Participants: Hubert Garavel, Wendelin Serwe.

The EnergyBus [0] is an upcoming industrial standard for electric power transmission and management, based on the CANopen field bus. It is developed by a consortium assembling all major industrial players (such as Bosch, Panasonic, and Emtas) in the area of light electric vehicles (LEV); their intention is to ensure interoperability between all electric LEV components. At the core of this initiative is a universal plug integrating a CAN-Bus [0] with switchable power lines. The central and innovative role of the EnergyBus is to manage the safe electricity access and distribution inside an EnergyBus network.

In the framework of the European FP7 project SENSATION (see § 8.2.1.1) a formal specification in LNT of the main EnergyBus protocols is being developed by Alexander Graf-Brill and Holger Hermanns at Saarland University [48], with the active collaboration of CONVECS.

In 2015, we pursued the analysis of the LNT model, involving both verification (by means of state-space exploration and model checking techniques) and validation (using test cases automatically derived from the formal LNT model).

6.5.5. AutoFlight Control System

Participant: Fatma Jebali.

[0] http://www.energybus.org

[0] http://www.can-cia.org

In collaboration with Eric Jenn (IRT Saint Exupery, Toulouse), we studied an AutoFlight Control System (AFCS), provided by Thales Avionics. The goal of an AFCS is to improve the quality of flight and enhance the operational capability of the aircraft. The architecture of the AFCS comprises two parts. The first part (FCP, Flight Control Panel) consists of a control panel, which enables the pilot to interact with the system. The second part (AFS, Automatic Flight System) is in charge of performing functions such as guidance and automatic pilot. For safety purposes, each part is organized into a command and monitoring channels. The command channel ensures the function allocated to the component. The monitoring channel ensures that the command channel operates correctly. To ensure a sufficient availability level, a high level of redundancy is built inside the system. Components communicate using various communication means with different latencies (AFDX, A429, discrete).

Since AFCSs have stringent safety and time-critical requirements, formal verification is required to ensure their correctness. To this aim, we have applied the GRL approach for the formal modelling and verification of GALS systems (see § 6.1.5). In a first step, we have addressed the AFCS without redundancy. We have written a GRL description (750 lines), which can be parameterized by the activation paces of different synchronous components. We have written a set of correctness properties in MCL, which we have verified on the GRL model.

6.5.6. Graphical User-Interfaces and Plasticity

Participants: Hubert Garavel, Frédéric Lang, Raquel Oliveira.

In the context of the Connexion project (see § 8.1.1.2) and in close collaboration with Gaëlle Calvary and Sophie Dupuy-Chessa (IIHM team of the LIG laboratory), we study the formal description and validation of graphical user-interfaces using the most recent features of the CADP toolbox. The case study assigned to LIG in this project is a prototype graphical user-interface [38] designed to provide human operators with an overview of a running nuclear plant. The main goal of the system is to inform the operators about alarms resulting from faults, disturbances, or unexpected events in the plant. Contrary to conventional control rooms, which employ large desks and dedicated hardware panels for supervision, this new-generation interface uses standard computer hardware (i.e., smaller screen(s), keyboard, and mouse), thus raising challenging questions on how to best provide synthetic views of the plant status. Another challenge is to introduce plasticity in such interface, so as to enable several supervision operators, including mobile ones outside of the control room, to get accurate information in real time.

We formally specified this new-generation interface in LNT, encompassing not only the usual components traditionally found in graphical user-interfaces, but also a model of the physical world (namely, a nuclear reactor with various fault scenarios) and a cognitive model of a human operator in charge of supervising the plant. Also, several desirable properties of the interface have been expressed in MCL and verified on the LNT model using CADP. At last, we used our formal model to check conformance of execution traces generated by an industrial control room prototype provided by a partner in the project.

In 2015, we finalized our approach to formally verifying safety critical interactive systems provided with plastic user interfaces, either using equivalence checking (to check whether different versions of user interfaces present the same interaction capabilities and appearance) or model checking (to check a set of properties over a model of the system). The results have been published in international conferences [26], [27] and journals [17], and in Raquel Oliveira's PhD thesis [11].

6.5.7. Fault-Tolerant Routing for Network-on-Chip Architectures

Participant: Wendelin Serwe.

Fault-tolerant architectures provide adaptivity for on-chip communications, but also increase the complexity of the design, so that formal verification techniques are needed to check their correctness. In collaboration with Chris Myers and Zhen Zhang (University of Utah, USA), we studied an extension of the link-fault tolerant Network-on-Chip (NoC) architecture introduced by Wu *et al* [64] that supports multiflit wormhole routing. A major difference with similar architectures existing in the literature is that the considered routing algorithm is not statically proven free of deadlocks, but rather implements deadlock avoidance (by dynamically detecting possible deadlock situations and avoiding them by dropping packets).

In 2015, we detected a potential livelock in the previously developed formal LNT model [65]. The correction of this problem led to an improved routing algorithm, for which the state space for 2x2 NoCs could be generated compositionally. We also experimented with the analysis of larger configurations on Grid'5000, but even a 2x3 NoC is still too large, so that compositional state space generation fails with an intermediate state space of several billions of states. This work led to a publication accepted in an international journal [18] and a PhD thesis [66].

6.5.7.1. Other Case Studies

The demo examples of CADP, which have been progressively accumulated since the origins of the toolbox, are a showcase for the multiple capabilities of CADP, as well as a test bed to assess the new features of the toolbox. In 2015, the effort to maintain and enhance these demos has been pursued. The progressive migration to LNT has continued, by translating five demos (16, 21, 22, 36, and 38) from LOTOS to LNT. A new demo 05 (airplane-ground communication protocol) has been added. The code of many demos was updated to use the latest features of LNT, such as "`in var`" parameters and "`assert`" statements. Demos 14 and 16 have been greatly simplified by inlining MCL and XTL temporal logic formulas in SVL scripts, using the "`property`", "`check`", and "`|=`" statements recently added to SVL. Nine demos (02, 08, 17, 20, 27, 28, 31, 33, and 36) have been simplified by using the new possibility to pass value parameters to LOTOS and LNT processes directly in SVL scripts. XTL formulas have been shortened in demos 23 and 27. The illustration of the EXEC/CAESAR framework in demo 38 has been integrated as a property into the SVL script. Finally, demo 38 led to a publication in an international workshop [29].

HYCOMES Team

6. New Results

6.1. Embedded Systems Design

6.1.1. Loosely Time-Triggered Architectures: Improvements and Comparisons

Participant: Albert Benveniste.

Loosely Time-Triggered Architectures (LTTAs) are a proposal for constructing distributed embedded control systems. They build on the quasi-periodic architecture, where computing units execute 'almost periodically', by adding a thin layer of middleware that facilitates the implementation of synchronous applications. In [7], we have shown how the deployment of a synchronous application on a quasi-periodic architecture can be modeled using a synchronous formalism. Then we have detailed two protocols, Back-Pressure LTTA, reminiscent of elastic circuits, and Time-Based LTTA, based on waiting. Compared to previous work, we presented controller models that can be compiled for execution and a simplified version of the Time-Based protocol. We also compared the LTTA approach with architectures based on clock synchronization.

6.2. Hybrid Systems Modeling

Participants: Ayman Aljarbouh, Albert Benveniste, Benoît Caillaud, Khalil Ghorbal.

6.2.1. Robust Simulation for Hybrid Systems: Chattering Path Avoidance

The sliding mode approach is recognized as an efficient tool for treating the chattering behavior in hybrid systems. However, the amplitude of chattering, by its nature, is proportional to magnitude of discontinuous control. A possible scenario is that the solution trajectories may successively enter and exit as well as slide on switching mani-folds of different dimensions. Naturally, this arises in dynamical systems and control applications whenever there are multiple discontinuous control variables. The main contribution of [9] is to provide a robust computational framework for the most general way to extend a flow map on the intersection of p intersected $(n-1)$-dimensional switching manifolds in at least p dimensions. We explored a new formulation to which we can define unique solutions for such particular behavior in hybrid systems and investigate its efficient computation/simulation. An extended version of this work has been presented at the Baltic Young Scientists Conference [8].

6.2.2. A Hierarchy of Proof Rules for Checking Positive Invariance of Algebraic and Semi-Algebraic Sets

In [6], we studied sound proof rules for checking positive invariance of algebraic and semi-algebraic sets, that is, sets satisfying polynomial equalities and those satisfying finite boolean combinations of polynomial equalities and inequalities, under the flow of polynomial ordinary differential equations. Problems of this nature arise in formal verification of continuous and hybrid dynamical systems, where there is an increasing need for methods to expedite formal proofs. We study the trade-off between proof rule generality and practical performance and evaluate our theoretical observations on a set of benchmarks. The relationship between increased deductive power and running time performance of the proof rules is far from obvious; we discuss and illustrate certain classes of problems where this relationship is interesting.

6.2.3. A Formally Verified Hybrid System for Safe Advisories in the Next-Generation Airborne Collision Avoidance System

The Next-Generation Airborne Collision Avoidance System (ACAS X) is intended to be installed on all large aircraft to give advice to pilots and prevent mid-air collisions with other aircraft. It is currently being developed by the Federal Aviation Administration (FAA). In [16] we determined the geometric configurations under which the advice given by ACAS X is safe under a precise set of assumptions and formally verify these configurations using hybrid systems theorem proving techniques. We considered subsequent advisories and showed how to adapt our formal verification to take them into account. We examined the current version of the real ACAS X system and discussed some cases where our safety theorem conflicts with the actual advisory given by that version, demonstrating how formal, hybrid systems proving approaches are helping to ensure the safety of ACAS X. Our approach is general and could also be used to identify unsafe advice issued by other collision avoidance systems or confirm their safety.

6.2.4. Domain Globalization: Using Languages to Support Technical and Social Coordination

When a project is realized in a globalized environment, multiple stakeholders from different organizations work on the same system. Depending on the stakeholders and their organizations, various (possibly overlapping) concerns are raised in the development of the system. In this context a Domain Specific Language (DSL) supports the work of a group of stakeholders who are responsible for addressing a specific set of concerns. We contributed to a book chapter [11], identifying the open challenges arising from the coordination of globalized domain-specific languages. We identified two types of coordination: technical coordination and social coordination. After presenting an overview of the current state of the art, we discussed first the open challenges arising from the composition of multiple DSLs, and then the open challenges associated to the collaboration in a globalized environment.

6.3. Contracts for Systems Design

Participants: Albert Benveniste, Benoît Caillaud.

6.3.1. Contracts for Systems Design: Theory, Methodology and Application Cases

Aircrafts, trains, cars, plants, distributed telecommunication military or health care systems, and more, involve systems design as a critical step. Complexity has caused system design times and costs to go severely over budget so as to threaten the health of entire industrial sectors. Heuristic methods and standard practices do not seem to scale with complexity so that novel design methods and tools based on a strong theoretical foundation are sorely needed. Model-based design as well as other methodologies such as layered and compositional design have been used recently but a unified intellectual framework with a complete design flow supported by formal tools is still lacking. Recently an “orthogonal” approach has been proposed that can be applied to all methodologies introduced thus far to provide a rigorous scaffolding for verification, analysis and abstraction/refinement: contract-based design. Several results have been obtained in this domain but a unified treatment of the topic that can help in putting contract-based design in perspective is missing. We have published two research reports [13], [12], that intend to provide such treatment where contracts are precisely defined and characterized so that they can be used in design methodologies such as the ones mentioned above with no ambiguity. In addition, the first report [13] provides an important link between interface and contract theories to show similarities and correspondences. This report is complemented by a companion report [12] where contract based design is illustrated through use cases.

6.3.2. Contracts for Schedulability Analysis

In [10] we proposed a framework of Assume / Guarantee contracts for schedulability analysis. Unlike previous work addressing compositional scheduling analysis, our objective is to provide support for the OEM / supplier subcontracting relation. The adaptation of Assume / Guarantee contracts to schedulability analysis requires some care, due to the handling of conflicts caused by shared resources. We illustrate our framework in the context of Autosar methodology now popular in the automotive industry sector.

MUTANT Project-Team

7. New Results

7.1. Weakly-Supervised Discriminative Model for Audio-to-Score Alignment

We consider a new discriminative approach to the problems of segmentation and of audio-to-score alignment. For each musical event, templates have to be built or learnt before performing any alignment. Because annotating a large database music files would be a tedious task, we develop an original approach to learn templates without annotations, but only the knowledge of the music scores associated to music files. We consider the two distinct informations provided by the music scores: (i) an exact ordered list of musical events and (ii) an approximate prior information about relative duration of events. We extend the celebrated Dynamic Time Warping algorithm (DTW) to a convex problem that learns optimal classifiers for all events while jointly aligning files, using this weak supervision only. We show that the relative duration between events can be easily used as a penalization of our cost function and allows us to drastically improve performances of our approach. We describe in details our approach and preliminary results obtained on a large-scale database in [18].

This work was done in collaboration with the SIERRA project-team at Inria Paris.

7.2. Semi-Markov Models for Real-time MIDI-to-Score Alignment

We develop a new stochastic model of symbolic (MIDI) performance of polyphonic scores, based on Semi-Markov models, to align MIDI performances of music scores. In our approach, the evolution of the music performer and the production of performed notes are modeled with a hierarchical extension of hidden semi-Markov models (HSMM). By comparing with a previously studied model based on hidden Markov model (HMM), we give theoretical reasons why the present model is advantageous to deal with complex music event such as trills, tremolos, arpeggios, and other ornaments. This is also confirmed empirically by comparing the accuracy of score following and analysing the errors. We also develop a hybrid of this HSMM-based model and the HMM-based model which is computationally more efficient and retains the advantages of the former model. The present model yields one of the state-of-the-art score following algorithms for symbolic performance and can possibly be applicable for other music recognition problems. Details and results are published in [19].

This work was done in collaboration with Eita Nakamura from the National Institute of Informatics of Tokyo, Japan.

7.3. Real-time Audio-to-Score Alignment of Singing Voice

Singing voice is specific in music: a vocal performance conveys both music (melody/pitch) and lyrics (text/phoneme) content. We develop and original approach that aims at exploiting the advantages of melody and lyric information for real-time audio-to-score alignment of singing voice. First, lyrics are added as a separate observation stream into a template-based hidden semi-Markov model (HSMM), whose observation model is based on the construction of vowel templates. Second, early and late fusion of melody and lyric information are processed during real-time audio-to-score alignment. An experiment conducted with two professional singers (male/female) shows that the performance of a lyrics-based system is comparable to that of melody-based score following systems. Furthermore, late fusion of melody and lyric information substantially improves the alignment performance. Finally, maximum a posteriori adaptation (MAP) of the vowel templates from one singer to the other suggests that lyric information can be efficiently used for any singer. Preliminary results are published in [15].

7.4. Online Methods for Audio Segmentation and Clustering

Audio segmentation is an essential problem in many audio signal processing tasks, which tries to segment an audio signal into homogeneous chunks. Rather than separately finding change points and computing similarities between segments, we focus on joint segmentation and clustering, using the framework of hidden Markov and semi-Markov models. We introduced a new incremental EM algorithm for hidden Markov models (HMMs) and showed that it compares favorably to existing online EM algorithms for HMMs. Early experimental results on musical note segmentation and environmental sound clustering are promising and will be pursued further in 2015.

Theoretical results were published in [11] in collaboration with the SIERRA project-team, and experimental results were further extended in [32]. Early experimental setups show that our algorithms out perform state-of-the-art supervised methods for Percussion Sound classification. In collaboration with IRCyNN (Nantes) we are currently studying algorithmic extensions to complex environmental sounds.

7.5. Adaptive Synchronization Strategies for Automatic Accompaniment

José Echeveste developped several synchonization strategies in the framework of his PhD thesis. Their formalization is based on a dynamic real-time extension of the time map formalism, going beyond state-of-the-art where the largest body of literature on time maps is devoted to static functions, defined and known at all times before any manipulation is done. Only the latest work of Liang and Danneberg (2011) have considered dynamic time map in the synchronization problem. However their approach suffer from a consistency drawback: the convergence of the tempo depends on the events occuring during the catching trajectory. In our approach we have developped a lag-depend formulation of the catching trajectory, which is insensitive to the actual events. This adaptive strategy consider only the deviation in tempo and position and is otherwise context-independant, it ensure convergence both in position and tempo, and it is efficient: there is no need to a fine sampling clock to discretize the time evolution: as long as the prediction time map do not change, delays are computed only once using the accompaniment time map. Our approach is general enough to handle various important issues in automatic accompaniment: latency management, integration of non constant tempo specifications in the score (*accelerando*, *ritardanto*, *rubato*...), handling of missing events, *etc.* Synchronization strategies have been fully formalized in the PhD report of José Echeveste [8] together with a complete Antescofo core including other dynamic constructions.

7.6. Temporal objects for the design of reusable library in Antescofo

Composers develops their own idiosyncratic compositional language through their pieces. In addition, composers and sound engineers have to face drastically different performance set-up for the same piece. This situation advocate for the development of new generic mechanism to simplify the development of generic yet dedicated libraries in Antescofo. In cooperation with various composers (Marco Stroppa, Julia Blondeau, Jason Freeman, Jose Miguel Fernandez, Yann Marez) we have introduced seevral new mechanisms in Antescofo to ease the building of dedicated yet reusable library of compositional pieces: extnesion of the functional language to include new control structure, introduction of *continuation combinators* making possible to start actions at the end of other durative actions, marshalling of Antescofo values, *etc.* The most notable ones are actor-based features to implement *temporal objects*. Object templates are specified and then instantiated at will. A temporal object encapsulate a local state; it can react to logical condition; it offers instantaneous as well as durative methods; reaction to synchronous broadcast can be defined as well as exceptional condition handlers. These new features are currently tested in the development of new pieces and are expected to evolve following the feedbacks from these applications.

7.7. Embedding real-time audio computation in Antescofo

DSP processing in Antescofo is an experimental extension of the language started in 2014 and aimed at driving various DSP processing capabilities directly within Antescofo. DSP processors are defined directly in an Antescofo score, harnessing various signal processing libraries. These DSP processors are then dynamically connected together using Antescofo audio links. Input and output channels are used to link these processors with the host environment while internal channels connect DSP among themselves. The connections are specified with a new kind of Antescofo actions, the patch. So, the connections can be changed dynamically in response to the events detected by the listening machine and can be synchronized using the expressive repertoire of synchronization strategies available in Antescofo. Ordinary Antescofo variables can be used to control the DSP computations, which add an additional level of dynamicity. Currently, FAUST and and a few specific signal processors (notably FFT) can be defined. Several benefits results of this tight integration. The network of signal processors is heterogeneous, mixing DSP nodes specified with different tools. The network of signal processors can change dynamically in time following the result of a computation. This approach answers the shortcomings of fixed (static) dataflow models of the Max or PureData host environments. Signal processing is controlled at a symbolic level and can be guided, *e.g.* by information available in the augmented score (like position, expected tempo, etc.). The tight integration makes possible to specify, concisely and more effectively, finer and more precise control of the signal processing, at a lower computational cost. One example is the use of symbolic curve specification to specify variations of control parameters at sample rate. It makes it possible to embed sound analysis inside Antescofo as well. At last but not least, signal processing can be done more efficiently. For example, in the *remaking* of Boulez' piece *Antheme 2* there is an improvment of performance in time of 45 % compared to the original version with the audio effects managed in Max.

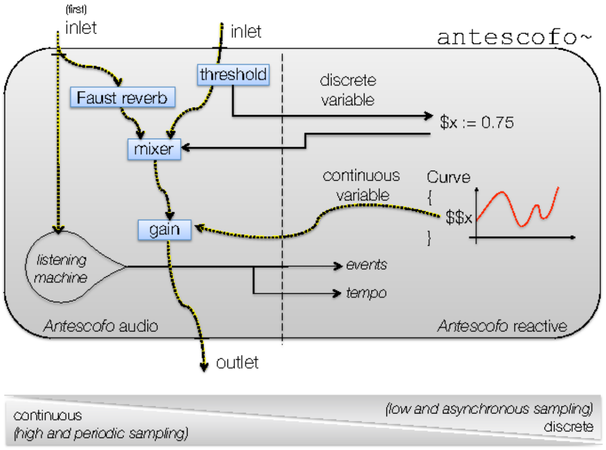


Figure 8. Articulation between DSP and the reactive engine.

The current work focuses on the development of a dedicated type system enabling a finer control of scheduling and audio buffer size, refining results previously developped in the cyclostatic scheduling of synchronous dataflow. Early results are published in [29].

7.8. Visualizing Timed and Hierarchical Code Structures

This work applies an information visualisation perspective to a set of revisions in the timeline-based representation of action items in *AscoGraph*, the dedicated user interface to Antescofo. Our contribution is twofold: (a) a design study of the proposed new model, and (b) a technical, algorithmic component. In the former, we show how our model relates to principles of information coherence and clarity, facility of seeking and navigation, hierarchical distinction and explicit linking. In the latter, we frame the problem of arranging action rectangles in a 2D space as a strip packing problem, with the additional constraint that the (horizontal) time coordinates of each block are fixed. We introduce three algorithms of increasing complexity for automatic arrangement, estimate their packing performance and analyse their strengths and weaknesses. We evaluate the systemic improvements achieved and their applicability for other time-based datasets. Furthermore, algorithms for efficient automatic stacking of time-overlapping action blocks are developped, as well as mathematical proof for their time-coherency during dynamic visualizations.

Results are implemented in Section 6.2 and reported in [12] and [13].

7.9. Model-based Testing an Interactive Music System

We have been pursuing our studies on the application of model-based timed testing techniques to the interactive music system (IMS) Antescofo, in the context of the Phd of Clément Poncelet and in relation with the developments presented in Section 6.3 .

Several formal methods have been developed for automatic conformance testing of critical embedded software, with the execution of a real implementation under test (IUT, or black-box) in a testing framework, where carefully selected inputs are sent to the IUT and then the outputs are observed and analyzed. In conformance model-based testing (MBT), the input and corresponding expected outputs are generated according to formal models of the IUT and the environment. The case of IMS presents important originalities compared to other applications of MBT to realtime systems. On the one hand, the time model of IMS comprises several time units, including the wall clock time, measured in seconds, and the time of music scores, measured in number of beats relatively to a tempo. This situation raises several new problems for the generation of test suites and their execution. On the other hand, we can reasonably assume that a given mixed score of Antescofo specifies completely the expected timed behavior of the IMS, and compile automatically the given score into a formal model of the IUT's expected behavior, using an intermediate representation. This give a fully automatic test method, which is in contrast with other approaches which generally require experts to write the specification manually.

We have developed online and offline approches to MBT for Antescofo. The offline approach relies on tools of the Uppaal suite [38], [37], using a translation of our models into timed automata. These results have been presented during the 30th ACM/SIGAPP Symposium On Applied Computing, track Software Verification and Testing [21] and an article describing this approach has been accepted for publication in the Journal of New Music Research. The online approach is based on a new virtual machine executing the models of score in intermediate representation (see Section 6.3).

7.10. Representation of Rhythm and Quantization

Rhythmic data are commonly represented by tree structures (rhythms trees) in assisted music composition environments, such as OpenMusic, due to the theoretical proximity of such structures with traditional musical notation. We are studying the application in this context of techniques and tools for processing tree structure, which were originally developed for other areas such as natural language processing, automatic deduction, Web data processing... We are particularly interested in two well established formalisms with solid theoretical foundations: tree automata and term rewriting.

Our first main contribution in that context is the development of a new framework for rhythm transcription, the problem of the generation, from a sequence of timestamped notes, *e.g.* a file in MIDI format, of a score in traditional music notation) – see Section 6.4 . This problem arises immediately as insoluble unequivocally: we shall calibrate the system to fit the musical context, balancing constraints of precision, or of simplicity / readability of the generated scores. We are developing in collaboration with Jean Bresson (Ircam) and Slawek Staworko (LINKS, currently on leave at University of Edinburgh) an approach based on algorithms for the enumeration of large sets of weighted trees (tree series), representing possible solutions to a problem of transcription. The implementation work is performed by Adrien Ycart, under a research engineer contract with Ircam. This work has been presented in [23].

Our second contribution, in collaboration with Prof. Masahiko Sakai (Nagoya University), is a proposal of a structural theory (equational system on rhythm trees) defining equivalence on rhythm notations [14], [16]. This approach can be used for example to generate, by transformation, different notations possible the same rate, with the ability to select in accordance with certain constraints. We have also conducted related work on the theory of term rewriting [17].

PARKAS Project-Team

6. New Results

6.1. Reasoning about C11 Program Transformations

Participants: Francesco Zappa Nardelli, Robin Morisset.

We have shown that the weak memory model introduced by the 2011 C and C++ standards does not permit many of common source-to-source program transformations (such as expression linearisation and "roach motel" reordering) that modern compilers perform and that are deemed to be correct. As such it cannot be used to define the semantics of intermediate languages of compilers, as, for instance, LLVM aimed to. We consider a number of possible local fixes, some strengthening and some weakening the model. We have evaluated the proposed fixes by determining which program transformations are valid with respect to each of the patched models. We have provided formal Coq proofs of their correctness or counterexamples as appropriate.

A paper on this work has been accepted in [18]. In collaboration with Viktor Vafeiadis (MPI-SWS, Germany) and Thibaut Balabonski (U. Paris Sud).

6.2. Language design on top of JavaScript

Participant: Francesco Zappa Nardelli.

This research project aims at improving the design of the JavaScript language. We propose a typed extension of JavaScript combining dynamic types, concrete types and like types to let developers pick the level of guarantee that is appropriate for their code. We have implemented our type system in the V8 JavaScirpt engine and we have explored the performance and software engineering benefits.

A paper on this work has been accepted in ECOOP 2015 [21].

With Gregor Richards (Waterloo University) and Jan Vitek (Northeastern University).

6.3. Synchronous Functional Language with Integer Clocks

Participant: Adrien Guatto.

Adrien Guatto defended his PhD thesis on the modular description of space/time tradeoffs at the language level. His thesis work extends the n-synchronous framework proposed by Cohen, Mandel, Plateau, Pouzet and others. Clocks now feature arbitrary positive integers that model bursty communication between subprograms: “integer clocks”. The activation conditions of Lustre are revisited in this new setting to become “local time scales” that allow subprograms to perform several steps atomically relative to their context. The thesis details the integration of these features in a clock type system for a higher-order functional language, giving full formal treatment of its metatheory and compilation to finite-state digital circuits.

6.4. Fidelity in Real-Time Programming

Participants: Guillaume Baudart, Timothy Bourke.

In this work we study embedded systems with a significant mix of discrete reactive behaviours and ‘physical’ timing constraints. The idea is to make the most of the advantages of synchronous languages for precisely specifying discrete behaviours but to adapt or extend them to treat real-time constraints more abstractly, that is, without an *a priori* definition of an eventual sampling interval.

This year we concluded our study of the Loosely Timed-Triggered Architectures (LTTA) by developing simplified models of the underlying implementations and protocols. This enabled us to improve the protocols, simplify the correctness and performance arguments, and compare them to systems built using modern clock synchronization algorithms. We developed our models in the Zélus programming language which enables (instances of) them to be compiled for simulation and contributes to our work on better exploiting synchronous languages for real-time specification and analysis. This work was presented at the EMSOFT conference and a journal article has been submitted.

This year we also concluded our study of the Quasi-synchronous Approach to modelling real-time distributed systems. We formalized the relation between the discrete abstraction proposed by Paul Caspi and the real-time archictectures for which it is intended. This enabled us to precisely state a correctness requirement for the abstraction and to show that it is sound for systems of two nodes (a typical case explored in other publications) but not for general systems of three or more nodes. Our formalization clarifies the relation between the causality of traces of the real-time system and the causality introduced by the synchronous abstraction. This enables us to state and show necessary and sufficient restrictions on the communication topologies and timing characteristics of systems to ensure soundness. A paper explaining this result has been drafted and will be submitted early in 2016.

6.5. Verified compilation of Lustre

Participants: Timothy Bourke, Marc Pouzet.

Synchronous dataflow languages and their compilers are increasingly used to develop safety-critical applications, like fly-by-wire controllers in aircraft and monitoring software for power plants. A striking example is the SCADE Suite tool of ANSYS/Esterel Technologies which is DO-178B/C qualified for the aerospace and defense industries. This tool allows engineers to develop and validate systems at the level of abstract block diagrams that are automatically compiled into executable code.

Formal modelling and verification in an interactive theorem prover can potentially complement the industrial certification of such tools to give very precise definitions of language features and increased confidence in their correct compilation; ideally, right down to the binary code that actually executes.

This year we picked up on previous work in the PARKAS team to develop a verified compiler for a Lustre/SCADE-like synchronous language. We focused on the critical and until now unresolved compiler stage that transforms dataflow equations into imperative code. We developed, in Coq, a prototype compiler for the core language (without modular resets or tuples) and showed its correctness with respect to a dataflow semantics based on functions from natural numbers to present or absent values. This required the development of a novel intermediate model for relating delayed dataflow streams to imperative memories in such a way that a critical induction could be stated and proved. We further showed how to justify a post-transformation optimization that is essential for the efficiency of clock-directed code generation. We are preparing a paper describing these results. Work continues on both semantic questions (existence of a semantics for well-type and well-clocked programs, treatment of resets, etc.) and compilation issues (integration with the verified CompCert compiler).

In collaboration with Pierre-Évariste Dagand (CNRS) and Lionel Reig (Collège de France).

POSET Team

7. New Results

7.1. Efficient interactive score

We have proposed a solution to the problem of real-time performance for interactive multimedia applications, specifically in the interpretation of interactive multimedia scores. For that, we have proposed a new parallel implementation of interactive scores on a reconfigurable hardware. We take advantage of the parallelism and reliability provided by Field Programmable Gate Arrays (FPGAs) to perform in real-time the hardware representation of scores. The results of the simulations show that our approach allows the system to react instantaneously to user interactions. Moreover, the real-time constraints of the score are satisfied [21].

7.2. Modeling with tile

In [3], [8] it has been observed that musical objects are conveniently modeled by tiles. These modeling experiments have been continued this year by showing, in particular, how both high-level music modeling and low-level signal combination can be modeled by means of tiles [23]. This has been further extended relating classical musical constructs with tile modeling features [34].

7.3. Tiles and inverse semigroups

In [10] it has already been observed that the theory of inverse semigroups [0] is the adequate mathematical framework to define and study tiles and their languages. In this direction, we have shown that strings, trees and even many types of graphs can be unified into a notion of higher-dimensional strings [24], [35]. Using techniques of partial algebra [4], this notion recovers advanced results on formal languages of graphs of bounded tree-width [0], which shows the robustness of the approach.

7.4. Reactive programming with tile

The first step towards programming music with tiles is proposed as a Domain-Specific Language : the T-calculus [9]. Further collaboration with Paul Hudak [7] led us to various implementation experiments on top of Haskell [30], [33], [29]. Within the ADT “Tuilage” and S. Archipoff’s PhD thesis in progress, we eventually managed to integrate tile modeling into reactive programming as illustrated, in December 2015, by the first concert of the Idex Arts & Science project “Sound of Algorithm” in collaboration with the musician Edwin Buger.

7.5. Behavioral properties of higher-order programs

In a series of results [28], [27], we have been able to cast to traditional denotational semantics the behavioral properties captured by Monadic Second Order Logic (MSOL) and weak MSOL. The main difficulty was to represent infinitary properties in finitary models. From a foundational point of view, these results exhibit once more the robustness of approaches based on recognizability to capture complex properties of programs. They also make salient the problem of program evaluation in finite models as a milestone towards effective model-checking of higher-order programs.

[0] See [43] for general presentation of inverse semigroup theory, and [45], [44] for graph-based representation of inverse semigroup elements.

[0] See [38] for an up-to-date presentation of the formal language theory of graphs.

7.6. Art & Science project

This year has seen the members of PoSET involved in a number of Art & Science projection, especially some granted by Idex Bordeaux, including but not limited to : *Illumination* created Aurelio Edler-Copes, in partnership with compagnie Eclats, performed in November 2015 at Molière Theater, *Mobiles* and *Le Chant du filament #2* respectively created by Renaud Rubiano and Nicolas Villenave, performed or displayed in November 2015 during FACTS festival, the Art and Science Festival of Bordeaux University, and, *Le son des algorithmes* with Edwin Buger that led to a first musical performance in December 2015.

SPADES Project-Team

6. New Results

6.1. Components and contracts

Participants: Sophie Quinton, Jean-Bernard Stefani.

6.1.1. Multi-viewpoint contracts for the negotiation of embedded software updates

In the context of the CCC project (http://ccc-project.org/) we address the issue of change after deployment in safety-critical embedded system applications. Our goal is to substitute lab-based verification with in-field formal analysis to determine whether an update may be safely applied. This is challenging because it requires an automated process able to handle multiple viewpoints such as functional correctness, timing, etc. For this purpose, we propose an original methodology for contract-based negotiation of software updates. The use of contracts allows us to cleanly split the verification effort between the lab and the field. In addition, we show how to rely on existing viewpoint-specific methods for update negotiation. We have started validating our approach on a concrete example inspired by the automotive domain in collaboration with our German partners from TU Braunschweig.

6.1.2. Location Graphs

The design of configurable systems can be streamlined and made more systematic by adopting a component-based structure, as demonstrated with the FRACTAL component model [2]. However, the formal foundations for configurable component-based systems, featuring higher-order capabilities where components can be dynamically instantiated and passivated, and non-hierarchical structures where components can be contained in different composites at the same time, are still an open topic. We have recently introduced the location graph model [88], where components are understood as graphs of locations hosting higher-order processes, and where component structures can be arbitrary graphs.

We have continued the development of the location graph model and extended it in several directions. First we have introduced basic capabilities and predicate parameters in the model to allow for different forms of architectural invariants, such as different forms of encapsulation, to be maintained even in presence of dynamic graph modifications. Second, we have started developing the premises of a refinement theory for location graphs, showing in particular how one could refine a location process into a whole graph. Finally, we have shown how to handle heterogeneous forms of composition in the same location graph, turning each location into a composition operator. This work has not yet been published.

6.2. Real-Time multicore programming

Participants: Vagelis Bebelis, Adnan Bouakaz, Pascal Fradet, Alain Girault, Gregor Goessler, Xavier Nicollin, Jean-Bernard Stefani.

6.2.1. A time predictable programming language for multicores

Time predictability (PRET) is a topic that emerged in 2007 as a solution to the ever increasing unpredictability of today's embedded processors, which results from features such as multi-level caches or deep pipelines [59]. For many real-time systems, it is mandatory to compute a strict bound on the program's execution time. Yet, in general, computing a tight bound is extremely difficult [92]. The rationale of PRET is to simplify both the programming language and the execution platform to allow more precise execution times to be easily computed [38].

Following our past results on the PRET-C programming language [36], we have proposed a time predictable synchronous programming language for multicores, called FOREC. It extends C with a small set of ESTEREL-like synchronous primitives to express concurrency, interaction with the environment, looping, and a synchronization barrier [93] (like the `pause` statement in ESTEREL). FOREC threads communicate with each other via shared variables, the values of which are *combined* at the end of each tick to maintain deterministic execution. FOREC is compiled into threads that are then statically scheduled for a target multicore chip. Our WCET analysis takes into account the access to the shared TDMA bus and the necessary administration for the shared variables. We achieve a very precise WCET (the over-approximation being less than 2%) thanks to a reachable space exploration of the threads' states.

Recent results have addressed the semantics, the compiler, and the experiments. In particular, we have seeked to provide several combine policies for shared variables, in a way similar as concurrent revisions [49].

This work has been conducted within the RIPPES associated team.

6.2.2. Modular distribution of synchronous programs

Synchronous programming languages describe functionally centralized systems, where every value, input, output, or function is always directly available for every operation. However, most embedded systems are nowadays composed of several computing resources. The aim of this work is to provide a language-oriented solution to describe *functionally distributed reactive systems*. This research started within the Inria large scale action SYNCHRONICS and is a joint work with Marc Pouzet (ENS, PARKAS team from Rocquencourt) and Gwenaël Delaval (UGA, CTRL-A team from Grenoble).

We are working on defining a *fully-conservative* extension of a synchronous data-flow programming language (the HEPTAGON language, inspired from LUCID SYNCHRONE [51]). The extension, by means of *annotations* adds *abstract location parameters* to functions, and *communications* of values between locations. At deployment, every abstract location is assigned an actual one; this yields an executable for each actual computing resource. Compared to the PhD of Gwenaël Delaval [56], [57], the goal here is to achieve *modular* distribution even in the presence of non-static clocks, *i.e.*, clocks defined according to the value of inputs.

By *fully-conservative*, we have three aims in mind:

1. A non-annotated (*i.e.*, centralized) program will be compiled exactly as before;
2. An annotated program eventually deployed onto only one computing location will behave exactly as its centralized couterpart;
3. The input-output semantics of a distributed program is the same as its centralized counterpart.

By *modular*, we mean that we want to compile each function of the program into a single function capable of running on any computing location. At deployment, the program of each location may be optimized (by simple Boolean-constant-propagation, dead-code and unused-variable elimination), yielding different optimized code for each computing location.

We have formalized the type-system for inferring the location of each variable and computation. In the presence of local clocks, added information is computed from the existing clock-calculus and the location-calculus, to infer necessary communication of clocks between location. The overall structure of the new compiler is defined, including new algorithms for deployment (and code optimization), achieving the three aims detailed above.

6.2.3. Analysis and scheduling of parametric dataflow models

Recent data-flow programming environments support applications whose behavior is characterized by dynamic variations in resource requirements. The high expressive power of the underlying models (*e.g.*, Kahn Process Networks or the CAL actor language) makes it challenging to ensure predictable behavior. In particular, checking *liveness* (*i.e.*, no part of the system will deadlock) and *boundedness* (*i.e.*, the system can be executed in finite memory) is known to be hard or even undecidable for such models. This situation is troublesome for the design of high-quality embedded systems.

Recently, we have introduced the *Schedulable Parametric Data-Flow* (SPDF) MoC for dynamic streaming applications [62], which extends the standard dataflow model by allowing rates to be parametric, and the *Boolean Parametric Data Flow* (BPDF) MoC [42], [41] which combines integer parameters (to express dynamic rates) and boolean parameters (to express the activation and deactivation of communication channels). High dynamicity is provided by integer parameters which can change at each basic iteration and boolean parameters which can change even within the iteration. We have presented static analyses that ensure the liveness and the boundedness of BDPF graphs.

This year, we have focused on the *symbolic* analysis of parametric data-flow graphs. This work has been conducted within the RIPPES associated team.

6.2.4. Synthesis of switching controllers using approximately bisimilar multiscale abstractions

The use of discrete abstractions for continuous dynamics has become standard in hybrid systems design (see *e.g.*, [90] and the references therein). The main advantage of this approach is that it offers the possibility to leverage controller synthesis techniques developed in the areas of supervisory control of discrete-event systems [83]. The first attempts to compute discrete abstractions for hybrid systems were based on traditional systems behavioral relationships such as simulation or bisimulation, initially proposed for discrete systems most notably in the area of formal methods. These notions require inclusion or equivalence of observed behaviors which is often too restrictive when dealing with systems observed over metric spaces. For such systems, a more natural abstraction requirement is to ask for closeness of observed behaviors. This leads to the notions of approximate simulation and bisimulation introduced in [63].

These approaches are based on sampling of time and space where the sampling parameters must satisfy some relation in order to obtain abstractions of a prescribed precision. In particular, the smaller the time sampling parameter, the finer the lattice used for approximating the state-space; this may result in abstractions with a very large number of states when the sampling period is small. However, there are a number of applications where sampling has to be fast; though this is generally necessary only on a small part of the state-space. We have been exploring two approaches to overcome this state-space explosion.

We are currently investigating an approach using mode sequences of given length as symbolic states for our abstractions. By using mode sequences of variable length we are able to adapt the granularity of our abstraction to the dynamics of the system, so as to automatically trade off precision against controllability of the abstract states.

6.2.5. Typical Worst-Case Analysis of real-time systems

We focus on the problem of computing tight deadline miss models for real-time systems, which bound the number of potential deadline misses in a given sequence of activations of a task. In practical applications, such guarantees are often sufficient because many systems are in fact not hard real-time. Our major contribution this year is a general formulation of that problem in the context of systems where some tasks occasionally experience sporadic overload [26]. Based on this new formulation, we present an algorithm that can take into account fine-grained effects of overload at the input of different tasks when computing deadline miss bounds. We show in experiments with synthetic as well as industrial data that our algorithm produces bounds that are much tighter than in previous work, in sufficiently short time. In addition, we improve, in the preemptive case, the criterion proposed in [71] for establishing how much overload can be tolerated in a time window while still guaranteeing absence of deadline misses: our new criterion is a necessary and sufficient condition (as opposed to the sufficient condition of [71]) and therefore yields better results.

In parallel, we have developed an extension of sensitivity analysis for budgeting in the design of weakly-hard real-time systems. During design, it often happens that some parts of a task set are fully specified while other parameters, e.g. regarding recovery or monitoring tasks, will be available only much later. In such cases, sensitivity analysis can help anticipate how these missing parameters can influence the behavior of the whole system so that a resource budget can be allocated to them. It is, however, sufficient in many application contexts to budget these tasks in order to preserve weakly-hard rather than hard guarantees. We have thus developed an extension of sensitivity analysis for deriving task budgets for systems with hard and weakly-hard requirements.

We currently validate our approach on synthetic test cases and a realistic case study given by our partner Thales.

6.3. Language Based Fault-Tolerance

Participants: Dmitry Burlyaev, Pascal Fradet, Alain Girault, Yoann Geoffroy, Gregor Goessler, Jean-Bernard Stefani, Atena Abdi, Ismail Assayad.

6.3.1. Fault Ascription in Concurrent Systems

The failure of one component may entail a cascade of failures in other components; several components may also fail independently. In such cases, elucidating the exact scenario that led to the failure is a complex and tedious task that requires significant expertise.

The notion of causality *(did an event e cause an event e'?)* has been studied in many disciplines, including philosophy, logic, statistics, and law. The definitions of causality studied in these disciplines usually amount to variants of the counterfactual test "e is a cause of e' if both e and e' have occurred, and in a world that is as close as possible to the actual world but where e does not occur, e' does not occur either". In computer science, almost all definitions of logical causality — including the landmark definition of [70] and its derivatives — rely on a causal model that may not be known, for instance in presence of black-box components. For such systems, we have been developing a framework for blaming that helps us establish the causal relationship between component failures and system failures, given an observed system execution trace. The analysis is based on a formalization of counterfactual reasoning [7].

We have instantiated our approach to a synchronous data flow framework defined by a subset of the LUSTRE [69] language, and implemented the analysis in LOCA (see Section 5.2).

In [25] we have shown that we can improve precision of the analysis if (1) we can emulate execution of components instead of relying on their specifications, and (2) take into consideration input/output dependencies between components to avoid blaming components for faults induced by other components. We have demonstrated the utility of the extended analysis with a case study for a closed-loop patient-controlled analgesia system.

We have further proposed in [23] a general semantic framework for fault ascription. Our framework relies on configuration structures to handle concurrent systems, partial and distributed observations in a uniform way. It defines basic conditions for a counterfactual analysis of necessary and sufficient causes, and it presents a refined analysis that conforms to our basic conditions while avoiding various infelicities.

6.3.2. Tradeoff exploration between energy consumption and execution time

We have continued our work on multi-criteria scheduling, in two directions. First, in the context of dynamic applications that are launched and terminated on an embedded homogeneous multi-core chip, under execution time and energy consumption constraints, we have proposed a two layer adaptive scheduling method. In the first layer, each application (represented as a DAG of tasks) is scheduled statically on subsets of cores: 2 cores, 3 cores, 4 cores, and so on. For each size of these sets (2, 3, 4, ...), there may be only one topology or several topologies. For instance, for 2 or 3 cores there is only one topology (a "line"), while for 4 cores there are three distinct topologies ("line", "square", and "T shape"). Moreover, for each topology, we generate statically several schedules, each one subject to a different total energy consumption constraint, and consequently with a different Worst-Case Reaction Time (WCRT). Coping with the energy consumption constraints is achieved thanks to Dynamic Frequency and Voltage Scaling (DVFS). In the second layer, we use these pre-generated static schedules to reconfigure dynamically the applications running on the multi-core each time a new application is launched or an existing one is stopped. The goal of the second layer is to perform a dynamic global optimization of the configuration, such that each running application meets a pre-defined quality-of-service constraint (translated into an upper bound on its WCRT) and such that the total energy consumption be minimized. For this, we (1) allocate a sufficient number of cores to each active application, (2) allocate the unassigned cores to the applications yielding the largest gain in energy, and (3) choose for each application the best topology for its subset of cores (*i.e.*, better than the by default "line" topology). This is a joint work with Ismail Assayad (U. Casablanca, Morocco) who visited the team in September 2015.

Second, in the context of a static application (again represented a DAG of tasks) running on an homogeneous multi-core chip, we have worked on the static scheduling minimizing the WCRT of the application under the multiple constraints that the reliability, the power consumption, and the temperature remain below some given threshold. There are multiple difficulties: (1) the reliability is not an invariant measure w.r.t. time, which makes it impossible to use backtrack-free scheduling algorithms such as list scheduling [37]; to overcome this, we adopt instead the Global System Failure Rate (GSFR) as a measure of the system's reliability that is invariant with time [64]; (2) keeping the power consumption under a given threshold requires to lower the voltage and frequency, but this has a negative impact both on the WCRT and on the GSFR; keeping the GSFR below a given threshold requires to replicate the tasks on multiple cores, but this has a negative impact both on the WCRT, on the power consumption, and on the temperature; (3) keeping the temperature below a given threshold is even more difficult because the temperature continues to increase even after the activity stops, so each scheduling decision must be assessed not based on the current state of the chip (*i.e.*, the temperature of each core) but on the state of the chip at the end of the candidate task, and cooling slacks must be inserted. This is a joint work with Atena Abdi (Amirkabir U., Iran) who is a PhD visitor in the team.

6.3.3. Automatic transformations for fault tolerant circuits

In the past years, we have studied the implementation of specific fault tolerance techniques in real-time embedded systems using program transformation [1]. We are now investigating the use of automatic transformations to ensure fault-tolerance properties in digital circuits. To this aim, we consider program transformations for hardware description languages (HDL). We consider both single-event upsets (SEU) and single-event transients (SET) and fault models of the form *"at most 1 SEU or SET within n clock cycles"*.

We have proposed novel fault-tolerance transformations based on time-redundancy. In particular, we have presented a transformation using double-time redundancy (DTR) coupled with micro-checkpointing, rollback and a speedup mode [19]. The approach is capable to mask any SET every 10 cycles and keeps the same input/output behavior regardless error occurrences. Usually transparent masking requires triple redundancy and voting. Experimental results on the ITC'99 benchmark suite indicate that the hardware overhead of DTR is 2.7 to 6.1 times smaller than full TMR with a double loss in throughput. The method does not require any specific hardware support and is an interesting alternative to Triple Modular Redundancy (TMR) for logic intensive designs.

We have also designed a transformation that allows the circuit to change its level of time-redundancy. This feature allows the circuit to dynamically and temporarily lower (resp. increase) fault-tolerance and speed up (resp. slow down) its computation without interruption [20]. The motivations for such changes can be based on the current radiation environment or the processing of critical data. When hardware size is limited and fault-tolerance is only occasionally needed, that scheme is a better choice than static TMR, which involves a constant high area overhead

These time redundancy transformations (DTR and adaptive fault-tolerance) have been patented [50]

We have described how to formally certify fault-tolerant transformations using the COQ proof assistant [53] (see Section 5.3). The transformations are described on a simple gate-level hardware description language LDDL (Low-level Dependent Description Language). This combinator language is equiped with dependent types and ensures that circuits are well-formed by construction (gates correctly plugged, no dangling wires, no combinational loops, ...). Fault-models are specified in the operational semantics of the language. The main theorem states that, for any circuit, for any input stream and for any SET allowed by the fault-model, its transformed version produces a correct output [18]. The primary motivation of this work was to certify DTR whose intricacy requested a formal proof to make sure that no single-point of failure existed. We have first applied this approach to the correctness proofs of TMR, TTR (Triple Time Redundancy) and finally DTR.

This research is part of Dmitry Burlyaev's pHD thesis [11] defended in November 2015.

6.3.4. A formal approach for the synthesis and implementation of fault-tolerant embedded systems

We have been working for several years on the usage of discrete controller synthesis (DCS) [83] to provide the automated addition of fault-tolerance in embedded systems with formal guarantees [65]. The first key idea is that the initial system model (usually an LTS) includes both the expected behaviors, the unexpected ones (that is, the failures), and the reconfigurations (typically repair actions). The second key idea is that the failures are modeled as *uncontrollable* events. Then, thanks to an exhaustive state space traversal, DCS is able to generate a *controller* that will prevent the system from entering a “bad” state (*e.g.*, a configuration of the system where a task is active on a faulty processor). From the point of view of fault-tolerance, this approach combines the advantages of static guarantees with that of dynamic reconfiguration (hence without the penalty of static redundancy).

Through this new work, we have demonstrated the feasibility of a complete workflow to synthesize and implement correct-by-construction fault tolerant distributed embedded systems consisting of real-time periodic tasks [24]. Correct-by-construction is provided by the use of DCS, which allows us to guarantee that the synthesized controlled system guarantees the functionality of its tasks even in the presence of processor failures. For this step, our workflow uses the HEPTAGON domain specific language [58] and the SIGALI DCS tool [79]. The correct implementation of the resulting distributed system is a challenge, all the more since the controller itself must be tolerant to the processor failures. We achieve this step thanks to the libDGALS real-time library [89] (1) to generate the glue code that will migrate the tasks upon processor failures, maintaining their internal state through migration, and (2) to make the synthesized controller itself fault-tolerant.

TEA Project-Team

7. New Results

7.1. Polychronous automata

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

We have defined a model of *polychronous automata* based on clock relations [13]. A specificity of this model is that an automaton is submitted to clock constraints: these finite-state automata define transition systems to express explicit reactions together with properties, in the form of Boolean formulas over logical time, to constrain their behavior. This allows one to specify a wide range of control-related configurations, either reactive, or restrictive with respect to their control environment. A semantic model is defined for these polychronous automata, that relies on a Boolean algebra of clocks. Polychronous automata integrate smoothly with data-flow equations in the polychronous model of computation.

This formal model of automata also supports the recommendations adopted by the SAE committee on the AADL to implement a timed and synchronous behavioural annex for the standard [0].

A minimal syntactic extension of the Signal language has been defined to integrate polychronous automata in Polychrony. We have added a new syntactic category of *process*, called `automaton`. In such an automaton process, labeled processes represent states, and generic processes such as `Transition` are used to represent the automaton features. Usual equations can be used in these automaton processes to specify constraints or to define computations.

We have also defined and implemented the refinement of Signal processes as automata. A given Signal program may be seen as an automaton which contains one single state and one single transition, labeled by a clock. This clock is the upper bound of all the clocks of the program (the *tick* of the program). The construction of a refined automaton from a Signal program is based on delayed signals, viewed as state variables (in particular Boolean ones). A state of the automaton is a Signal program with some valuation of its state variables. Transitions are labeled by clocks, which represent the events that fire these transitions. The principle of the construction consists in dividing a given state according to the possible values of a state variable (i.e., *true* and *false* for Boolean state variables) in order to get two states, and thus two new Signal programs. Each one of these two states is obtained using a rewriting of the starting program. Moreover, the absence of value for the state variable (which can be considered as another possible value) is taken into account in the clocks labelling the transitions. The construction of the automaton is a hierarchic process. Thanks to the clock hierarchy, this construction, which would be expensive in the worst case (the size of the explicit automaton being an exponential of its number of state variables), may be heavily simplified.

7.2. Runtime verification and trace analysis

Participants: Vania Joloboff, Daian Yue, Frédéric Mallet.

When engineers design a new cyber physical system, there are well known requirements that can be translated as system properties that must be verified. These properties can be expressed in some formalism and when the model has been designed, the properties can be checked at the model level, using model checking techniques or other model verification techniques. When building a virtual prototype of the system, including a combination of simulated hardware, firmware and application software, the executable models can be augmented also with property verification, for example in the PSL language, or simply by introducing assertions in the implementation code.

[0] *Logically timed specification in the AADL: a synchronous model of computation and communication (recommendations to the SAE committee on AADL).* L. Besnard, E. Borde, P. Dissaux, T. Gautier, P. Le Guernic, and J.-P. Talpin. Technical Report RT-0446, Inria, 2014.

This requires that the properties are well specified at the time the virtual prototype is assembled. However it is also the case that many intrinsic properties are actually unforeseen when the virtual prototype is assembled, for example that some hardware buffer overflow should not remain unnoticed by the software. In most cases, during system design the simulation fails: the engineers then must investigate the cause of the failure. Most of time the failure is due to an unexpected sequence of states and transitions that involve several components mixing hardware and software that could not be checked at the model level (e.g. state explosion) or was simply unforeseen. The engineers then have to investigate the cause of failure.

A widely used technique for that consists in storing all of the trace data of simulation sessions into trace files, which are analyzed later with specialized trace analyzer tools. Such trace files have become huge, possibly hundred of Gigabytes as all data are stored into the trace files, and have become untractable by human manual handling. The engineers use some kind of search tools to identify the cause of failure and after iterative refinement steps, which are very time consuming, eventually identify the reason, most often some unforeseen causality chain of events and state transitions that lead to a failure. A new system property can then be captured and included into the set of verified properties.

In order to better identify the reason for such failures and capture the missing properties that the system should verify we have started to work on a new run time verification approach based on trace analysis. Approaches like PSL requires that the properties to verify are known before hand. Our approach is attempting for the engineers to experiment various property verification of failing simulations without re-building the virtual prototype. We are investigating a technique for trace analysis that makes it possible to investigate properties either statically working from a trace file or dynamically by introducing a dynamic verification component into the virtual prototype.

The first idea is to introduce a formal mapping/filtering technique such that the raw data generated by a virtual prototype can be mapped onto a formal trace model. For that, we propose to use a model transformer whose code is generated from a higher level. Using the Eclipse modelling framework, we propose for the virtual prototyping engineers to first describe using a Domain Specific Language how the raw output of the simulator can be filtered and mapped to a formal model. This Domain Specific Language takes as input the description of the simulator output, and the description of the formal output, following fixed meta models. In current version, the meta model of the virtual prototype dictates that it generates 'trace items' where each trace item is specified as a sequence of identified binary data variables (bits, bytes, words..) that carry a timestamp.

The model transformer generates code (in our case C++) that is dynamically invoked by the virtual prototype to dynamical map the trace output. An advantage of doing that is that all irrelevant data with regards to a tested property can be ignored and the size of trace files can be considerably reduced. For our experiment, we have chosen logical clock CCSL as our formal target formalism. The Eclipse EMF tool we have defined allows users to define a mapping model from the local simulation events from the SimSoC simulator to a logical clock format.

The second idea is to hide the complexity of the formal method formulas into a user friendly property specification language. For example, we do not want to expose the end-users engineers to understand the intricacies of CCSL or LTL. The property specification language is translated into CCSL formulas, which in turn generate automata. It should be possible then, to some extent, to change the formalism underneath the language without changing the properties expressed by the user.

The property specification language ultimately compiles into automata that parse the formal trace output generated above. At runtime of the virtual prototype, the mapping library is dynamically loaded by the simulator and generates input for the automata. The verification of the properties can be dynamic, with a true runtime verification, or statically by analyzing the (much smaller) trace file after a failure.

This year we have investigated this approach, designed the architecture described above and carried some experimental work, but a significant part of the implementation still remains to be done. We have started designing a new property specification language where the users can express properties such as causality (e.g. the train must not start if the door is opened) or jittering or clock drift in image processing [11], [10]. There remain some theoretical issues with regards to which properties can be effectively verified.

7.3. Integration of Polychrony with QGen

Participants: Christophe Junke, Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

The FUI project P gave birth to the QGen qualifiable model compiler, developed by Adacore. The tool accepts a discrete subset of Simulink expressed in a language called P and produces C or Ada code. It is currently not known if an architectural description language is going to be integrated in QGen, as originally planned.

We developed a transformation tool named P2S for expressing P system models in Signal, using the EMF (Eclipse Modelling Framework) technology. P2S tool is written in Clojure, a language inspired by Lisp running on the Java Virtual Machine, which helped us define a terse and expressive API for manipulating Signal models while remaining fully interoperable with existing Java libraries (including Eclipse plugins and especially Polychrony ones).

We experimented this transformation tool on small to medium use cases provided by members of the P project. Our work is detailed in a conference paper titled "Integration of Polychrony and QGen Model Compiler", which will appear at ERTSS'16 [0]. A perspective of our work is to convert the intermediate code emitted by QGen as Signal too (under development), in order to produce a fully executable Signal model of Simulink models, and combine them with architectural description of systems in AADL, and/or P's architecture language.

7.4. Formal semantics and model-based analysis of AADL specifications

Participants: Loïc Besnard, Etienne Borde, Thierry Gautier, Paul Le Guernic, Clément Guy, Jean-Pierre Talpin, Huafeng Yu.

Last year, the SAE committee on the AADL adopted our recommendations to implement a timed and synchronous behavioural annex for the standard. We have defined a new model of polychronous constrained automata that has been provided as semantic model for our proposal of an extension of the AADL behavioural annex. An experimental implementation of the semantic features of this "timing annex" will be provided through the Polychrony framework. For that purpose, representations of automata have been introduced in the Signal toolbox of Polychrony. The implementation will enrich the already existing transformation from AADL models to Signal programs to consider behaviour of AADL models, and will be integrated in the POP environment for Eclipse. The transformation from AADL behaviour annex to Signal programs use the Signal extension for polychronous automata, which are used as the common semantic domain. The implementation is currently tested with the adaptive cruise control case study developed with Toyota ITC.

Our work with the SAE committee is sponsored by Toyota, with whom we started a new project in 2014 jointly with VTRL as US partner. The main topic of our project is the semantic-based model integration of automotive architectures, virtual integration, toward formal verification and automated code synthesis [19]. The project led to the elaboration of a case study of an adaptive cruise control system, supported through an AADL implementation and a video of demonstration. The case study implementation is an AADL model representing the whole adaptive cruise control system, from car devices (e.g., brakes, throttle or radar) to software behavior, including embedded hardware (buses, processors and memories). It will be used in the future to demonstrate property and constraint analyses through heterogeneous systems. Huafeng Yu, our main collaborator at Toyota ITC, presented the video of demonstration at the annual Toyota show case. Early returns from the show case express a growing interest of Toyota for architecture and timing of car embedded systems, which could lead to new collaborations.

7.5. Refinement types for reactive system models

Participants: Pierre Jouvelot, Sandeep Shukla, Jean-Pierre Talpin.

[0] *Integration of polychrony in the QGen model compiler*. C. Junke, T. Gautier, L. Besnard, J.-P. Talpin. ERTS'16 - European Congress on Embeddd Real-Rime Software and Systems, 2016.

We introduced a new technique born from the field of functional programming to adapt and extend it to the case of reaction systems, the notion of refinement types of Jahla et al. [0]. Our idea is to formulate the analysis of algebraic properties in synchronous and reactive programs as data-dependent type properties formulated using multi-sorted logic formulas, which we call liquid clocks [20], [18]. Our objectives are to cover the case of several models of concurrency and computation: synchronous, asynchronous, data-parallel; as well as to formulate such algebraic properties for linear, continuous and logical forms of time, all into the same type-theoretical framework. This work, born from two collaborations With USAF/VT and with the ANR Feever project, will be pursued within the TIX international partnership.

7.6. Formal verification of timing aspects of cyber-physical systems using a contract theory

Participants: Jean-Pierre Talpin, Benoit Boyer, David Mentre, Simon Lunel.

This is a new project in collaboration with Mitsubishi Electronics Research Centre Europe (MERCE). The primary goal of our project is to ensure correctness-by-design in cyber-physical systems, i.e., systems that mix software and hardware in a physical environment, e.g., Mitsubishi factory automation lines. We plan to explore a multi-sorted algebraic framework for static analysis and formal verification starting from a simple use case extracted from Mitsubishi factory automation documentations. This will serve as a basis to more ambitious research where we intend to leverage recent advance in type theory, SMT solvers for nonlinear real arithmetic (dReal and δ-decidability) and contracts theory (meta-theory of Benveniste et al., Ruchkin's contracts) to provide a general framework of reasoning about heterogeneous factory components.

[0] *Liquid Types*. P. M. Rondon, M. Kawaguchi, R. Jhala. PLDI, 2008

ANTIQUE Project-Team

6. New Results

6.1. Memory Abstraction

6.1.1. Abstraction of arrays based on non contiguous partitions

Participants: Jiangchao Liu, Xavier Rival [correspondant].

Abstract interpretation, Memory abstraction, Array abstract domains. In [19], we studied array abstractions.

Array partitioning analyses split arrays into contiguous partitions to infer properties of cell sets. Such analyses cannot group together non contiguous cells, even when they have similar properties. We proposed an abstract domain which utilizes semantic properties to split array cells into groups. Cells with similar properties will be packed into groups and abstracted together. Additionally, groups are not necessarily contiguous. This abstract domain allows to infer complex array invariants in a fully automatic way. Experiments on examples from the Minix 1.1 memory management demonstrated its effectiveness.

6.1.2. Static analysis for unstructured sharing

Participants: Huisong Li, Bor-Yuh Evan Chang [University of Colorado, Boulder, USA], Xavier Rival [correspondant].

Abstract interpretation, Memory abstraction, Separation logic. In [18], we studied the abstraction of shared data-structures.

Shape analysis aims to infer precise structural properties of imperative memory states and has been applied heavily to verify safety properties on imperative code over pointer-based data structures. Recent advances in shape analysis based on separation logic has leveraged summarization predicates that describe unbounded heap regions like lists or trees using inductive definitions. Unfortunately, data structures with *unstructured sharing*, such as graphs, are challenging to describe and reason about in such frameworks. In particular, when the sharing is unstructured, it cannot be described inductively in a local manner. In this work, we proposed a global abstraction of sharing based on set-valued variables that when integrated with inductive definitions enables the specification and shape analysis of structures with unstructured sharing.

6.1.3. Synthesizing short-circuiting validation of data structure invariants

Participants: Yi-Fan Tsai, Devin Coughlin, Bor-Yuh Evan Chang [University of Colorado, Boulder, USA], Xavier Rival [correspondant].

In [28], we studied the synthesis of short-circuiting validators for data-structure invariants.

This work introduces *incremental verification-validation*, a novel approach for checking rich data structure invariants expressed as separation logic assertions. Incremental verification-validation combines static verification of separation properties with efficient, *short-circuiting* dynamic validation of arbitrarily rich data constraints. A data structure invariant checker is an inductive predicate in separation logic with an executable interpretation; a short-circuiting checker is an invariant checker that stops checking whenever it detects at *run time* that an assertion for some sub-structure has been fully proven *statically*. At a high level, our approach does two things: it statically proves the separation properties of data structure invariants using a static shape analysis in a standard way but then leverages this proof in a novel manner to synthesize short-circuiting dynamic validation of the data properties. As a consequence, this approach enables dynamic validation to make up for imprecision in sound static analysis while simultaneously leveraging the static verification to make the remaining dynamic validation efficient. This work has shown empirically that short-circuiting can yield asymptotic improvements in dynamic validation, with low overhead over no validation, even in cases where static verification is incomplete.

6.2. Abstract domains

6.2.1. Abstract domains and solvers for set reasoning

Participants: Arlen Cox, Bor-Yuh Evan Chang [University of Colorado, Boulder, USA], Huisong Li, Xavier Rival [correspondant].

In [15], we studied the abstraction and inferrence of set properties.

When constructing complex program analyses, it is often useful to reason about not just individual values, but collections of values. Symbolic set abstractions provide building blocks that can be used to partition elements, relate partitions to other partitions, and determine the provenance of multiple values, all without knowing any concrete values. To address the simultaneous challenges of scalability and precision, we formalized and implemented an interface for symbolic set abstractions and constructed multiple abstract domains relying on both specialized data structures and off-the-shelf theorem provers. We developed techniques for lifting existing domains to improve performance and precision. We evaluated these domains on real-world data structure analysis problems.

6.2.2. Abstraction of optional numerical values

Participants: Jiangchao Liu, Xavier Rival [correspondant].

In [20], we designed a functor to lift a numerical abstract domain into an abstract domain that accounts for *optional* numerical values.

We proposed a technique to describe properties of numerical stores with optional values, that is, where some variables may have no value. Properties of interest include numerical equalities and inequalities. Our approach lifts common linear inequality based numerical abstract domains into abstract domains describing stores with optional values. This abstraction can be used in order to analyze languages with some form of *option* scalar type. It can also be applied to the construction of abstract domains to describe complex memory properties that introduce symbolic variables, e.g., in order to summarize unbounded sets of program variables, and where these symbolic variables may be undefined, as in some array or shape analyses. We described the general form of abstract states, and propose sound and automatic static analysis algorithms. We evaluated our construction in the case of an array abstract domain.

6.3. Static analysis of JavaScript applications

6.3.1. Desynchronized multi-state abstractions for open programs in dynamic languages

Participants: Arlen Cox [correspondant], Bor-Yuh Evan Chang [University of Colorado, Boulder, USA], Xavier Rival.

Abstract interpretation, Dynamically typed languages, Verification In [16], we have studied desynchronized multi-state abstractions for open programs in dynamic languages (libraries).

Dynamic language library developers face a challenging problem: ensuring that their libraries will behave correctly for a wide variety of client programs without having access to those client programs. This problem stems from the common use of two defining features for dynamic languages: callbacks into client code and complex manipulation of attribute names within objects. To remedy this problem, we introduced two state-spanning abstractions. To analyze callbacks, the first abstraction desynchronizes a heap, allowing partitions of the heap that may be affected by a callback to an unknown function to be frozen in the state prior to the call. To analyze object attribute manipulation, building upon an abstraction for dynamic language heaps, the second abstraction tracks attribute name/value pairs across the execution of a library. We implemented these abstractions and use them to verify modular specifications of class-, trait-, and mixin-implementing libraries.

6.4. Static analysis of spreadsheet applications

Participants: Tie Cheng [correspondant], Xavier Rival.

Abstract interpretation, Spreadsheet applications, Verification In [14], we have proposed a static analysis to detect type unsafe operations in spreadsheet applications including formulas and macros.

Spreadsheets are widely used, yet are error-prone: they use a weak type system, allowing certain operations that will silently return unexpected results, like comparisons of integer values with string values. However, discovering these issues is hard, since data and formulas can be dynamically set, read or modified. We defined a static analysis that detects all run-time type-unsafe operations in spreadsheets. It is based on an abstract interpretation of spreadsheet applications, including spreadsheet tables, global re-evaluation and associated programs. Our implementation supports the features commonly found in real-world spreadsheets. We ran our analyzer on the EUSES Spreadsheet Corpus. This evaluation shows that our tool is able to automatically verify a large number of real spreadsheets, runs in a reasonable time and discovers complex bugs that are difficult to detect by code review or by testing.

6.5. Distributed systems verification and programming language

Participants: Cezara Drăgoi [correspondant], Thomas Henzinger [IST Austria, Austria], Damien Zufferey [MIT, CSAIL, USA].

Fault-tolerant distributed systems, Programming languages, Verification Fault-tolerant distributed algorithms play an important role in many critical/high-availability applications. These algorithms are notoriously difficult to implement correctly, due to asynchronous communication and the occurrence of faults, such as the network dropping messages or computers crashing. Noteworthy is the lack of automated verification techniques for distributed systems, highly contrasting the mass distribution and development of distributed software. Therefore, our main motivation is to increase the confidence we have in distributed systems using formal verification methods. However, due to the complexity distributed systems have reached, we believe it is no longer realistic nor efficient to assume that high level specifications can be proved when development and verification are two disconnected steps in the software production process. We think that the difficulty does not only come from the algorithms but from the way we think about distributed systems. Therefore, we are interested in finding an appropriate programming model for fault-tolerant distributed algorithms, that increases the confidence we have distributed software. We introduced PSYNC, a domain specific language based on the Heard-Of model, which views asynchronous faulty systems as synchronous ones with an adversarial environment that simulates asynchrony and faults by dropping messages. We defined a runtime system for PSYNC that efficiently executes on asynchronous networks. We formalize the relation between the runtime system and PSYNC in terms of observational refinement. PSYNC introduces a high-level lockstep abstraction (on top of the standard asynchronous semantics), which simplifies the design and implementation of fault-tolerant distributed algorithms and enables automated formal verification. We have implemented an embedding of PSYNC in the SCALA programming language with a runtime system for asynchronous networks. We showed the applicability of PSYNC by implementing several important fault-tolerant distributed algorithms and we compared the implementation of consensus algorithms in PSYNC against implementations in other languages in terms of code size, runtime efficiency, and verification.

6.6. Derivation of Qualitative Dynamical Models from Biochemical Networks

Participants: Wassim Abou-Jaoudé [IBENS], Jérôme Feret [correspondant], Denis Thieffry [IBENS].

Systems biology, Logical models, Automatic derivation As technological advances allow a better identification of cellular networks, more and more molecular data are produced allowing the construction of detailed molecular interaction maps. One strategy to get insights into the dynamical properties of such systems is to derive compact dynamical models from these maps, in order to ease the analysis of their dynamics.

Starting from a case study, we present in [13] a methodology for the derivation of qualitative dynamical models from biochemical networks. Properties are formalized using abstract interpretation. We first abstract states and traces by quotienting the number of instances of chemical species by intervals. Since this abstraction is too coarse to reproduce the properties of interest, we refine it by introducing additional constraints. The resulting abstraction is able to identify the dynamical properties of interest in our case study.

6.7. Annotation of rule-based models with formal semantics to enable creation, analysis, reuse and visualization

Participants: G. Misirli, M. Cavaliere, W. Waites, M. Pocock, C. Madsen, O. Gifellon, R. Honorato-Zimmer, P. Zuliani, V. Danos [correspondant], A. Wipat.

In [35] We present an annotation framework and guidelines for annotating rule-based models, encoded in the commonly used Kappa and BioNetGen languages. Biological systems are complex and challenging to model and therefore model reuse is highly desirable. To promote model reuse, models should include both information about the specifics of simulations and the underlying biology in the form of metadata. The availability of computationally tractable metadata is especially important for the effective automated interpretation and processing of models. Metadata are typically represented as machine-readable annotations which enhance programmatic access to information about models. Rule-based languages have emerged as a modelling framework to represent the complexity of biological systems. Annotation approaches have been widely used for reaction-based formalisms such as SBML. However, rule-based languages still lack a rich annotation framework to add semantic information, such as machine-readable descriptions, to the components of a model. We introduced an annotation framework and guidelines for annotating rule-based models, encoded in the commonly used Kappa and BioNetGen languages. We adapted widely adopted annotation approaches to rule-based models. We initially proposed a syntax to store machine-readable annotations and describe a mapping between rule-based modelling entities, such as agents and rules, and their annotations. We then described an ontology to both annotate these models and capture the information contained therein, and demonstrate annotating these models using examples. Finally, we presented a proof of concept tool for extracting annotations from a model that can be queried and analyzed in a uniform way. The uniform representation of the annotations can be used to facilitate the creation, analysis, reuse and visualization of rule-based models. Although examples are given, using specific implementations the proposed techniques can be applied to rule-based models in general.

6.8. Quantitative genomic analysis of RecA protein binding during DNA double-strand break repair reveals RecBCD action in vivo

Participants: Charlotte Cockram, Milana Filatenkova, Vincent Danos [correspondant], Meriem Karoui, Leach David.

Understanding molecular mechanisms in the context of living cells requires the development of new methods of in vivo biochemical analysis to complement established in vitro biochemistry. A critically important molecular mechanism is genetic recombination, required for the beneficial reassortment of genetic information and for DNA double-strand break repair (DSBR). Central to recombination is the RecA (Rad51) protein that assembles into a spiral filament on DNA and mediates genetic exchange. Here we developed a method that combines chromatin immunoprecipitation with next-generation sequencing (ChIP-Seq) and mathematical modeling to quantify RecA protein binding during the active repair of a single DSB in the chromosome of Escherichia coli. In [29] we have used quantitative genomic analysis to infer the key in vivo molecular parameters governing RecA loading by the helicase/ nuclease RecBCD at recombination hot-spots, known as Chi. Our genomic analysis has also revealed that DSBR at the lacZ locus causes a second RecBCD-mediated DSBR event to occur in the ter- minus region of the chromosome, over 1 Mb away.

6.9. Moment Semantics for Reversible Rule-Based Systems

Participants: Vincent Danos [correspondant], Tobias Hinder, Ricardo Honorato-Zimmer, Sandro Stuck.

In [34] we developed a notion of stochastic rewriting over marked graphs – i.e. directed multigraphs with degree constraints. The approach is based on double-pushout (DPO) graph rewriting. Marked graphs are expressive enough to internalize the 'no-dangling-edge' condition inherent in DPO rewriting. Our main result is that the linear span of marked graph occurrence-counting functions – or motif functions – form an algebra which is closed under the infinitesimal generator of (the Markov chain associated with) any such rewriting system. This gives a general procedure to derive the moment semantics of any such rewriting system, as a countable (and recursively enumerable) system of differential equations indexed by motif functions. The differential system describes the time evolution of moments (of any order) of these motif functions under the rewriting system. We illustrate the semantics using the example of preferential attachment networks; a well-studied complex system, which meshes well with our notion of marked graph rewriting. We show how in this case our procedure obtains a finite description of all moments of degree counts for a fixed degree.

6.10. Dirichlet is Natural

Participants: Vincent Danos [correspondant], Ilias Garnier.

In [32] the authors reconstruct a family of higher-order probabilities known as the Dirichlet process.

Giry and Lawvere's categorical treatment of probabilities, based on the probabilistic monad G, offer an elegant and hitherto unexploited treatment of higher-order probabilities. The goal of this paper is to follow this formulation to reconstruct a family of higher-order probabilities known as the Dirichlet process. This family is widely used in non-parametric Bayesian learning.

Given a Polish space X, we build a family of higher-order probabilities in G(G(X)) indexed by M(X), the set of non-zero finite measures over X. The construction relies on two ingredients. First, we develop a method to map a zero-dimensional Polish space X to a projective system of finite approximations, the limit of which is a zero-dimensional compactification of X. Second, we use a functorial version of Bochner's probability extension theorem adapted to Polish spaces, where consistent systems of probabilities over a projective system give rise to an actual probability on the limit. These ingredients are combined with known combinatorial properties of Dirichlet processes on finite spaces to obtain the Dirichlet family on X. We prove that the Dirichlet family is a natural transformation from the monad M to GG over Polish spaces, which in particular is continuous in its parameters. This is an improvement on extant constructions of Dirichlet.

6.11. Mechanistic links between cellular trade-offs, gene expression, and growth

Participants: Andrea Weisse, Diego Oyarzun, Vincent Danos [correspondant], Peter Swain.

Intracellular processes rarely work in isolation but continually interact with the rest of the cell. In microbes, for example, we now know that gene expression across the whole genome typically changes with growth rate. The mechanisms driving such global regulation, however, are not well understood. In [36] we considered three trade-offs that, because of limitations in levels of cellular energy, free ribosomes, and proteins, are faced by all living cells and we construct a mechanistic model that comprises these trade-offs. Our model couples gene expression with growth rate and growth rate with a growing population of cells. We show that the model recovers Monod's law for the growth of microbes and two other empirical relationships connecting growth rate to the mass fraction of ribosomes. Further, we can explain growth-related effects in dosage compensation by paralogs and predict host–circuit interactions in synthetic biology. Simulating competitions between strains, we find that the regulation of metabolic pathways may have evolved not to match expression of enzymes to levels of extracellular substrates in changing environments but rather to balance a trade-off between exploiting one type of nutrient over another. Although coarse-grained, the trade-offs that the model embodies are fundamental, and, as such, our modeling framework has potentially wide application, including in both biotechnology and medicine.

6.12. Thermodynamic graph-rewriting

Participants: Vincent Danos [correspondant], Russell Harmer, Ricardo Honorato-Zimmer.

In [33] we developped a new thermodynamic approach to stochastic graph-rewriting. The ingredients are a finite set of reversible graph-rewriting rules called generating rules, a finite set of connected graphs P called energy patterns and an energy cost function. The idea is that the generators define the qualitative dynamics, by showing which transformations are possible, while the energy patterns and cost function specify the long-term probability π of any reachable graph. Given the generators and energy patterns, we construct a finite set of rules which (i) has the same qualitative transition system as the generators; and (ii) when equipped with suitable rates, defines a continuous-time Markov chain of which π is the unique fixed point. The construction relies on the use of site graphs and a technique of 'growth policy' for quantitative rule refinement which is of independent interest. This division of labour between the qualitative and long-term quantitative aspects of the dynamics leads to intuitive and concise descriptions for realistic models (see the examples in S4 and S5). It also guarantees thermodynamical consistency (AKA detailed balance), otherwise known to be undecidable, which is important for some applications. Finally, it leads to parsimonious parameterizations of models, again an important point in some applications.

6.13. Kappa Rule-Based Modelling in Synthetic Biology

Participants: John Wilson-Kanamori, Vincent Danos [correspondant], Ty Thomson, Ricardo Honorato-Zimmer.

This [37] is a chapter of a book that provides complete coverage of the computational approaches currently used in Synthetic Biology. Rule-based modeling, an alternative to traditional reaction-based modeling, allows us to intuitively specify biological interactions while abstracting from the underlying combinatorial complexity. One such rule-based modeling formalism is Kappa, which we introduce to readers in this chapter. We discuss the application of Kappa to three modeling scenarios in synthetic biology: a unidirectional switch based on nitrosylase induction in Saccharomyces cerevisiae, the repressilator in Escherichia coli formed from BioBrick parts, and a light-mediated extension to said repressilator developed by the University of Edinburgh team during iGEM 2010. The second and third scenarios in particular form a case-based introduction to the Kappa BioBrick Framework, allowing us to systematically address the modeling of devices and circuits based on BioBrick parts in Kappa. Through the use of these examples, we highlight the ease with which Kappa can model biological interactions both at the genetic and the protein–protein interaction level, resulting in detailed stochastic models accounting naturally for transcriptional and translational resource usage. We also hope to impart the intuitively modular nature of the modeling processes involved, supported by the introduction of visual representations of Kappa models. Concluding, we explore future endeavors aimed at making modeling of synthetic biology more user-friendly and accessible, taking advantage of the strengths of rule-based modeling in Kappa.

This Chapters focus on computational methods and algorithms for the design of bio-components, insight on CAD programs, analysis techniques, and distributed systems. Written in the highly successful Methods in Molecular Biology series format, the chapters include the kind of detailed description and implementation advice that is crucial for getting optimal results in the laboratory.

Authoritative and practical, Computational Methods in Synthetic Biology serves as a guide to plan in silico the in vivo or in vitro construction of a variety of synthetic bio-circuits.

CELTIQUE Project-Team

6. New Results

6.1. Certified compilation

We thrive at improving the technology of certified compilation. Our work builds on the infrastructure provided by the CompCert compiler. We are working both at improving the guarantees provided by certified compilation and at formalising state-of-the-art optimisation techniques.

6.1.1. Safer CompCert

Participants: Sandrine Blazy, Frédéric Besson, Pierre Wilke.

The CompCert compiler is proved with respect to an abstract semantics. In previous work [52], we propose a more concrete memory model for the CompCert compiler [68]. This model gives a semantics to more programs and lift the assumption about an infinite memory. This model makes CompCert safer because more programs are captured by the soundness theorem of CompCert and because it allows to reason about memory consumption.

We are investigating the consequences this model on different compiler passes of CompCert [32]. As a sanity check, we prove formally that the existing memory model is an abstraction of our more concrete model thus validating formally the soundness of CompCert's abstract semantics of pointers. We have also port the front-end of the compiler to our new semantics and are working on the compiler back-end.

6.1.2. Verification of optimization techniques

Participants: Sandrine Blazy, Delphine Demange, Yon Fernandez de Retana, David Pichardie.

The CompCert compiler foregoes using SSA, an intermediate representation employed by many compilers that enables writing simpler, faster optimizers. In previous work [51], we have proposed a formally verified SSA-based middle-end for CompCert, addressing two problems raised by Leroy in 2009: giving an intuitive formal semantics to SSA, and leveraging its global properties to reason locally about program optimizations. Since then, we have studied in more depth the SSA-based optimization techniques with the objective to make the middle-end more realistic, in terms of the efficiency of optimizations, and to rationalize the way the correctness proofs of optimizations are conducted and structured.

First, we have studied in [34] the problem of a verified, yet efficient (i.e. as implemented in production compilers) technique for testing the dominance relation between two nodes in a control flow graph. We propose a formally verified validator of untrusted dominator trees, on top of which we implement and prove correct a fast dominance test.

Second, in [20], we implement and verify two prevailing SSA optimizations (Sparse Conditional Constant Propagation and Global Value Numbering), conducting the proofs in a unique and common sparse optimization proof framework, factoring out many of the dominance-based reasoning steps required in proofs of SSA-based optimizations. Our experimental evaluations indicate both a better precision, and a significant compilation time speedup.

Finally, we have studied (paper under review at the international conference Compiler Construction 2016) the destruction of the SSA form (i.e. at the exit point of the middle-end), a problem that has remained a difficult problem, even in a non-verified environment. We formally defined and proved the properties of the generation of Conventional SSA (CSSA) which make its destruction simple to implement and prove. We implemented and proved correct a coalescing destruction of CSSA, à la Boissinot et al., where variables can be coalesced according to a refined notion of interference. Our CSSA-based, coalescing destruction allows us to coalesce more than 99% of introduced copies, on average, and leads to encouraging results concerning spilling and reloading during post-SSA allocation.

6.2. Certified Static Analyses

6.2.1. Certified Analyses for JavaScript

Participants: Martin Bodin, Thomas Jensen, Alan Schmitt.

We have continued our work on the certification of analyses for JavaScript by developing a systematic way to build certified abstract interpreters from big-step semantics using the Coq proof assistant. We based our approach on Schmidt's abstract interpretation principles for natural semantics, and used a pretty-big-step (PBS) semantics, a semantic format proposed by Charguéraud. We proposed a systematic representation of the PBS format and implemented it in Coq. We then showed how the semantic rules can be abstracted in a methodical fashion, independently of the chosen abstract domain, to produce a set of abstract inference rules that specify an abstract interpreter. We proved the correctness of the abstract interpreter in Coq once and for all, under the assumption that abstract operations faithfully respect the concrete ones. We finally showed how to define correct-by-construction analyses: their correction amounts to proving they belong to the abstract semantics. This work has been published at CPP 2015 [19].

In addition, we have worked on hybrid typing of information flow for JavaScript, in collaboration with José Fragoso Santos and Tamara Rezk at Inria Sophia-Antipolis. Our analysis combined static and dynamic typing in order to avoid rejecting programs due to imprecise typing information. This work has been published at TGC 2015 [21].

6.2.2. Certified Analyses for safety-critical C programs

Participants: Sandrine Blazy, Vincent Laporte, David Pichardie.

We designed and proved sound, using the Coq proof assistant, a static analyzer, Verasco [26], based on abstract interpretation for most of the ISO C 1999 language (excluding recursion and dynamic allocation). Verasco establishes the absence of run-time errors in the analyzed programs. It enjoys a modular architecture that supports the extensible combination of multiple abstract domains, both relational and non-relational. Verasco integrates with the CompCert formally-verified C compiler so that not only the soundness of the analysis results is guaranteed with mathematical certitude, but also the fact that these guarantees carry over to the compiled code.

6.2.3. Certified Analyses for binary codes

Participants: Sandrine Blazy, Vincent Laporte, David Pichardie.

Static analysis of binary code is challenging for several reasons. In particular, standard static analysis techniques operate over control flow graphs, which are not available when dealing with self-modifying programs which can modify their own code at runtime. We formalized in the Coq proof assistant some key abstract interpretation techniques that automatically extract memory safety properties and control flow graphs from binary code [13], and operate over a small subset of the x86 assembly. Our analyzer is formally proved correct and has been run on several self-modifying challenges, provided by Cai et al. in their PLDI 2007 paper. This an extended version of out ITP 2014 paper.

6.3. Static analysis of functional programs using tree automata and term rewriting

Participants: Thomas Genet, Yann Salmon.

We develop a specific theory and the related tools for analyzing programs whose semantics is defined using term rewriting systems. The analysis principle is based on regular approximations of infinite sets of terms reachable by rewriting. The tools we develop use, so-called, Tree Automata Completion to compute a tree automaton recognizing a superset of all reachable terms. This over-approximation is then used to prove properties on the program by showing that some “bad” terms, encoding dangerous or problematic configurations, are not in the superset and thus not reachable. This is a specific form of, so-called, Regular Tree Model Checking. Now, we aim at applying this technique to the static analysis of programming languages whose semantics is based on terms, like functional programming languages. We already shown that static analysis of first order functional programs with a call-by-value evaluation strategy can be automated using tree automata completion [22]. This is the subject of the PhD thesis Yann Salmon has defended [11]. Now, one of the objective is to lift those results to the static analysis of higher-order functions.

6.4. Static analysis of functional specifications

Participants: Thomas Jensen, Oana Andreescu.

We have developed a static dependency analysis for a strongly typed, high-level functional specifications written in a specification formalism developed by the SME Prove & Run. In the context of interactive formal verification of complex systems, much effort is spent on proving the preservation of the system invariants. However, most operations have a localized effect on the system, which only really impacts few invariants at the same time. Identifying those invariants that are unaffected by an operation can substantially ease the proof burden for the programmer. Our dependency analysis computes a conservative approximation of the input fragments on which the operations depend. It is a flow-sensitive interprocedural analysis that handles arrays, structures and variant data types. We have validated the scalability of the analysis to complex transition systems by applying it to a functional specification of the MINIX operating system. This work was reported in [25].

6.5. Semantics

6.5.1. Energy-valued semantics

Participant: David Cachera.

We develop a *-continuous Kleene ω-algebra of real-time energy functions [36]. Together with corresponding automata, these can be used to model systems which can consume and regain energy (or other types of resources) depending on available time. Using recent results on *-continuous Kleene ω-algebras and computability of certain manipulations on real-time energy functions, it follows that reachability and Büchi acceptance in real-time energy automata can be decided in a static way which only involves manipulations of real-time energy functions. This works opens the way to static analysis techniques for energy-valued semantics of real-time systems.

DEDUCTEAM Team

7. New Results

7.1. Termination

In [15], Frédéric Blanqui showed how to extend the notion of reducibility introduced by Girard for proving the termination of β-reduction in the polymorphic λ-calculus, to prove the termination of various kinds of rewrite relations on λ-terms, including rewriting modulo some equational theory and rewriting with matching modulo $\beta\eta$, by using the notion of computability closure. This provides a powerful termination criterion for various higher-order rewriting frameworks, including Klop's Combinatory Reductions Systems with simple types and Nipkow's Higher-order Rewrite Systems.

In [16], Frédéric Blanqui, together with Jean-Pierre Jouannaud and Albert Rubio, introduced the computability path ordering (CPO), a recursive relation on terms obtained by lifting a precedence on function symbols. A first version, core CPO, is essentially obtained from the higher-order recursive path ordering (HORPO) by eliminating type checks from some recursive calls and by incorporating the treatment of bound variables as in the so-called computability closure. The well-foundedness proof shows that core CPO captures the essence of computability arguments à la Tait and Girard, therefore explaining its name. We further show that no more type check can be eliminated from its recursive calls without loosing well-foundedness, but one for which we found no counterexample yet. Two extensions of core CPO are then introduced which allow one to consider: the first, higher-order inductive types; the second, a precedence in which some function symbols are smaller than application and abstraction.

Another extension of CPO, to dependently typed terms, has been developed by Jean-Pierre Jouannaud and Jianqi Li in [50].

Jean-Pierre Jouannaud and Albert Rubio showed in [51] how to modify recursive path orders for higher-order terms which, like CPO, include $\beta\eta$-reductions, into orders that are compatible with $\beta\eta$-conversion. The result is a powerful order for proving termination of higher-order rewrite rules based on higher-order pattern matching.

Gaëtan Gilbert and Olivier Hermant have introduced a constructive way to perform proof normalization through completeness proofs [23].

Frédéric Blanqui formalized Ramsey's proof of the (infinite) Ramsey's theorem [54] (see http://color.inria.fr/).

7.2. Confluence

Jean-Pierre Jouannaud, in collaboration with Jiaxiang Liu, has started a program in order to enable confluence proofs in $\lambda\Pi$ modulo, investigating several open confluence problems for non-terminating relations. In [27], together with Mizuhito Ogawa, they introduced the new class of layered rewrite systems, and showed that their confluence can be reduced to that of their critical pairs computed by using unification over infinite rational terms when they do not increase the layer-depth of terms. This shows why an old example of non-terminating, left non-linear, critical pair free rewrite system due to Klop was non-confluent: it indeed had a critical pair in infinite rational trees. In the same paper, they also give an example of a non-confluent, layer-depth increasing system which has no critical pairs, hence showing that layer-depth plays a key role.

7.3. Automated theorem proving

In [25], Guillaume Bury, Raphaël Cauderlier and Pierre Halmagrand presented the extension of the automated theorem prover Zenon to ML-style polymorphism.

In [20], Guillaume Bury, David Delahaye, Damien Doligez, Pierre Hamalgrand and Olivier Hermant introduced an encoding of the set theory of the B method using polymorphic types and deduction modulo, used for the automated verification of proof obligations in the framework of the BWare project.

In [24], Kailiang Ji designed a strategy to translate model-checking problems into proving the satisfiability of a set of first-order formulas. The focus is to give an encoding of temporal properties expressed in CTL as first-order formulas, by translating the logical equivalence between temporal operators into rewrite rules. In this way, proof-search algorithms designed for Deduction Modulo, such as Resolution Modulo or Tableaux Modulo, can be used to verify temporal properties of finite transition systems. This strategy is implemented in iProver Modulo, and the testing results show that Resolution Modulo can be considered as a new way to quickly determine whether a temporal property is violated or not in transition system models.

7.4. $\lambda\Pi$ modulo and Dedukti

Gaëtan Gilbert, supervised by Arnaud Spiwack, wrote a prototype of a principle unification and type inference mechanism for Dedukti, based on a monadic API. This prototype separates with an abstraction barrier a unifier kernel which implements correct unification primitives from the unification algorithm and heuristics. The unification algorithm is written in a style which closely mirrors a pen-and-paper deduction rule presentation.

Éric Uzena, supervised by David Delahaye and Arnaud Spiwack, wrote a prototype of an extension of Dedukti with associative and commutative symbols and rewriting modulo associativity and commutativity of these symbols.

7.5. Encodings into Dedukti and interoperability

Ali Assaf, Guillaume Burel, Raphaël Cauderlier, David Delahaye, Gilles Dowek, Catherine Dubois, Frédéric Gilbert, Pierre Hamalgrand, Olivier Hermant, and Ronan Saillard have written a synthetic paper on the Dedukti system and on the expression of theories in this system. This paper is submitted to publication.

Ali Assaf [32] proved that Cousineau and Dowek's embedding of functional pure type systems [41] is conservative with respect to the original systems, using a new notion of reducibility called relative normalization. Together with Cousineau and Dowek's original result on the preservation of typing, this result justifies the use of the $\lambda\Pi$-calculus modulo as a logical framework.

Ali Assaf's translation of the calculus of inductive constructions to the $\lambda\Pi$-calculus modulo, which was presented at the TYPES conference in 2014, has been published in the postproceedings of TYPES 2014 [39]. This translation, which is based on the translation of pure type systems by Cousineau and Dowek [41], is implemented in the automated translation tool Coqine.

Ali Assaf and Guillaume Burel presented their translation of HOL to Dedukti at the PxTP 2015 workshop [18]. This translation, which is based on the translation of pure type systems by Cousineau and Dowek [41], is implemented in the automated translation tool Holide.

Raphaël Cauderlier and Catherine Dubois' translation of object calculus and subtyping to Dedukti, which was presented at the TYPES conference in 2014, has been published in the post-proceedings of TYPES 2014 [34].

In [26], Raphaël Cauderlier and Pierre Halmagrand presented a shallow embedding into Dedukti of proofs produced by ZenonModulo, an extension of the tableau-based first-order theorem prover Zenon to deduction modulo and typing.

In [33], Ali Assaf and Raphaël Cauderlier have combined simple developments written in Coq and HOL using Dedukti and the existing translation tools Coqine and Holide. This work is a first step towards using Dedukti as a framework for proof interoperability.

7.6. Proof theory

Guillaume Burel, Gilles Dowek and Ying Jiang have introduced a general framework to prove the decidability of reachability and provability problems. This framework uses an analogy between the objects recognized by an automaton and cut-free proofs. Various aspects of this work have been published at FroCoS [19], LPAR [21], and another paper is in preparation.

Gilles Dowek's paper on the definition of the classical connectives and quantifiers has been published [30].

Arnaud Spiwack gave a predicative shallow embedding of a weak version of system U^- in dependent type theory, for Hurkens's paradox to hold. He also showed that a variety of incarnations of Hurkens's paradox are straightforward instantiations of this encoding, greatly simplifying existing proofs.

Arnaud Spiwack developped a topos-theoretic methodology to reason equationally on circuit languages. Results that hold for combinational circuits are lifted to sequential circuits thanks to a transfer principle. This approach allows, in particular, to simplify reasoning about more complex temporal gates than the unit delay. These results aim at enriching the compiler of the Faust audio signal processing programming language, which features such complex temporal gates.

For the sake of reliability, the kernels of Interactive Theorem Provers (ITPs) are kept relatively small in general. On top of the kernel, additional symbols and inference rules are defined. Some dependency analysis of symbols of HOL Light indicates that the depth of dependency could be reduced by introducing a few more symbols to the kernel. Shuai Wang showed that extending the kernel of HOL Light is a successful attempt to reduce proof size and speed up proof-checking. More specifically, symbols and inference rules of universal quantification and implication were added to the kernel. This approach has been proved to give equivalent proof-checking results with the size of the proof files reduced to 24% on average and a speedup of 38% for proof-checking overall.

7.7. Computation models

Pablo Arrighi and Gilles Dowek have studied the expression of mecanic motions in cellular automata. Part of this work has been published in TPNC [17] and another paper is in preparation.

Arnaud Spiwack developped a variant of Turing machine where the tape is replaced by an unlabeled tree. The additional structure makes combining machines much easier, making it tractable to give explicit descriptions of rather complex machines. The cost model of these machines models that of purely functional programming languages, making it possible to compare mathematically the complexity of imperative algorithms and of purely functional algorithms.

ESTASYS Team

6. New Results

6.1. Heterogeneous Systems

Participants: Axel Legay, Jean Quilbeuf.

> This part concerns Tasks 1, 2 and 4 of the action. We characterize and formalize heterogeneous aspects of SoS and then we define efficient monitoring algorithms and representations for their requirements. We then combine the results with Statistical Model Checking (Task 5).

Systems of Systems (SoS) are very large scale systems with particular characteristics. SoS are not directly built from scratch by a single designer or a single team but are obtained as the composition of simpler systems. SoS have strong reliability and dependability requirements, as they aim to provide a service over a long running period. SoS may dynamically modify themselves by connecting to new systems, updating or disconnecting faulty ones, making it impossible to statically know the set of subsystems that are part of the SoS before runtime.

One of the main difficulty arising when developing SoS is the fact that subsystems may have been designed with a different goal in mind. In particular, some subsystems may have their own goal which differs from the global goal of the SoS. Furthermore, each subsystem may be developed in a particular computation model, making it difficult to find a common unifying semantics for the whole SoS. Finally, SoS may exhibit some emergent behaviors that are hardly predictable at design time.

One of the solutions to allow simulation of an SoS is to rely on a common interface for interconnecting the subsystems. The Functional Mockup Interface (FMI) standard is a natural candidate for such an interface. The different components of an SoS developed in different models of computation can be translated to Functional Mockup Units (FMU). Then a so-called master algorithm coordinates the FMUs composing the system. The execution of each FMU is either directly handled by the master algorithm or relies on an external tool for its execution.

Because the subsystems composing an SoS are of heterogeneous nature, it is difficult to find a common semantics model for the whole system. Furthermore, building such a transition system is not tractable due to the complexity of the system. Thus verification through traditional model checking is not possible for SoS. However, since the FMI/FMU framework enables simulation of such systems, the statistical model checking approach can be used.

The DANSE EU project aims to provide a complete tool chain from the modeling to the verification of SoS. At the higher level, the modeling is done in UPDM using the RHAPSODY tool. At the same level, the designer can express requirements over the model using some patterns written in GCSL. The UPDM model can then be translated into a FMI/FMU format that can be simulated by a dedicated tool, named DESYRE. Similarly, the GCSL requirements are transformed into BLTL formulas. Finally, the PLASMA statistical model checker has been integrated with the DESYRE tool chain in order to check the BLTL formulas based on the simulations provided by DESYRE.

6.1.1. Papers:

papier DANSE(en cours) Ensuring a correct behaviour of SoS has a significant social impact. Their complexity and inherent dynamicity pose a serious challenge to traditional design methodologies. We propose a methodology and a tool-chain supporting design and validation of SoSs. We integrate SMC with existing industrial practice, by addressing both methodological and technological issues. Our contribution is summarized as follows: (1) a methodology for continuous and scalable validation of SoS formal requirements; (2) a natural-language based formal specification language able to express complex SoS requirements; (3) adoption of widely used industry standards for simulation and heterogeneous systems integration (FMI and UPDM); (4) development of a robust SMC tool-chain integrated with system design tools used in practice. We illustrate the application of our SMC tool-chain and the obtained results on an industrial case study from the DANSE project.

6.2. Statistical Model Checking

Participants: Axel Legay, Sean Sedwards, Jean Quilbeuf, Louis-Marie Traonouez, Chan Ngo, Cyrille Jegourel.

> This section covers Tasks 4 and 5 of the action. It consists in developping Simulation based techniques and efficient statistical algorithms for SoS.

The use of test cases remains the default means of ensuring the correct behaviour of systems in industry, but this technique is limited by the need to hypothesise scenarios that cause interesting behaviour and the fact that a reasonable set of test cases is unlikely to cover all possible eventualities. Static analysis is more thorough and has been successful in debugging very large systems, but its ability to analyse complex dynamical properties is limited. In contrast, model checking is an exhaustive technique that verifies whether a system satisfies a dynamical temporal logic property under all possible scenarios. For nondeterministic and probabilistic systems, numerical model checking quantifies the probability that a system satisfies a property. It can also be used to quantify the expected cost or reward of sets of executions.

Numerical model checking gives precise, accurate and certain results by exhaustively exploring the state space of the model, however the exponential growth of the state space with system size (the ‘state explosion problem) typically limits its applicability to “toy” systems. Symbolic model checking using efficient data structures can make certain very large models tractable. It may also be possible to construct simpler but behaviourally equivalent models using various symmetry reduction techniques, such as partial order reduction, bisimulation and lumping. If a new system is being constructed, it may be possible to guarantee the overall behaviour by verifying the behaviour of its subcomponents and limiting the way they interact. Despite these techniques, however, the size, unpredictability and heterogeneity of real systems usually make numerical techniques infeasible. Moreover, even if a system has been specified not to misbehave, it is nevertheless necessary to check that it meets its specification.

Simulation-based approaches are becoming increasingly tractable due to the availability of high performance parallel hardware and algorithms. In particular, statistical model checking (SMC) combines the simplicity of testing with the formality of numerical model checking. The core idea of SMC is to create multiple independent execution traces of a system and count how many satisfy a property specified in temporal logic. The proportion of satisfying traces is an estimate of the probability that the system satisfies the property. By thus modelling the executions of a system as a Bernoulli random variable, the absolute error of the estimate can be bounded using, for example, a confidence interval or a Chernoff bound. It is also possible to use efficient sequential hypothesis testing, to decide with specified statistical confidence whether the probability of a property is above or below a given threshold. Since SMC requires multiple independent simulations, it may be efficiently divided on parallel computer architectures, such as grids, clusters, clouds and general purpose computing on graphics processors (GPGPU).

Knowing a result with less than 100% confidence is often sufficient in real applications, since the confidence bounds may be made arbitrarily tight. Moreover, a swiftly achieved approximation may prevent a lot of wasted time during model design. For many complex systems, SMC offers the only feasible means of quantifying performance. Historically relevant SMC tools include APMC, YMER and VESTA. Well-established numerical model checkers, such as PRISM and UPPAAL, are now also including SMC engines. Dedicated SMC tools under active development include COSMOS and our own tool PLASMA. Recognising that SMC may be applied to any discrete event trace obtained by stochastic simulation, we have devised PLASMA-lab, a modular library of SMC algorithms that may be used to construct domain-specific SMC tools. PLASMA-lab has become the main vehicle of our ongoing development of SMC algorithms.

Statistical model checking (SMC) addresses the state explosion problem of numerical model checking by estimating quantitative properties using simulation. To advance the state of the art of SMC we address the ongoing challenges of rare events and nondeterminsm. We also make novel use of SMC by applying it to motion planning in the context of assisted living. Rare events are often of critical importance and are challenging to SMC because they appear infrequently in simulations. Nondeterministic models are useful to model unspecified interactions, but simulation requires that nondeterminism is resolved.

We also applied SMC in the context of Systems of Systems (SoS). In the frame of the DANSE project, Plasma-Lab was used to verify SoS, and completely integrated with the DANSE tool-chain. We are currently working on verification of dynamic SoS, where systems can appear and disappear during execution. This work is done in collaboration with the ArchWare team from IRISA. We will interface Plasma-Lab with a simulator for the Pi-ADL language that enables simulation of dynamic systems.

Our group is devising cutting edge techniques for SMC. In particular, we are developing new algorithms for non-deterministic systems as well as for dynamic systems. Rare event systems are also addressed. Finally, we also devote a large amount of time to applying our technology to realistic case studies described in high-level languages such as Simulink or System C, or even a robot moving an eldery person in a commercial center.

6.2.1. Papers:

[2] (J) People with impaired physical and mental ability often find it challenging to negotiate crowded or unfamiliar environments, leading to a vicious cycle of deteriorating mobility and sociability. To address this issue we present a novel motion planning algorithm that is able to intelligently deal with crowded areas, permanent or temporary anomalies in the environment (e.g., road blocks, wet floors) as well as hard and soft constraints (e.g., “keep a toilet within reach of 10 meters during the journey”, “always avoid stairs”). Constraints can be assigned a priority tailored on the user’s needs. The planner has been validated by means of simulations and experiments with elderly people within the context of the DALi FP7 EU project.

[3] (J) Markov decision processes (MDP) are useful to model optimisation problems in concurrent systems. To verify MDPs with efficient Monte Carlo techniques requires that their nondeterminism be resolved by a scheduler. Recent work has introduced the elements of lightweight techniques to sample directly from scheduler space, but finding optimal schedulers by simple sampling may be inefficient. Here we describe “smart” sampling algorithms that can make substantial improvements in performance.

[21] (C) Rare properties remain a challenge for statistical model checking (SMC) due to the quadratic scaling of variance with rarity. We address this with a variance reduction framework based on lightweight importance splitting observers. These expose the model-property automaton to allow the construction of score functions for high performance algorithms. The confidence intervals defined for importance splitting make it appealing for SMC, but optimising its performance in the standard way makes distribution inefficient. We show how it is possible to achieve equivalently good results in less time by distributing simpler algorithms. We first explore the challenges posed by importance splitting and present an algorithm optimised for distribution. We then define a specific bounded time logic that is compiled into memory-efficient observers to monitor executions. Finally, we demonstrate our framework on a number of challenging case studies.

[23] (C) Exhaustive verification can quantify critical behaviour arising from concurrency in nondeterministic models. Rare events typically entail no additional challenge, but complex systems are generally untractable. Recent work on Markov decision processes allows the extremal probabilities of a property to be estimated using Monte Carlo techniques, offering the potential to handle much larger models. Here we present algorithms to estimate extremal rewards and consider the challenges posed by rarity. We find that rewards require a different interpretation of confidence and that reachability rewards require the introduction of an auxiliary hypothesis test. We show how importance sampling can significantly improve estimation when probabilities are low, but find it is not a panacea for rare schedulers.

[36] (J; accepted) We propose a new SMC technique based on CUSUM, an algorithm originally used in signal processing, that detects probability change at runtime on a single execution of a system. The principle is to monitor the execution at regular time intervals, and to perform Monte Carlo checks over the samples of the execution. The results of these checks are used to compute the CUSUM ratio, whose variation allows to detect a change of the probability measure of the system. We demonstrate the algorithm to detect failures in a Simulink model of a temperature controller. Computing the exact time at which failures may happen is then useful to schedule maintenance operations.

[42] (W) Many embedded and real-time systems have a inherent probabilistic behaviour (sensors data, unreliable hardware,...). In that context, it is crucial to evaluate system properties such as "the probability that a particular hardware fails". Such properties can be evaluated by using probabilistic model checking. However, this technique fails on models representing realistic embedded and real-time systems because of the state space explosion. To overcome this problem, we propose a verification framework based on *Statistical Model Checking*. Our framework is able to evaluate probabilistic and temporal properties on large systems modelled in SystemC, a standard system-level modelling language. It is fully implemented as an extension of the Plasma-lab statistical model checker. We illustrate our approach on a multi-lift system case study.

[27] (W) Stochastic Petri nets are commonly used for modeling distributed systems in order to study their performance and dependability. This report proposes a realization of stochastic Petri nets in SystemC for modeling large embedded control systems. Then statistical model checking is used to analyze the dependability of the constructed model. Our verification framework allows users to express a wide range of useful properties to be verified which is illustrated through a case study.

[25] (C: accepted) Transaction-level modeling with SystemC has been very successful in describing the behavior of embedded systems by providing high-level executable models, in which many of them have an inherent probabilistic behavior, i.e., random data, unreliable components. It is crucial to evaluate the quantitative and qualitative analysis of the probability of the system properties. Such analysis can be conducted by constructing a formal model of the system and using probabilistic model checking. However, this method is infeasible for large and complex systems due to the state space explosion. In this work, we demonstrate the successful use of *Statistical Model Checking* to carry out such analysis directly from large SystemC models and allows designers to express a wide range of useful properties. This work is going to presented at 17th IEEE High Assurance Systems Engineering Symposium in January, 2016.

6.3. Formal Models for Variability

Participants: Axel Legay, Rudolf Fahrenberg, Jin Hyun Kim.

> This part of the report is more concerned with task 2. It studies variability aspects in the broad scope. As in the first year, we have decided to use the concept of product lines as a general framework to reason on the problematic.

The behaviour of a software system is often described in terms of its features, where each *feature* is a unit of functionality that adds value to the system. *Feature-oriented software development (FOSD)* is a software-development strategy that is based on feature decomposition and modularity. Features can be separate modules that are developed in isolation, allowing for parallel, incremental, or multi-vendor development of features. Feature orientation is particularly important in *software product lines*, where a family of related products is managed and evolved in terms of its features: a product line comprises a collection of mandatory and optional features, and individual products are derived by selecting among and integrating features from this feature set. A product line can be expressed as a single model, in which feature-specific behaviour is conditional on the presence of the feature in a product.

The downside of FOSD is that, although features are conceptualized, developed, managed, and evolved as separate concerns, they are not truly separate. They can interfere with each other, for example by trying to control the same variables, by issuing events that trigger other features, or by imposing conditions that suppress other features. Most of the early work on feature interactions focused on interactions that manifest themselves as logical inconsistencies, such as conflicting actions, nondeterminism, deadlock, invariant violation, or unsatisfiability. More recently, a more general definition of feature interaction has been introduced, in terms of a feature that is developed and verified to be correct in isolation but is found to behave differently when combined with other features, and it was shown how such *behaviour interactions* could be detected as a violation of bisimulation.

Another problem is that FTS models are monolithic models of full product lines. There is no means of modelling individual features and composing them into products or product-line models, or of specifying feature increments to an existing product-line model. As such, FTSs cannot be the mathematical basis for modelling technologies that support feature decomposition, composition, or incremental evolution of a product line.

6.3.1. Papers:

[11] (C) Featured Transition Systems (FTSs) is a popular representation for software product lines: an entire product line is compactly represented as a single transition-machine model, in which feature-specific behaviour is guarded by feature expressions that are satisfied (or not) by the presence or absence of individual features. In previous work, FTS models were monolithic in the sense that the modeller had to construct the full FTS model of the product line in its entirety. To allow for modularity of FTS models, we propose here a language for extending an existing FTS model with new features. We demonstrate the language using a running example and present results about the language's expressivity, commutativity of feature extensions, feature interactions, and resolution of such interactions.

[12] (C) We suggest a method for measuring the degree to which features interact in feature-oriented software development. To this end, we extend the notion of simulation between transition systems to a similarity measure and lift it to compute a behaviour interaction score in featured transition systems. We then develop an algorithm which can compute the degree of feature interactions in a featured transition system in an efficient manner.

6.4. Privacy and Security

Participants: Axel Legay, Fabrizio Biondi, Jean Quilbeuf, Thomas Given-Wilson, Sébastien Josse.

6.4.1. Information-Theoretical Quantification of Security Properties

> This part of the work was not foreseen at the beginning of the action. It concerns security aspects, and more precisely quantifying privacy of data. This aspect is in fact central for SoS and all our algorithms developed for Tasks 4 and 5 should be adapted to solve a series of problems linked to privacy in interconnected object and dynamical environment. For now, we only studied the foundations.

Information theory provides a powerful quantitative approach to measuring security and privacy properties of systems. By measuring the *information leakage* of a system security properties can be quantified, validated, or falsified. When security concerns are non-binary, information theoretic measures can quantify exactly how much information is leaked. The knowledge of such informations is strategic in the developments of component-based systems.

The quantitative information-theoretical approach to security models the correlation between the secret information of the system and the output that the system produces. Such output can be observed by the attacker, and the attacker tries to infer the value of the secret by combining this information with its knowledge of the system.

Armed with the produced output and the source code of the system, the attacker tries to infer the value of the secret. The quantitative analysis we implement computes with arbitrary precision the number of bits of the secret that the attacker will expectedly infer. This expected number of bits is the information leakage of the system.

The quantitative approach generalizes the qualitative approach and thus provides superior analysis. In particular, a system respects non-interference if and only if its leakage is equal to zero. In practice very few systems respect non-interference, and for those who dont it is imperative to be able to distinguish between the ones leaking a very small amount of bits and the ones leaking a significant amount of bits, since only the latter are considered to pose a security vulnerability to the system.

Since black box security analyzes are immediately invalidated whenever an attacker gains information about the source code of the system, we assume that the attacker has a white box view of the system, meaning that it has access to the systems source code. This approach is also consistent with the fact that many security protocol implementations are in fact open source.

The scope of modern software projects is too large to be analyzed manually. For this reason we provide tools that can support the analyst and locate security vulnerabilities in large codebases and projects. We work with a variety of tools, including commercial software analysis tools being adapted with our techniques, and tools such as QUAIL developed here by our team.

We applied the leakage analysis provided by QUAIL to several case studies. Our case studies (voting protocol and smart grid coordination) have in common that a publicly disclosed information is computed from the secret of every participant in the model. In the voting example, the vote of a given voter is secret, but the number of votes for each candidates is public. Similarly, in the smart grid example, the consumption of one of the houses is secret, but the consumption of a whole quarter can be deduced. Qualitative analyses are either too restrictive or too permissive on these types of systems. For instance, non-interference will reject them as the public information depends on the secret. Declassification approaches will accept them, even if the number of voters or consumers is 2, in which case the secret can be deduced.

The development of better tools for quantitative security builds upon both theoretical developments in information theory, and development of the tools themselves. These often progress in parallel with each supporting the findings of the other, and increasing the demands and understanding upon each other.

6.4.1.1. Papers:

[34] (C; submitted) Systems dealing with confidential data may leak some information by their observable outputs. Quantitative information flow analysis provides a method for quantifying the amount of such information leakage. To avoid the high computational cost of exhaustive search, statistical analysis has been studied to estimate information leakage by analyzing only a small but representative subset of the system's behavior. In this paper we propose a new compositional statistical analysis method for quantitative information flow that combines multiple statistical analyses with static trace analysis. We use partial knowledge of the system's source code or specification, therefore improving both quality and cost of the analysis. The new method can optimize the use of weighted statistical analysis by performing it on components of the system and appropriately adapting their weights. We show this approach combined with the precision of trace analysis produces better estimates and narrower confidence intervals than the state of the art.

[38] (J) The quantification of information leakage provides a quantitative evaluation of the security of a system. We propose the usage of Markovian processes to model deterministic and probabilistic systems. By using a methodology generalizing the lattice of information approach we model refined attackers capable to observe the internal behavior of the system, and quantify the information leakage of such systems. We also use our method to obtain an algorithm for the computation of channel capacity from our Markovian models. Finally, we show how to use the method to analyze timed and non-timed attacks on the Onion Routing protocol.

[40] (C) Quantitative security analysis evaluates and compares how effectively a system protects its secret data. We introduce QUAIL, the first tool able to perform an arbitrary-precision quantitative

analysis of the security of a system depending on private information. QUAIL builds a Markov Chain model of the system's behavior as observed by an attacker, and computes the correlation between the system's observable output and the behavior depending on the private information, obtaining the expected amount of bits of the secret that the attacker will infer by observing the system. QUAIL is able to evaluate the safety of randomized protocols depending on secret data, allowing to verify a security protocol's effectiveness. We experiment with a few examples and show that QUAIL's security analysis is more accurate and revealing than results of other tools.

[41] (C) Quantitative security techniques have been proven effective to measure the security of systems against various types of attackers. However, such techniques are based on computing exponentially large channel matrices or Markov chains, making them impractical for large programs. We propose a different approach based on abstract trace analysis. By analyzing directly sets of execution traces of the program and computing security measures on the results, we are able to scale down the exponential cost of the problem. Also, we are able to appy statistical simulation techniques, allowing us to obtain significant results even without exploring the full space of traces. We have implemented the resulting algorithms in the QUAIL tool. We compare their effectiveness against the state of the art LeakWatch tool on two case studies: privacy of user consumption in smart grid systems and anonymity of voters in different voting schemes.

[37] (C) In an election, it is imperative that the vote of the single voters remain anonymous and undisclosed. Alas, modern anonymity approaches acknowledge that there is an unavoidable leak of anonymity just by publishing data related to the secret, like the election's result. Information theory is applied to quantify this leak and ascertain that it remains below an acceptable threshold. We apply modern quantitative anonymity analysis techniques via the state-of-the-art QUAIL tool to the voting scenario. We consider different voting typologies and establish which are more effective in protecting the voter's privacy. We further demonstrate the effectiveness of the protocols in protecting the privacy of the single voters, deriving an important desirable property of protocols depending on composite secrets.

[39] (C) In recent years, quantitative security techniques have been providing effective measures of the security of a system against an attacker. Such techniques usually assume that the system produces a finite amount of observations based on a finite amount of secret bits and terminates, and the attack is based on these observations. By modeling systems with Markov chains, we are able to measure the effectiveness of attacks on non-terminating systems. Such systems do not necessarily produce a finite amount of output and are not necessarily based on a finite amount of secret bits. We provide characterizations and algorithms to define meaningful measures of security for non-terminating systems, and to compute them when possible. We also study the bounded versions of the problems, and show examples of non-terminating programs and how their effectiveness in protecting their secret can be measured.

6.4.2. Equivocation-based Security Measures for Shared-Key Cryptosystems

Ensuring privacy and security of communication is a fundamental concern of cyber-physical systems and handled by encryption. Information-theoretic reasoning allows the modelling of security properties via unconditional security. That is, information-theoretic approaches formalise security properties that do not rely upon unproven computational hardness results, and are not vulnerable to advances in computing hardware, software or theory. For example, such unconditional security guarantees are not weakened by quantum computers, mem-computers, or new mathematical discoveries.

Traditionally the strongest measure of the security of a system is *perfect secrecy* as proposed by Shannon. However, this relies upon having a large key that is used only once. In practice a measure of the security of cryptosystems that does not meet this requirement is more useful. To this end we presented *max-equivocation*, a measure of the maximum achievable security given the keys available. Indeed max-equivocation not only formalizes the best possible security, but also generalizes perfect secrecy.

Max-equivocation holds even when inputs to the systems (i.e. keys and messages) are not uniform. This corresponds to many real world scenarios, and indeed we have shown that existing approaches are non-optimal as they do not consider these perturbations in the inputs. We provide necessary and sufficient conditions for achieving max-equivocation, formalizing exactly when it can be achieved in practice.

We further generalize to consider scenarios where message spaces are not complete, i.e. there are messages that are invalid and could never be produced. This allows reasoning over (and contrasting with) many prior approaches as well as formalizing their strengths and weaknesses under max-equivocation.

The most common attack against such cryptosystems is to consider when the attacker sees a single (encrypted) message and tries to guess the content. This can be measured by the *vulnerability* of the system, i.e. the probability that the attacker will guess correctly the message. We formalize a *relative vulnerability* for when the attacker makes this guess under incorrect assumptions about the messages. We formalize that the attacker can never improve their chances at guessing the message with incorrect assumptions.

Now we consider what information the attacker can gain by observing the cryptosystem. We show that the encryption function alone reveals information about the possible message distributions to the attacker. In the worse case scenario an encryption function may admit only a single message distribution. Thus the construction of the encryption function should consider this and (when possible) admit many solutions.

Further we consider what the attacker can learn by observing the communication of a cryptosystem. We show that the attacker can learn the probability distribution over the ciphertexts (encrypted messages), and combined with the information from the encryption function converge upon a distribution for the messages. Again if the encryption function admits one solution then the attacker learns the exact message distribution. We show that even when a single solution will not be found, the attacker still converges upon a message distribution that can only improve their attacks.

In addition to formalizing how these attacks work, and thus how to protect against them when constructing cryptosystems, we also consider not sharing the encryption function as a mechanism to avoid the attacker exploiting it. We formalize how to still communicate effectively in this scenario, and the advantages and disadvantages of this approach.

We present several algorithms to demonstrate the practicality of the techniques. The algorithms to achieve max-equivocation consider the message distribution and compute an encryption function that achieves close to max-equivocation. Since these algorithms are tailored for the message distributions, they out perform generic algorithms. We also present algorithms that are able to perform well without revealing the entire encryption function, and thus revealing less information to the attacker and hindering cryptoanalysis.

Thus we show that unconditional security is not only more resistant to technology changes, but also can be formalised for many scenarios, and is achievable in practice.

6.4.2.1. Papers:

[29] (C, submitted) Recent work has presented max-equivocation as a measure of the resistance of a cryptosystem to attacks when the attacker is aware of the encoder function and message distribution. Here we consider the vulnerability of a cryptosystem in the one-try attack scenario when the attacker has incomplete information about the encoder function and message distribution. We show that encoder functions alone yield information to the attacker, and combined with inferable information about the ciphertexts, information about the message distribution can be discovered. We show that the whole encoder function need not be fixed or shared a priori for an effective cryptosystem, and this can be exploited to increase the equivocation over an a priori shared encoder. Finally we present two algorithms that operate in these scenarios and achieve good equivocation results, ExPad that demonstrates the key concepts, and ShortPad that has less overhead than ExPad.

[13], [28] (C; J, submitted) Preserving the privacy of private communication is a fundamental concern of computing addressed by encryption. Information-theoretic reasoning models unconditional security where the strength of the results is not moderated by computational hardness or unproven results. Perfect secrecy is often considered the ideal result for a cryptosystem, where knowledge of the ciphertext reveals no information about the message or key, however often this is impossible to

achieve in practice. An alternative measure is the equivocation, intuitively the average number of message/key pairs that could have produced a given ciphertext. We show a theoretical bound on equivocation called max-equivocation and show that this generalizes perfect secrecy when achievable, and provides an alternative measure when perfect secrecy is not. We derive bounds for max-equivocation, and show that max-equivocation is achieved when the entropy of the ciphertext is minimized. We consider encryption functions under this new perspective, and show that in general the theoretical best is unachievable, and that some popular approaches such as Latin squares or Quasigroups are also not optimal. We present some algorithms for generating encryption functions that are practical and achieve 90 - 95% of the theoretical best, improving with larger message spaces.

6.4.3. Malware Classification via Deobfuscation and Behavioral Fingerprinting

A fundamental problem to guarantee the security of systems is to be able to discriminate between legitimate processes and processes with malicious behavior. Malicious software, or malware, has to be identified and prevented from executing on the system, and its actions reverted by a disinfection process. To be able to recognize and disinfect malware it is necessary to be able to extract a behavioral fingerprint or signature from a binary file, and to construct a database of such signatures for comparison. The signatures in the database have to be classified accoring to the malware's family and category, allowing the correct disinfection method to be deployed.

Automatic extraction of behavioral signatures in the form of temporal logical graphs or control flow graphs is a recent but very effective technique, and malware developers have already adapted malware compilation chains to include techniques to hinder reverse engineering and thus prevent the extraction of such signatures. These obfuscation techniques include the addition of obfuscated conditional statements leading to dead code, control flow flattening based on complex function like cryptographic hash functions, and source code virtualization on an embedded interpreter.

Consequently, deobfuscation has to be developed along with fingerprinting techniques to be able to effectively extract malware signatures. We are pushing the state of the art in both subjects, advancing generalized and targeted deobfuscation and deploying them on an innovative virtualization and malware fingerprinting tool.

Mixed Boolean Arithmetic (MBA) obfuscation is an obfuscation technique developed by Cloakware Inc. and deployed in obfuscating compilation chains for both legitimate code and malware. We have deployed state-of-the-art SMT solvers to evaluate their effectiveness against MBA-obfuscated conditionals and ascertained their limited effectiveness. So we have developed an algebraic simplification technique targeting the algebraic structure of MBA obfuscation, and proved such technique to be extremely effective, being able to deobfuscate statements in orders of magnitude less time than the time required to obfuscate them in the first place.

While the algebraic simplification technique is very effective against MBA obfuscation, it is completely tailored to MBA obfuscation. We instead explore a completely general method based on dynamic program synthesis. Synthesis algorithms, like the ones based on Reed-Muller expansion techniques, interrogate the target (in this case the obfuscated conditional) multiple times considering it as a black-box oracle, and synthesize the function expressed by the target frm the answers to such interrogation. We determined that synthesis is significantly more efficient than SMT solving in synthesizing the obfuscated function in a very compact form, and thus very promising as a generalized deobfuscation method.

While more targeted deobfuscation techniques are required to coutneract control flow flattening and virtualization, the deobfuscation of conditional statements is an important step for malware fingerprinting. We plan to use our tool to classify a large database of malware, producing an extensive database of malware signatures representing multiple versions and families of malicious code. Malware mining and evolution techniques can be deployed on such database to construct different signatures for unknown variants of similar malware, thus improving the effectiveness of the detection process.

6.4.3.1. Papers:

[30] (C, submitted) The obfuscation of conditional statements is a simple and efficient way to disturb the identification of the control flow graph of a program. Mixed Boolean arithmetics (MBA) techniques provide concrete ways to achieve this obfuscation of conditional statements. In this work, we

study the effectiveness of automated deobfuscation of MBA obfuscation, using algebraic, SMT-based and synthesis-based techniques. We experimentally ascertain the practical feasibility of MBA obfuscation. We study using SMT-based approaches with different state-of-the-art SMT solvers to counteract MBA obfuscation, and we show how the deobfuscation complexity can be greatly reduced by algebraic simplification. We also consider synthesis-based deobfuscation and find it to be more effective than SMT-based deobfuscation. We discuss complexity and limits of all methods, and conclude that MBA obfuscation is not effective enough to be considered a reliable method for control flow or white-box obfuscation.

6.5. Energy-Centric Systems

Participants: Axel Legay, Uli Fahrenberg.

> This part is concerned with Tasks 1 and 2. Mostly, we focus on quantifying properties of interconnected objects such as Cyber Physical Systems (CPS) (SoS and CPS share a lot of commonalities).

Energy and *resource management* problems are important in areas such as embedded systems or autonomous systems. They are concerned with the question whether a given system admits infinite schedules during which (1) certain tasks can be repeatedly accomplished and (2) the system never runs out of energy (or other specified resources). Formal modeling and analysis of such problems has attracted some attention in recent years.

6.5.1. *Papers:*

- [18] (C; accepted) We define and study basic properties of *-continuous Kleene ω-algebras that involve a *-continuous Kleene algebra with a *-continuous action on a semimodule and an infinite product operation that is also *-continuous. We show that *-continuous Kleene ω-algebras give rise to iteration semiring-semimodule pairs, and that for Büchi automata over *-continuous Kleene ω-algebras, one can compute the associated infinitary power series.
- [17] (C; accepted) Energy problems are important in the formal analysis of embedded or autonomous systems. Using recent results on *-continuous Kleene ω-algebras, we show here that energy problems can be solved by algebraic manipulations on the transition matrix of energy automata. To this end, we prove general results about certain classes of finitely additive functions on complete lattices which should be of a more general interest.
- [15] (C; accepted) We develop a *-continuous Kleene ω-algebra of real-time energy functions. Together with corresponding automata, these can be used to model systems which can consume and regain energy (or other types of resources) depending on available time. Using recent results on *-continuous Kleene ω-algebras and computability of certain manipulations on real-time energy functions, it follows that reachability and Büchi acceptance in real-time energy automata can be decided in a static way which only involves manipulations of real-time energy functions.

6.6. Languages for composition

Participants: Axel Legay, Thomas Given-Wilson.

> This part is concerned with Task 1, especially to describe the composition of complex systems, and to study expressivity of existing formalisms.

Contemporary cyber-physical systems are inherently constructed out of a variety of agents with communication and interaction forming a key role in the behaviour of the system as a whole. Traditional approaches to reasoning over a single computation or treating the system as a single agent prove unsatisfactory for understanding the capabilities, strengths, and weaknesses of such systems.

Since communication is a fundamental to such systems it is necessary to understand the role the communication primitives themselves play. There are many approaches to communication primitives, often chosen for their ability to easily represent desired behaviour. However, the formal properties of many implementations or chosen models have not been presented.

An alternative to formalising each possible model individually is to abstract away and reason over families of models based on their communication primitives. This allows keys results to be achieved in one model, and then generalised to the entire family, or transferred to other families based upon formal relations between these families. Thus making it possible for results to be easily applied to many models or systems without repeating significant effort.

6.6.1. Papers:

[20] (C), [32] (J; submitted) The expressiveness of communication primitives has been explored in a common framework based on the π-calculus by considering four features: synchronism (asynchronous vs synchronous), arity (monadic vs polyadic data), communication medium (shared dataspaces vs channel-based), and pattern-matching (binding to a name vs testing name equality vs intensionality). Here another dimension coordination is considered that accounts for the number of processes required for an interaction to occur. Coordination generalises binary languages such as π-calculus to joining languages that combine inputs such as the Join Calculus and general rendezvous calculus. By means of possibility/impossibility of encodings, this paper shows coordination is unrelated to the other features. That is, joining languages are more expressive than binary languages, and no combination of the other features can encode a joining language into a binary language. Further, joining is not able to encode any of the other features unless they could be encoded otherwise.

[33] (C; submitted) The expressiveness of communication primitives has been explored in a common framework by considering four features: synchronism, arity, communication medium, and pattern-matching. These all assume asymmetric communication between input and output primitives, however some calculi consider more symmetric approaches to communication such as fusion calculus and Concurrent Pattern Calculus. Symmetry can be considered either as allowing a mixture of input and output in an action or co-action, or as the unification of actions. By means of possibility/impossibility of encodings, this paper shows that: the action and co-action approach is related to or more expressive than many previously considered languages; and the unification approach is more expressive than some, but mostly unrelated to other languages.

GALLIUM Project-Team

7. New Results

7.1. Formal verification of compilers and static analyzers

7.1.1. The CompCert formally-verified compiler

Participants: Xavier Leroy, Jacques-Henri Jourdan, François Pottier, Bernhard Schommer [AbsInt GmbH].

In the context of our work on compiler verification (see section 3.3.1), since 2005 we have been developing and formally verifying a moderately-optimizing compiler for a large subset of the C programming language, generating assembly code for the PowerPC, ARM, and x86 architectures [6]. This compiler comprises a back-end, which translates the Cminor intermediate language to PowerPC assembly and is reusable for source languages other than C [5], and a front-end, which translates the CompCert C subset of C to Cminor. The compiler is mostly written within the specification language of the Coq proof assistant, from which Coq's extraction facility generates executable OCaml code. The compiler comes with a 50000-line, machine-checked Coq proof of semantic preservation establishing that the generated assembly code executes exactly as prescribed by the semantics of the source C program.

This year, we improved the CompCert C compiler in several directions:

- The generation of debugging information in DWARF format was implemented by Bernhard Schommer at AbsInt. Consequently, CompCert-compiled programs can now be debugged using standard debuggers. Xavier Leroy extended the back-end compilation passes and their proofs to propagate debugging information throughout the compilation pipeline.
- The CompCert formal semantics was made more precise in order to increase confidence. We tightened the semantics of pointer comparisons against the null pointer. We formalized the distinction between public and private (`static`) global definitions, and used it to prove the correctness of the "Unusedglob" pass that removes unreferenced private definitions.
- The calling conventions used to pass function arguments and results of `struct` and `union` types were revised in order to comply with the Application Binary Interfaces of the target platforms.
- We added partial support for extended inline assembly, an extension of the C language popularized by the GCC compiler and often used in low-level code.
- Detailed explanations of syntax errors are now produced. This usability feature builds on François Pottier's work on error reporting in LR parsers (see section 7.4.4).
- The PowerPC back-end was extended to support the PowerPC 64-bit extensions and the Freescale E5500 variant.

We released two versions of CompCert, integrating these enhancements: version 2.5 in June and version 2.6 in December. This is the public version of CompCert, available for evaluation and research purposes. In parallel, our industrial partner, AbsInt Angewandte Informatik GmbH, sells a commercial version of CompCert with long-term maintenance.

7.1.2. Formal verification of static analyzers based on abstract interpretation

Participants: Jacques-Henri Jourdan, Xavier Leroy, Sandrine Blazy [team Celtique], Vincent Laporte [team Celtique], David Pichardie [team Celtique], Sylvain Boulmé [Grenoble INP, VERIMAG], Alexis Fouilhé [Université Joseph Fourier de Grenoble, VERIMAG], Michaël Périn [Université Joseph Fourier de Grenoble, VERIMAG].

In the context of the ANR Verasco project, we are investigating the formal specification and verification in Coq of a realistic static analyzer based on abstract interpretation. This static analyzer handles a large subset of the C language (the same subset as the CompCert compiler, minus recursion and dynamic allocation); supports a combination of abstract domains, including relational domains; and should produce usable alarms. The long-term goal is to obtain a static analyzer that can be used to prove safety properties of real-world embedded C code. The overall architecture and specification of Verasco is described in a paper that was presented at POPL 2015 [19].

This year, Jacques-Henri Jourdan continued the development of this static analyzer, with two goals. First, Jacques-Henri Jourdan improved the precision and analysis time of the existing abstract domains. The existing communication system between domains was instantiated to the cooperation between the abstract domain of intervals and the abstract domain of congruences. Second, Jacques-Henri Jourdan implemented and formalized in our static analyzer the Octagon abstract domain of Miné [46]. This led to new results in the theory behind this abstract domain, allowing Jourdan to use sparse data structures for representing octagons.

7.1.3. A SPARK Front-end for CompCert

Participants: Pierre Courtieu, Zhi Zang [Kansas University].

SPARK is a language, and a platform, dedicated to developing and verifying critical software. It is a subset of the Ada language. It shares with Ada a strict typing discipline and gives strict guarantees in terms of safety. SPARK goes one step further by disallowing certain “dangerous” features, that is, those that are too difficult to statically analyze (aliasing, references, etc). Given its dedication to safety critical software, we think that the SPARK platform can benefit from a certified compiler. We are working on adding a SPARK front-end to the CompCert verified compiler.

Defining a semantics for SPARK in Coq is previous joint work with Zhi Zang from Kansas University. The current front-end is based on this semantics. The compiler has been written and tested, and the proofs of correctness are currently under way.

7.1.4. Verified JIT compilation of Coq

Participants: Maxime Dénès, Xavier Leroy.

Last year, we started the Coqonut project, whose objective is to develop and formally verify an efficient, compiled implementation of Coq’s reduction. This year, we made progress on this verification effort:

- We ported our OCaml prototype to Coq and started its verification, notably of the first phase of the compiler which involves uncurrying, using untyped step-indexed logical relations.
- We adapted (part of) the Coq x86 macro assembler by Andrew Kennedy, Nick Benton, Jonas B. Jensen and Pierre-Evariste Dagand to x86-64. This macro assembler framework is used in Coqonut’s backend to generate assembly or machine code.

7.2. Language design and type systems

7.2.1. Full reduction in the presence of inconsistent assumptions

Participants: Didier Rémy, Gabriel Scherer.

Gabriel Scherer and Didier Rémy continued their work on assumption hiding and presented it at ESOP 2015 [22]. This work aims at restoring confluence when mixing full and weak reduction and providing a continuum between consistent and inconsistent abstraction. Assumption hiding supports fine-grained control of dependencies between computations and the logical hypotheses they depend on. Although studied for a language of coercions, the solution is more general and should be applicable to any language with abstraction over propositions that are left implicit, either for the user’s convenience in a surface language or because they have been erased prior to computation in an internal language.

7.2.2. Equivalence and normalization of lambda-terms with sums

Participants: Gabriel Scherer, Guillaume Munch-Maccagnoni [Université Paris-Diderot, laboratoire PPS].

Gabriel Scherer presented at TLCA 2015 his work on understanding equivalence of sum types using the proof-theoretical technique of focusing [24]. Independently, his collaboration with Guillaume Munch-Maccagnoni resulted in a presentation of sum equivalence using an abstract machine calculus [33]. This approach allows for a more concise and cleaner definition of the equivalence relation, and a finer-grained understanding of the role of purity assumptions in the program equivalence relation.

7.2.3. Types with unique inhabitants for code inference

Participants: Gabriel Scherer, Didier Rémy.

Gabriel Scherer and Didier Rémy presented at ICFP 2015 [23] an algorithm to decide whether a type has a unique inhabitant in the simply-typed lambda-calculus with sum types. This algorithm comes along with a prototype implementation. This minimal setting is not representative of the expressiveness of realistic programming languages, but already covers a first few interesting code inference scenarios for polymorphic libraries in functional languages with prenex polymorphism: for instance, we can infer the "bind" function of the exception monad.

7.2.4. Refactoring with ornaments in ML

Participants: Thomas Williams, Didier Rémy.

Thomas Williams and Didier Rémy continued working on ornaments for program refactoring and program transformation in ML. Ornaments have been introduced as a way to describe some changes in data type definitions that preserve their recursive structure, reorganizing, adding, or dropping some pieces of data. After a new data structure has been described as an ornament of an older one, some functions operating on the bare structure can be partially or sometimes totally lifted into functions operating on the ornamented structure.

We have previously described an algorithm to perform this lifting in ML. This description was informal. This year, we improved this algorithm by decomposing it in several steps and we formalized it. Using ornament inference, we first elaborate an ML program into a generic program, which can be seen as a template for all possible liftings of the original program. The generic program is defined in a superset of ML. It can then be instantiated with specific ornaments, and simplified back into an ML program. We also studied the properties of lifting, particularly the preservation of complexity and effects, with the aim of characterizing more precisely the syntactic liftings that can be produced by our algorithm.

On the practical side, our prototype ornamentation tool has been improved with an implementation of ornament inference. The generalized program gives a description of all possible extension points that must be filled by providing patches. In practice, a few heuristics are enough to automate most of the patching work. The rest can be filled interactively by the programmer. In the case of refactoring (the representation of a data type is modified without adding any data), the transformation is fully automatic.

7.2.5. The Mezzo programming language

Participants: Thibaut Balabonski [Université Paris Sud], François Pottier, Jonathan Protzenko.

Mezzo is a programming language proposal whose untyped foundation is very much like OCaml (i.e., it is equipped with higher-order functions, algebraic data structures, mutable state, and shared-memory concurrency) and whose type system offers flexible means of describing ownership policies and controlling side effects.

A comprehensive paper, which contains both a tutorial introduction to Mezzo and a description of its formal definition and proof, was submitted to TOPLAS in 2014. This year, after a round of reviewing, it was revised and accepted for publication [11]. A reflection on the design of Mezzo was presented at SNAPL 2015 [21].

7.3. Shared-memory parallelism

7.3.1. Weak memory models

Participants: Luc Maranget, Jade Alglave [Microsoft Research, Cambridge], Patrick Cousot [New York University], Keryan Didier.

Modern multi-core and multi-processor computers do not follow the intuitive “Sequential Consistency” model that would define a concurrent execution as the interleaving of the executions of its constituent threads and that would command instantaneous writes to the shared memory. This situation is due both to in-core optimisations such as speculative and out-of-order execution of instructions, and to the presence of sophisticated (and cooperating) caching devices between processors and memory. Luc Maranget took part in an international research effort to define the semantics of the computers of the multi-core era, and more generally of shared-memory parallel devices or languages, with a clear focus on devices.

More precisely, in 2015, Luc Maranget collaborated with Jade Alglave and Patrick Cousot to extend “Cats”, a domain-specific language for defining and executing weak memory models. A precise semantics for “Cats” is the core of a submitted journal article that also includes a study and formalisation of the HSA memory model — the Heterogeneous System Architecture foundation is an industry standards body targeting heterogeneous computing devices (see http://www.hsafoundation.com/). The new extensions of the Cats language have been integrated in the released version of the **diy** tool suite (see section 6.2).

Luc Maranget also co-authored a paper that will be presented at POPL 2016 [18]. This work describes an operational semantics for the new generation ARM processors. It is joint work with many researchers, including S. Flur and other members of P. Sewell’s team (University of Cambridge) and W. Deacon (ARM Ltd).

During his M2 internship, supervised by Luc Maranget, Keryan Didier significantly improved the **diy** tool suite, in particular by writing front-ends for ARMv8 and for a subset of the C language. Keryan Didier also wrote a new (as yet unreleased) tool to translate between various input languages, in particular from machine assemblers to generic assembler and back.

7.3.2. Algorithms and data structures for parallel computing

Participants: Umut Acar, Vitalii Aksenov, Arthur Charguéraud, Mike Rainey, Filip Sieczkowski.

The ERC Deepsea project, with principal investigator Umut Acar, started in June 2013 and is hosted by the Gallium team. This project aims at developing techniques for parallel and self-adjusting computation in the context of shared-memory multiprocessors (i.e., multicore platforms). The project is continuing work that began at Max Planck Institute for Software Systems between 2010 and 2013. As part of this project, we are developing a C++ library, called PASL, for programming parallel computations at a high level of abstraction. We use this library to evaluate new algorithms and data structures. We obtained three major results this year.

Our result on the development of fast and robust parallel graph traversal algorithms based on depth-first-search has been presented at the ACM/IEEE Conference on High Performance Computing [15]. This algorithm leverages a new sequence data structure for representing the set of edges remaining to be visited. In particular, it uses a balanced split operation for partitioning the edges of a graph among the processors involved in the computation. Compared with prior work, the new algorithm is designed to be efficient not just for particular classes of graphs, but for all input graphs.

Our second result is a calculus for parallel computing on hardware shared memory computers such as modern multicores. Many languages for writing parallel programs have been developed. These languages offer several distinct abstractions for parallelism, such as fork-join, async-finish, futures, etc. While they may seem similar, these abstractions lead to different semantics, language design and implementation decisions. In this project, we consider the question of whether it would be possible to unify these approaches to parallelism. To this end, we propose a calculus, called the *DAG-calculus*, which can encode existing approaches to parallelism based on fork-join, async-finish, and futures, and possibly others. We have shown that the approach is realistic by presenting an implementation in C++ and by performing an empirical evaluation. This work has been submitted for publication.

Our third result concerns the development of parallel dynamic algorithms. This year, we started developing a parallel dynamic algorithm for tree computations. The algorithm is dynamic in the sense that it admits changes to the underlying tree in the form of insertions and deletions of edges and vertices and updates the computation by doing total work that is linear in the size of the changes, but only logarithmic in the size of the tree. The algorithm is parallel in the sense that the updates take place in parallel. Parallel algorithms have been studied extensively in the past, but few of these are dynamic. Similarly, dynamic algorithms have also been studied extensively in the past, but few of these are parallel. Our work thus explores what in retrospect seems like an obvious gap in the literature. A paper describing this work is in preparation.

7.4. The OCaml language and system

7.4.1. The OCaml system

Participants: Damien Doligez, Alain Frisch [Lexifi SAS], Jacques Garrigue [University of Nagoya], Fabrice Le Fessant, Xavier Leroy, Luc Maranget, Gabriel Scherer, Mark Shinwell [Jane Street], Leo White [Jane Street], Jeremy Yallop [OCaml Labs, Cambridge University].

This year, we released versions 4.02.2 and 4.02.3 of the OCaml system. These are minor releases that fix about 100 bugs and implement 12 minor new features, including support for nonrecursive type definitions and a higher-level interface with documentation generation tools.

Most of our activity was devoted to preparing the next major release of OCaml, version 4.03.0, which is expected in the first quarter of 2016. The novelties we worked on include:

- Inline record types as arguments to constructors of sum types, combining the clarity and extensibility brought by named record fields with the compact in-memory representation of unnamed constructor arguments.
- Improved redudancy and exhaustiveness checks for pattern-matching over generalized algebraic data types (GADTs) [41].
- Improved unboxing optimizations for numbers, including the ability to mark arguments and results of external C functions as unboxed.
- The garbage collector was made more incremental, so as to reduce the worst-case GC pause times.
- The native-code compiler was ported to two new architectures: PowerPC 64 bits (including IBM's new little-endian variant) and IBM zSystems.

On the organization side, we switched to Github as the central repository for the OCaml development sources. Github facilitates collaborative work among the growing community of contributors to the OCaml code base. In 2015, more than 100 contributors proposed small or large improvements to the OCaml compiler distribution.

7.4.2. Memory profiling OCaml applications

Participants: Fabrice Le Fessant, Çagdas Bozman [OCamlPro], Albin Coquereau [OCamlPro].

Most modern languages make use of automatic memory management to discharge the programmer from the burden of explicitly allocating and releasing chunks of memory. As a consequence, when an application exhibits an unexpected usage of memory, programmers need new tools to understand what is happening and how to solve such an issue. In OCaml, the compact representation of values, with almost no runtime type information, makes the design of such tools more complex.

In the past, we have experimented with different tools to profile the memory usage of real OCaml applications, in particular one that saves snapshots of the heap after every garbage collection. Snapshots can then be analysed to display the evolution of memory usage, with detailed information on the types of values, where they were allocated and from where they are still reachable.

This year, we experimented in three new directions, mostly driven by the size of the snapshots to be analysed:

- We studied several ways of displaying snapshots. Because of the large amount of information contained in a snapshot, it is hard for a typical user to find what he or she is looking for. We tried multiple filtering methods, based on graph algorithms, to remove the least significant information from the reports given to the user.

- We experimented with new algorithms to compress and analyse *huge* memory snapshots, i.e., snapshots that are too big to fit in the computer's memory. Indeed, standard analyses on snapshots bigger than the available memory are too long to run in practice because of random disk accesses. Thus, we tried several compression methods for snapshots and graph-reduced them to fit in memory, without losing any information, reaching a 50x speedup in complete analysis time.
- We implemented a new graph algorithm to merge sets of blocks in memory by the sets of roots they are reachable from. Such a computation was heretofore supposed to be untractable in practice, but could actually be computed in our case on huge compressed snapshots in reasonable time.

7.4.3. Advanced development tools for OCaml

Participants: Fabrice Le Fessant, Pierre Chambart [OCamlPro], Michael Laporte [OCamlPro].

In order to promote the use of OCaml in industrial contexts, we have worked on improving the tools that accompany OCaml:

- We developed the first prototype of a native debugger for OCaml, based on the LLDB debugging framework on top of LLVM. For that, we first generated a full OCaml binding for the LLDB library, by parsing the C++ headers of the libraries and automatically generating OCaml and C++ stubs. We were then able to use the OCaml binding to develop several tools, ranging from a simple tool that displays the internal GC information of a finished OCaml application, to an almost complete debugger, which displays OCaml values using runtime type information added for memory profiling.
- We also developed a new profiling framework for OCaml, called *operf*. The framework is composed of two tools: *operf-micro* can be used to run micro-benchmarks directly from inside modified OCaml compiler sources, while the *operf-macro* tool can be used to evaluate the impact of a new compiler optimization on a large set of OPAM packages.
- Finally, we came up with new ideas for *ocp-build*, a generic building tool with OCaml-specific support, to improve the expressiveness of its package description language and to easily describe cross-compilation of OCaml packages.

7.4.4. Error diagnosis in Menhir parsers

Participant: François Pottier.

LR parsers are powerful and efficient, but traditionally have done a poor job of explaining syntax errors. Although it is easy to report where an error was detected, it seems difficult to explain what has been understood so far and what is expected next. The OCaml and CompCert compilers, until now, have offered little information to the user beyond the traditional "syntax error" message.

In 2003, Jeffery proposed associating a fixed diagnostic message with every state of the LR automaton (therefore ignoring the automaton's stack). This simple approach may seem tempting. However, a typical automaton has hundreds or thousands of states. Not all of them can trigger an error, but it is difficult to tell which can, and which cannot. Furthermore, for certain states, it is difficult (or even impossible) to write an accurate diagnostic message, because some vital contextual information resides in the stack, which Jeffery's method cannot access.

In 2015, François Pottier proposed a reachability algorithm for LR automata, which he implemented in the Menhir parser generator (see section 6.3). This algorithm allows finding out which states can trigger an error and (therefore) require writing a diagnostic message. Furthermore, Pottier proposed two mechanisms for influencing where errors are detected. If used appropriately, these mechanisms make it easier (or possible) to write an accurate diagnostic message.

Pottier applied this approach to the C grammar in the front-end of the CompCert compiler, therefore allowing CompCert to produce better diagnostic messages when a C program is syntactically incorrect.

A short paper describing this work will be presented at JFLA 2016 [29]. A longer paper is in submission.

7.4.5. Improvements to Menhir

Participants: Frédéric Bour [independent consultant], Jacques-Henri Jourdan, François Pottier, Yann Régis-Gianas [team πr^2], Gabriel Scherer.

In 2015, The Menhir parser generator (see section 6.3) was extended with many new features, several of which originated in the Merlin IDE for OCaml and were ported back into Menhir.

- The parsers generated by Menhir are now incremental: they can be stopped and resumed at any point, at essentially no cost. This is exploited in Merlin, where the text is re-parsed after every keystroke.
- The state of the parser can be inspected by the user. This allows building custom libraries, outside Menhir, for error diagnosis, error recovery, etc. This is exploited in Merlin, where a valid abstract syntax tree is built (and passed to the OCaml type-checker) even if the text contains syntax errors.
- A reachability algorithm has been implemented (see section 7.4.4). It allows finding out which states can trigger an error and (therefore) require a diagnostic message to be written. It is accompanied with several tools that help maintain the database of diagnostic messages as the grammar evolves.
- Compatibility with `ocamlyacc` has been improved, in particular insofar as the computation of locations is concerned. This should help port the OCaml parser from `ocamlyacc` to Menhir, a transition that we envision making in the near future. This should help improve the quality of OCaml's syntax error messages.

7.5. Software specification and verification

7.5.1. Machine-checked proofs of programs, including time complexity

Participants: Arthur Charguéraud, Armaël Guéneau, François Pottier.

In a security-critical setting, it is important to prove that a program is correct, and to do so formally, that is, via a machine-checked proof. It is also important, one may argue, to prove that the program does not require more resources than expected (where a "resource" may be time, memory space, disk space, network bandwidth, etc.). Otherwise, even though the program is "correct" in theory, it may turn out to be unusable in practice.

Separation Logic, extended with the notion of a "time credit", a permission to perform one step of computation, allows reasoning about the correctness and the (amortized) time complexity of a program. Using this approach, which Charguéraud implemented in the CFML tool, Charguéraud and Pottier produced a machine-checked proof of the correctness and time complexity of a Union-Find data structure, implemented as an OCaml module. This demonstrates that this approach scales up to difficult complexity analyses and down to the level of actual executable code (as opposed to pseudo-code). This work was presented at ITP 2015 [17].

During his M2 internship, Armaël Guéneau extended this approach so as to allow working conveniently with the big-O notation. He extended the CFML library and verified the time complexity of a binary random access list data structure due to Okasaki. This work has not been published yet.

7.5.2. Verified property-based random testing

Participants: Zoe Paraskevopoulou [ENS Cachan, team Prosecco], Cătălin Hriţcu [team Prosecco], Maxime Dénès, Leonidas Lampropoulos [U. of Pennsylvania], Benjamin C. Pierce [U. of Pennsylvania].

Property-based random testing has been popularized in the functional programming community by tools like QuickCheck. Its integration with a proof assistant creates an interesting opportunity: reusable or tricky testing code can be formally verified using the proof assistant itself.

We introduced a novel methodology for formally verified property-based testing and implemented it as a foundational verification framework for QuickChick, a port of QuickCheck to Coq. Our framework enables one to verify that the executable testing code is testing the right Coq property. To make verification tractable, we provided a systematic way for reasoning about the set of outcomes a random data generator can produce with non-zero probability, while abstracting away from the actual probabilities.

We also applied this methodology to a complex case study on testing an information-flow control abstract machine, demonstrating that our verification methodology is modular and scalable and that it requires minimal changes to existing code.

Maxime Dénès more specifically contributed to the development of the QuickChick Coq plug-in, to the development of Coq libraries for reasoning on the set of outcomes of random generators and to the verification of QuickChick's combinator library.

This work was presented at ITP 2015 [20].

7.5.3. Tools for TLA+

Participants: Damien Doligez, Leslie Lamport [Microsoft Research], Martin Riener [team VeriDis], Stephan Merz [team VeriDis].

Damien Doligez is head of the "Tools for Proofs" team in the Microsoft-Inria Joint Centre. The aim of this project is to extend the TLA+ language with a formal language for hierarchical proofs, formalizing Lamport's ideas [43], and to build tools for writing TLA+ specifications and mechanically checking the proofs.

This year, we released version 1.4.3 of the TLA+ Proof System (TLAPS) [40], the part of the TLA+ tools that handles mechanical checking of TLA+ proofs.

This was the last year of the ADN4SE project, which develops tools for rapid development of real-time software based on the PharOS real-time kernel developed by CEA. Within this project we built, in collaboration with CEA, a formal proof of determinacy of the message-passing subsystem of PharOS. We used this experience to improve our TLA+ tools and libraries.

We have started a rewrite of TLAPS from scratch, which will make it possible to handle all aspects of the TLA+ language, including temporal formulas and their proofs.

7.5.4. Certified distributed algorithms for autonomous mobile robots

Participants: Pierre Courtieu, Xavier Urbain [ENSIIE], Sébastien Tixeuil [U. Pierre et Marie Curie], Lionel Rieg [Collège de France].

The variety and complexity of the tasks that can be performed by autonomous robots are increasing. Many applications envision groups of mobile robots that self-organise and cooperate toward the resolution of common objectives, in the absence of any central coordinating authority.

We are developing a Coq-based verification platform for distributed algorithms for autonomous robots. This year, we mechanically proved and slightly generalized a non-trivial proof of impossibility of such an algorithm under certain hypotheses [14]. We also proved several algorithms in the literature, demonstrating the viability of the platform [13].

7.5.5. Contributions to ProofGeneral, an IDE for Coq

Participant: Pierre Courtieu.

User interface is a crucial issue for theorem provers like Coq. ProofGeneral [38], an emacs-based prover interface, is widely used among Coq users. In addition to synchronizing with the evolutions of Coq itself, we contributed many improvements to ProofGeneral during the past year, among which: a better debugging mode and message printing, user assistance for naming hypotheses and indenting proof scripts, and more.

MARELLE Project-Team

6. New Results

6.1. IDE for Coq

Participants: Enrico Tassi, Alexander Faithfull [ITU Copenhagen], Jesper Bengtson [ITU Copenhagen], Carst Tankink.

User interfaces for interactive proof assistants should rely on the advanced software available in integrated development environments. we collaborated with researchers from Copenhagen to build an Eclipse-based environment for the Coq system. This exploits the quick compilation chain that was developed for Coq 8.5. This work has been published in [15].

6.2. ELPI, Fast, Embeddable, λ-Prolog Interpreter

Participants: Enrico Tassi, Cvetan Dunchev [University of Bologna], Ferruccio Guidi [University of Bologna], Claudio Sacerdoti Coen [University of Bologna].

We developed a new interpreter that runs consistently faster than the other available implementations of λ-prolog. The key insight is the identification of a fragment of the language, which we call reduction-free fragment, that occurs quite naturally and that admits constant time reduction and unification rules. In the long run, we hope that this will contribute to developing elaborators that support a more efficient and adaptable usage of interactive proof tools. This work is published in [14].

6.3. Verified Proofs of Higher-Order Masking

Participants: Gilles Barthe [IMDEA Software, Madrid], Sonia Belaïd [Thales Communication], François Dupressoir [IMDEA Software, Madrid], Pierre-Alain Fouque [Université de Rennes, IUF], Benjamin Grégoire, Pierre-Yves Strub [IMDEA Software, Madrid].

We study the problem of automatically verifying higher-order masking countermeasures. We propose a method based on program verification techniques, to check the independence of sets of intermediate variables from secrets. This new language-based technique makes it possible to implement several algorithms that reduce the number of sets of variables that need consideration. The tool also has the capability to to give useful information when proofs fail, for instance by discovering possible attacks. This is based on EasyCrypt. This work has been published in [8].

6.4. Relational Reasoning via Probabilistic Coupling

Participants: Gilles Barthe [IMDEA Software, Madrid], Thomas Espitau [ENS Cachan], Benjamin Grégoire, Justin Hsu [University of Pennsylvania], Léo Stefanesco [ENS Lyon], Pierre-Yves Strub [IMDEA Software, Madrid].

Probabilistic coupling is a powerful tool for analyzing pairs of probabilistic processes. While the mathematical definition of coupling looks rather complex and cumbersome to manipulate, we show that the relational program logic pRHL—the logic underlying the EasyCrypt cryptographic proof assistant—already internalizes a generalization of probabilistic coupling. With this insight, constructing couplings is no harder than constructing logical proofs. We demonstrate how to express and verify classic examples of couplings in pRHL, and we mechanically verify several couplings in EasyCrypt. This work is described in [9].

6.5. Automated Proofs of Pairing-Based Cryptography

Participants: Gilles Barthe [IMDEA Software, Madrid], Benjamin Grégoire, /benedikt Schmidt [IMDEA Software, Madrid].

We implement a new tool, called AutoG&P, which supports extremely compact, and often fully automated, proofs of cryptographic constructions based on (bilinear or multilinear) Diffie-Hellman assumptions. For instance, we provide a 100-line proof of Waters' Dual System Encryption (CRYPTO'09), and fully automatic proofs of Boneh-Boyen Identity-Based Encryption (CRYPTO'04). Finally, we provide an automated tool that generates independently verifiable EasyCrypt proofs from AutoG&P proofs. This work has been published in [10].

6.6. Improvements on CBC MAC formalized in EasyCrypt

Participants: Benjamin Grégoire, Cécile Baritel-Ruet, Pierre-Alain Fouque.

In a paper of 2003, J. Black and P. Rogaway propose variations of cipher block chaining message authentication codes for the efficient authentication of arbitrary length messages. We formalize their work in EasyCrypt, resulting in formal proofs for CBC-MAC, EMAC, ECBC, FCBC and the most efficient of these variations, XCBC.

This work required the development of new EasyCrypt theories. A small flaw in the original paper was found and a fix has been proposed. This work was partially funded by the Brutus ANR project.

6.7. Buchberger's algorithm and advanced formalization of multinomials

Participant: Laurent Théry.

We studied how the Mathematical Components library could improve the formalization of of algorithms based on multivariate polynomials. In particular, we re-used Pierre-Yves Strub library of multivariate polynomials and re-did the proofs of correctnes for Buchberger's algorithm. This new piece of formalized algorithm is now available at the following address https://github.com/thery/grobner.

6.8. Proofs that e and π and transcendental

Participants: Sophie Bernard, Laurence Rideau, Yves Bertot, Pierre-Yves Strub [IMDEA Software, Madrid].

In the previous year, we developed formally verified proofs that e and π are transcendental. This year we cleaned up these proofs to obtain a common lemma that applies in both cases with simple hypotheses. In parallel, P.-Y. Strub streamlined the library on multivariate polynomials which plays a significant role in the case of π. This work has been published in [11].

In the future, we will probably extend this work to more general proofs of transcendance.

6.9. Algorithms for Real Algebraic Geometry

Participant: Cyril Cohen.

We formalized an efficient algorithm to count roots of a polynomial satisfying polynomial inequalities. This work was presented at the Types workshop in May and the Workshop on Algebra, Geometry, and Proofs in Symbolic computation.

6.10. Nominal sets in Coq

Participants: Cyril Cohen, Nicolas Tabareau, Matthieu Sozeau, Gabriel Lewertoski.

Cyril Cohen collaborated with members of the team $\pi.r^2$ on the implementation of nominal sets in Coq.

6.11. Formal Description of Dynamic Logic

Participants: Yves Bertot, Cyril Cohen, Jean-Yves Franceschi.

We developed a formal description of the language of dynamic logic in the Coq system.

6.12. Cubical Type Theory

Participants: Cyril Cohen, Thierry Coquand, Simon Huber, Anders Mörtberg.

We participate to the development of a software prototype, `cubicaltt`, https://github.com/mortberg/cubicaltt, that is expected to support an extension of type theory suited for homotopy type theory.

6.13. Finite set and finite maps

Participant: Cyril Cohen.

We extend the Math-Components library with a module concerning finite sets (in potentially infinite types) and finite maps. This module will play a crucial role in other experiments, like the experiments on dynamic logic, nominal sets, and cubical sets.

6.14. Formalization of a Newton Series Representation of Polynomials

Participants: Boris Djalal, Cyril Cohen.

We formalize an algorithm to change the representation of a polynomial to a Newton power series. This provides a way to compute efficiently polynomials whose roots are the sums or products of roots of other polynomials, and hence provides a base component of efficient computation for algebraic numbers. In order to achieve this, we formalize a notion of truncated power series and develop an abstract theory of poles of fractions. This work is described in [13].

6.15. Formal description of catalan numbers

Participant: José Grimm.

Catalan number can be defined by a recurrence, or by explicit formulas using binomial numbers. An important property is $C_{n+1} = \sum_{k\leq n} C_k C_{n-k}$. The easiest way to prove this formula is to use Dyck paths.

A Dyck path of size $2n$ is a sequence l of integers $+1$ and -1 so that the sum s_k of the k first terms is ≥ 0 for $k \leq 2n$ and $s_{2n} = 0$. The relation between Dyck paths and Catalan numbers is easy to prove and then properties of Dyck paths are quite simple to state and verify.

The proofs have been done with the Math-Components library.

6.16. Latex to XML translator

Participant: José Grimm.

This year, we released version 2.15.4 of Tralics, our LaTeX to XML translator. Array handling has been redesigned: for instance, an array preamble of the form `{>{$}c<{$}}` is now correctly interpreted; there is a possibility to add an attribute pair to any table, row or cell; for math environments like “align”, one label and one tag per row is allowed. Tralics is also able to read an XML file, and there are some primitives for inserting the result (or part of it) into the XML code under construction.

MEXICO Project-Team

7. New Results

7.1. Highlights

Please note that three of our most important and novel results are given in the 'Highlights' section above.

7.2. Specifying and Verifying Concurrent C Programs with TLA+

Verifying software systems automatically from their source code rather than modelling them in a dedicated language gives more confidence in establishing their properties. In [37] we propose a formal specification and verification approach for concurrent C programs directly based on the semantics of C. We define a set of translation rules and implement it in a tool (C2TLA+) that automatically translates C code into a TLA+ specification. The TLC model checker can use this specification to generate a model, allowing to check the absence of runtime errors and dead code in the C program in a given configuration. In addition, we show how translated specifications interact with manually written ones to: check the C code against safety or liveness properties; provide concurrency primitives or model hardware that cannot be expressed in C; and use abstract versions of translated C functions to address the state explosion problem. All these verifications have been conducted on an industrial case study, which is a part of the microkernel of the PharOS real-time system.

7.3. Active Diagnosis with Observable Quiescence

Active diagnosis of a discrete-event system consists in controlling the system such that faults can be detected. In [27] we extend the framework of active diagnosis presented in [7] by introducing modalities for actions and states and a new capability for the controller, namely observing that the system is quiescent. We design a game-based construction for both the decision and the synthesis problems that is computationally optimal. Furthermore we prove that the size and the delay provided by the active diagnoser (when it exists) are almost optimal.

7.4. Test Case Generation for Concurrent Systems Using Event Structures

In [23] we deal with the test-case generation problem for concurrent systems that are specified by true-concurrency models such as Petri nets. We show that using true-concurrency models reduces both the size and the number of test cases needed for achieving certain coverage criteria. We present a test-case generation algorithm based on Petri net unfoldings and a SAT encoding for solving controllability problems in test cases. Finally, we evaluate our algorithm against traditional test-case generation methods under interleaving semantics.

7.5. State Space Reduction Strategie for Model Checking Concurrent C Programs

Model checking is an effective technique for uncovering subtle errors in concurrent systems. Unfortunately, the state space explosion is the main bottleneck in model checking tools. In [31] we propose a state space reduction technique for model checking concurrent programs written in C. The reduction technique consists in an analysis phase, which defines an approximate agglomeration predicate. This latter states whether a statement can be agglomerated or not. We implement this predicate using a syntactic analysis, as well as a semantic analysis based on abstract interpretation. We show the usefulness of using agglomeration technique to reduce the state space, as well as to generate an abstract TLA+ specification from a C program.

7.6. Simple Priced Timed Games Are Not That Simple

Priced timed games are two-player zero-sum games played on priced timed automata (whose locations and transitions are labeled by weights modeling the costs of spending time in a state and executing an action, respectively). The goals of the players are to minimise and maximise the cost to reach a target location, respectively. In [25] we consider priced timed games with one clock and arbitrary (positive and negative) weights and show that, for an important subclass of theirs (the so-called simple priced timed games), one can compute, in exponential time, the optimal values that the players can achieve, with their associated optimal strategies. As side results, we also show that one-clock priced timed games are determined and that we can use our result on simple priced timed games to solve the more general class of so-called reset-acyclic priced timed games (with arbitrary weights and one-clock).

7.7. A Hybrid-Dynamical Model for Passenger-flow in Transportation Systems

In a network with different transportation modes, or multimodal public transportation system (MPTS), modes are linked among one another not by resources or infrastructure elements—which are not shared, e.g., between different metro lines—but by the flow of passengers between them. Now, the movements of passengers are steered by the destinations that individual passengers have, and by which they can be grouped into trip profiles. To use the strength of fluid dynamics, introduce in [30] a multiphase hybrid Petri net model, in which the vehicle dynamics is rendered by individual tokens moving in an infrastructure net, while passenger quantities are given as vectors—whose components correspond to trip profiles—and evolve at stations according to fluid dynamics. This model is intended as a building block for obtaining supervisory control, via transport operator actions, to mitigate congestion.

7.8. An Algebraic View of Space/Belief and Extrusion/Utterance for Concurrency/Epistemic Logic

In [29] we enrich spatial constraint systems with operators to specify information and processes moving from a space to another. We shall refer to these news structures as spatial constraint systems with extrusion. We shall investigate the properties of this new family of constraint systems and illustrate their applications. From a computational point of view the new operators provide for process/information extrusion, a central concept in formalisms for mobile communication. From an epistemic point of view extrusion corresponds to a notion we shall call utterance; a piece of information that an agent communicates to others but that may be inconsistent with the agent's beliefs. Utterances can then be used to express instances of epistemic notions, which are common place in social media, such as hoaxes or intentional lies. Spatial constraint systems with extrusion can be seen as complete Heyting algebras equipped with maps to account for spatial and epistemic specifications.

7.9. Preserving Partial Order Runs in Parametric Time Petri Nets

Parameter synthesis for timed systems aims at deriving parameter valuations satisfying a given property. In [22] we target concurrent systems; it is well known that concurrency is a source of state-space explosion, and partial order techniques were defined to cope with this problem. Here we use partial order semantics for parametric time Petri nets as a way to significantly enhance the result of an existing synthesis algorithm. Given a reference parameter valuation, our approach synthesizes other valuations preserving, up to interleaving, the behavior of the reference parameter valuation. We show the applicability of our approach using acyclic asynchronous circuits.

7.10. Non-Atomic Transition Firing in Contextual Nets

The firing rule for Petri nets assumes instantaneous and simultaneous consumption and creation of tokens. In the context of ordinary Petri nets, this poses no particular problem because of the system's asynchronicity, even if token creation occurs later than token consumption in the firing. With read arcs, the situation changes, and several different choices of semantics are possible. The step semantics introduced by Janicki and Koutny can be seen as imposing a two-phase firing scheme: first, the presence of the required tokens is checked, then consumption and production of tokens happens. Pursuing this approach further, we develop in [28] a more general framework based on explicitly splitting the phases of firing, allowing to synthesize coherent steps. This turns out to define a more general non-atomic semantics, which has important potential for safety as it allows to detect errors that were missed by the previous semantics. Then we study the characterization of partial-order processes feasible under one or the other semantics.

PARSIFAL Project-Team

7. New Results

7.1. The Checkers Proof Certifier

Participants: Tomer Libal, Giselle Reis, Hichem Chihani.

We presented a system description [29] of the Checkers proof certifier, which implements some of the theoretical ideas developed in the ProofCert project. This version of the system is capable of certifying a subset of the E-Prover superposition theorem prover. The system is mainly written in λProlog with a proof importing module written in Ocaml. The system is designed to allow modularity when designing the semantical translations of proof systems. For this capacity, the system supports, J. A. Robinson's resolution and the paramodulation technique of G. Robinson and L. Wos. On top of that, minimal support for some inference rules of the E-Prover was added.

7.2. Regular Patterns in Second-Order Unification

Participant: Tomer Libal.

We presented a paper [33] detailing a higher-order pre-unification procedure with improved termination over existing procedures. The classic higher-order unification procedure was presented by G. Huet in 1975 and is still used as the main unification procedure for higher-order automated theorem provers. This procedure does not terminate. In this project we have investigated the reasons for that and have shown that by choosing a specific (but complete) search strategy, an additional set of non-unifiable problems can be detected. As an example, we have shown that all unification problems generated by the Leo-III theorem prover when proving Cantor's theorem are decided by this procedure, in contrast to the classical unification procedure.

7.3. Static guarantees for message-passing computation

Participant: Stéphane Graham-Lengrand.

LCF [79] is a proof-search architecture, where search strategies are programmed via an API and successful proof-search runs are guaranteed correct, relying on the use of an abstract type `theorem`. We adapted the approach and defined principles for message-passing software architectures (where modules interact by exchanging messages), with the objective of guaranteeing message provenance and integrity. The principles rely on abstract types to *sign* messages at no run-time cost, and more generally rely on type-checking to provide static guarantees (i.e. at compile-time) that the messages produced by a trusted piece of code will not be altered or faked by an untrusted piece of code. We developed this primarily for safe theorem proving architectures, but the approach can be applied to other software architectures where modules with different levels of trust interact.

7.4. Proof-search with quantifiers and theories

Participant: Stéphane Graham-Lengrand.

We published our approach to proof-search on quantified problems in presence of one theory [22], where we identify the specifications required of the theory for the proof-search process to be sound and complete. Theories with unification procedures or quantifier elimination procedures satisfy our specifications, where *constraint streams* and *constraint projections* play a key role key. Interestingly enough, Bjorner and Janota [52] independently achieved a similar result with *model projections*. Our theory-generic approach allows a clear formulation of what it could mean to combine several quantifier-handling theories, hopefully generalising what the Nelson-Oppen combination technique does in a quantifier-free context. We recently obtained two new results towards this:

- First, the cumbersome, stream-querying, and backtracking mechanisms that were required to implement [22] have been re-expressed in a more satisfying message-passing computational framework.
- Second, we re-expressed the standard quantifier-free combination techniques, mentioned above, as a concurrent message-passing interaction between different theory-specific procedures, and simplified their proofs of correctness. This led to the major redesign of Psyche, mentioned above.

7.5. Realizability semantics of abstract focusing, formalized

Participant: Stéphane Graham-Lengrand.

In [21] we presented a parametric system for abstract focusing, building on Zeilberger's work [87], and parametrically capturing classical and intuitionistic focused systems. We presented its semantics, building on Munch-Maccagnoni's work [80], in terms of *abstract realizability models* (which were independently identified by Krivine). The goal was to emphasize the similarities and differences between focusing and realizability, in the way they exploit the *polarities* of formulae. The system and its semantics led to a substantially formalisation in the proof assistant Coq.

7.6. The Meta-Theory of Bisimulation-Up-To

Participants: Kaustuv Chaudhuri, Matteo Cimini, Dale Miller.

The method of proof by bisimulation has proved to be a very successful technique for showing the equivalence of processes. Unfortunately, in process calculi with infinite transition systems, such as in calculi with a replication operator, finding a bisimulation requires exploring an infinite search space, which moreover often tends to have rather intricate and complex structure. One way to combat this complexity—i.e., reduce the size of candidate bisimulation sets—is to identify redundancies among their members and then to replace redundant classes by unique inhabitants. This yields families of *bisimulation-up-to* proof methods that are parametric over the redundancy relation. For instance, if we consider bisimilarity itself as the redundancy, then we obtain *bisimulation up to bisimilarity*; with this relation, the singleton set $\{(!a, !!a)\}$ is a candidate set for showing that the processes $!a$ and $!!a$ are bisimilar, for example, when the bisimulation set with redundancies is infinite.

Since *a priori* there is no restriction on such redundancy relations, a key theoretical question is when a bisimulation-up-to relation is *sound*, i.e., that it is contained in a bisimulation. In the literature there have been a number of techniques proposed for showing soundness, but they often require the use of complex reasoning about lattices of fixed points. In [19] (CPP'15) we show how to use the built-in coinduction facilities of the Abella theorem prover to produce comparatively lightweight proofs of the soundness of many common bisimulation-up-to techniques for CCS and the π-calculus. A key feature of our approach is that we can use the facilities already provided by the Abella system for reasoning about the binding constructs for the π-calculus.

7.7. Characterizing Independence in Type Theory

Participants: Kaustuv Chaudhuri, Yuting Wang.

In formal proof languages based on type theory, it is often the case that a theorem is proved for a certain kind of typing context, but needs to be used in a different context. For example, theorems about natural numbers may be proved in an empty typing context, since the type of natural numbers contains no higher-order features (i.e., natural numbers are closed), but we may need to use these properties of natural numbers when reasoning about λ-terms in De Bruijn notation, where the typing context is non-empty. In such a situation, it is useful to automatically transport the existing theorems to the new kinds of contexts, since we know that the theorem in question cannot depend on the properties of λ-terms. While this example is rather trivial, it becomes non-trivial when theorems are proved about higher-order data structures, which are commonly encountered when reasoning about syntax with binding constructs.

One way to achieve such reuse automatically is a technique called *subordination*, which is based on analyzing the constructors for a certain type and defining syntactic criteria under which certain normal terms of one type can have subterms of another type. Unfortunately, the classical definition of subordination lacks a proof-theoretic justification, and has surprising properties in third-order (and higher) signatures.

In [36] (TLCA'15), we propose a proof-theoretic characterization of a kind of dual to subordination, called *independence*, that characterizes when normal terms of one type *cannot* contain subterms of another type. This is achieved by means of proving an inductive *strengthening* lemma about the signatures in the two-level logic approach. We also show how to automatically prove such lemmas in certain commonly encountered situations in the theorem prover Abella.

7.8. Disproving Non-Theorems with Saturating Search

Participants: Taus Brock-Nannestad, Kaustuv Chaudhuri.

High-performance automated reasoning techniques such as resolution and the inverse method are well suited for proving *true* conjectures, but are ill-behaved for *false* conjectures. For example, for a simple theory of even numbers that states that 0 is even and that $n + 2$ is even whenever n is even, it is obviously the case that the conjecture “3 is even” is unprovable, but the algorithm would loop forever proving “0 is even”, “2 is even”, “4 is even”, etc. This behavior is observed even in the best saturation-based (i.e., forward-reasoning) theorem provers.

In [25] (TABLEAUX'15), we show how to finitely constrain the search space of saturation-based theorem provers by the use of *unsound* extensions of the goal query. These unsound extensions, when combined with forward subsumption, guarantee that only a finite number of consequences would ever be constructed based on any goal query, so the proof search procedure is guaranteed to terminate. If a proof is found among them that does not use the unsound extensions, then we can can succeed with that proof. If no proof is found, then we can soundly assert that the original goal query was also unprovable, since even a weakened version of it was unprovable. The only other possibility is that a proof is found using the unsound extension; in this case, we use the particular instance of unsoundness to refine the original unsound goal to prevent it from being found again, while maintaining the invariant that the search space is finite, and rerun the search. Since first-order logic is undecidable, we may need to repeat the refinement procedure indefinitely, but for many kinds of domains, particularly those arising from typed signatures (such as the even numbers example above), we do eventually find a saturating approximation that guarantees that the conjecture has no proof.

This algorithm has been implemented as part of the Mætning theorem prover explained in the section on Software above. We plan to extend it in the future with various automatic refinement heuristics.

7.9. Encoding Bigraph Structure with Subexponentials

Participants: Kaustuv Chaudhuri, Giselle Reis.

Bigraphs were proposed by Robin Milner as a model of ubiquitous computing, which is computation that is aware both of *location* and of *connections*. As a formalism it subsumes many other process calculi such as CCS and the π-calculus. However, it has a number of problems *qua* syntax because it is based on graphs and a complicated theory of composition. The biggest of these problems is how to implement it in a formal reasoning system.

In recent years, many members (and ex-members) of the Parsifal team have been experimenting with a variant of linear logic that has not just a single pair but an arbitrary family of exponential connectives that are arranged in a pre-order. Each such pair of *subexponentials* may admit or reject the structural properties of weakening and contraction. One benefit of subexponentials is that it allows for *querying* the *absence* of certain kinds of exponential formulas without requiring all non-exponential formulas to be deleted as a consequence, which is the issue with ordinary linear logic.

In [28] (LPAR'15), we show how to represent the structure of bigraphs in terms of a simple theory of linear logic with subexponentials (SEL). We show that our representation is adequate, i.e., that it respects the composition and juxtaposition operations on bigraphs. Moreover, we show how one can ask queries about the nesting of places in the representation without modifying it, which gives us a technical means of encoding bigraph reactions as well. Some of the details for bigraph reactions remain to be worked out in future work.

7.10. Encoding Additive Connectives with Multiplicatives and Subexponentials

Participant: Kaustuv Chaudhuri.

In a recent workshop on Linearity [55], we have published the formal proof (that was obtained in 2009) that linear logic with three subexponentials in a certain lattice is undecidable. An extended version of this paper was submitted to a special issue on Linearity in Mathematical Structures in Computer Science and was accepted in November 2015.

The preprint of that extended paper [41] gives a direct embedding of propositional MALL (multiplicative and additive linear logic) using only multiplicative connectives and five subexponentials. This means that the additive connectives are, in fact, redundant when we have multiplicatives and subexponentials. Moreover, in the first-order case this encoding is polynomial and focally adequate, which means that MALL can be simulated at the highest fidelity – at the level of individual inference rules.

7.11. Computation in Focused Intuitionistic Logic

Participants: Taus Brock-Nannestad, Nicolas Guenot, Daniel Gustafsson.

Focusing is a proof-theoretical tecnique for eliminating unnecessary nondeterminism in proofs. Because it cuts down on nondeterminism, focusing is particularly useful for directing proof search. Focusing thus plays a key role in explaining the meaning and behaviour of logic programs.

Despite this success in clarifying the operational semantics of logic programming, focusing has not been as widely studied in the Curry-Howard style "proofs as programs" interpretation. Early results in this area established that λ-calculi associated with the focused calculi LJT and LJQ had evaluation strategies corresponding to call-by-name and call-by-value respectively. For the LJF calculus — which contains both LJT and LJQ as fragments — no such correspondence was known.

In [27] (PPDP'15) we show how a proof-term assignment to (a variant of) Liang and Miller's focused sequent calculus LJF permits a uniform treatment of the call-by-value and call-by-name reduction strategies of the λ-calculus, as well as combinations of these strategies. Additionally, we show how to extract an abstract machine from LJF by considering machine states as certain configurations of instances of the cut rule. The aforementioned correspondence extends to this setting, and we show that well-known abstract machines for call-by-value and call-by-name are in fact exactly the abstract machines that one gets when considering certain fragments of LJF.

In the seminal work of Paul Blain Levy, the call-by-push-value language was introduced as a way of subsuming the call-by-value and call-by-name strategies of the λ-calculus. It was later on conjectured that call-by-push-value was simply implementing a notion of focusing, and indeed this turns out to be the case, as we show in the aforementioned paper.

7.12. Focused Linear Logic and the λ-calculus

Participants: Taus Brock-Nannestad, Nicolas Guenot.

Linear Logic enjoys strong symmetries inherited from classical logic while providing a constructive framework comparable to intuitionistic logic. However, the computational interpretation of sequent calculus presentations of linear logic remains problematic, mostly because of the many rule permutations allowed in the sequent calculus.

In focused variants of Linear Logic, most of these rule permutations are eliminated by the focusing restriction — during focusing, a single formula is decomposed eagerly, and the focus is passed down to its subformulas. Conversely, during inversion, all invertible connectives are decomposed. Moreover, this decomposition is made fully determinstic by keeping the connectives in question in a list, and only decomposing the first connective of this list.

The end result of this is that a focused proof in Linear Logic almost always has one particular formula singled out as the one that will be decomposed. Thus, somewhat curiously, focused Linear Logic behaves much more like an intuitionistic sequent calculus (where at all times there is a single "special" formula on the right hand side of the sequent) than a classical calculus.

In [26] (MFPS'15), we study a term assignment for a focused version of Multiplicative Exponential Linear Logic (MELL), and show how the focusing technique gives rise to a calculus that straightforwardly embeds both a linear variant of the λ-calculus, and a sequent-based formulation of Parigot's $\lambda\mu$-calculus.

7.13. There is no complete linear term rewriting system for propositional logic

Participant: Lutz Straßburger.

Recently, we observed that the set of all sound linear inference rules in propositional logic is already coNP-complete [84]. This means that every boolean tautology can be written as a (left-and right-) linear rewrite rule. This raises the question of whether there is a rewriting system on linear terms of propositional logic that is sound and complete for the set of all such rewrite rules. We have shown (in a joint work with Anupam Das) that, as long as reduction steps are polynomial-time decidable, such a rewriting system does not exist unless coNP=NP. This is published in [20].

7.14. A (Bi)linear Implementation of Strong Call-by-Value

Participant: Beniamino Accattoli.

The elegant theory of the call-by-value λ-calculus relies on closed terms and weak evaluation (i.e., not under abstractions) and it is well-known that the number of call-by-value β-steps is a reasonable cost model. When turning to open terms or strong evaluation—that are used for instance in the implementation of Coq—the operational theory breaks, and the call-by-value λ-calculus has to be extended with some additional rewriting rules. In a joint work with Sacerdoti Coen [18], a proposal for open/strong call-by-value, called *fireball calculus*, is studied from the point of view of cost models and abstract machines. First, it is shown that open terms introduce a new malicious behavior, making the study of cost models non-trivial. Second, it is shown that the number of β-steps in the fireball calculus is a reasonable cost model. Third, a new abstract machine is introduced and its overhead is shown to be linear with respect to the number of β-steps and the size of the initial term, providing a surprisingly efficient implementation scheme.

7.15. Implementations of Strong Call-by-Name, Revisited

Participant: Beniamino Accattoli.

The literature about abstract machines for the strong evaluation (i.e., possibly under abstraction) of the ordinary (i.e., call-by-name) λ-calculus is scarce. Essentially, there is a single, old work: Crégut's abstract machine [60] (1990), that is an extension of Krivine abstract machine to compute full normal forms. Crégut studies the correctness of the machine by means of an explicit substitutions calculus. In this joint work with Barenbaum and Mazza [17], Crégut's work is revisited and simplified in the extreme. An alternative, simpler machine is introduced, the *Strong Milner abastract machine*. Its correctness is studied via *linear substitution calculus*, a new approach to explicit substitutions developed by Accattoli and Kesner that is much simpler than Crégut's approach. Moreover, a complexity analysis of the machine is provided: its overhead is shown to be linear in the number of steps in the linear substitution calculus and in the size of the inital term.

7.16. Foundational Proof Certificates

We have continued to explore a number of new aspects of framework we call *Foundational Proof Certificates* (FPCs). Besides having defined and implemented prototype checkers for FPCs in classical and intuitionistic logic [37] we have also extended the proof theory underlying numerous modal logics so that FPCs can be applied to modal logics [35]. We have also extended the notion of FPC to work also in the model checking setting [31]. In both the modal logic and model checking domains, the key to getting FPCs to work is to have descriptions of *focused proof systems* available for those logics.

Given that FPCs are declarative and semantically simple structures, it has been possible to find numerous applications of them outside the problem of simply checking them. It was shown, for example, that FPCs can be used to help define the semantics of the output from traditional theorem provers [23]. We have also used FPCs as proof outlines in order to define high-level tactics to direct proof search [24].

7.17. Multi-level Delimited Control

There has been a great deal of interest in recent years to providing interesting functional programming primitives that are based on classical logic and not just intuitionistic logic. Unfortunately, the standard sequent calculus proof theory for classical logic is far too chaotic to provide such a foundation. We have recently proposed adding to classical (linear) logic an assortment of *subexponentials* and to provide a rigid structure for their placement within formulas. This new framework allows for sequent calculus proof theory to provide to the functional programming paradigm the feature often called *multi-level delimited control* [32]. The main result in that paper is also noteworthy in that it shows how to build certain complex synthetic connectives even though the standard approach (using focusing proof systems) cannot be used.

PI.R2 Project-Team

6. New Results

6.1. Effects in proof theory and programming

Participants: Guillaume Claret, Pierre-Louis Curien, Hugo Herbelin, Étienne Miquey, Ludovic Patey, Pierre-Marie Pédrot, Yann Régis-Gianas, Alexis Saurin.

6.1.1. Axiom of dependent choice in classical arithmetic

In 2012, Hugo Herbelin showed that classical arithmetic in finite types extended with strong elimination of existential quantification proves the axiom of dependent choice. To get classical logic and choice together without being inconsistent is made possible first by constraining strong elimination of existential quantification to proofs that are essentially intuitionistic and secondly by turning countable universal quantification into an infinite conjunction of classical proofs evaluated along a call-by-need evaluation strategy so as to extract from them intuitionistic contents that complies to the intuitionistic constraint put on strong elimination of existential quantification. Étienne Miquey has been working on a sequent-calculus version of this system, using Danvy's methodology of semantic artifacts, to progressively reduce the consistency of such a system to the normalisation of Girard-Reynold's system F. To achieve this goal, he incidentally proposed a way to get a dependently-typed sequent calculus, as well as a method to type a state-and-continuation-passing style translation of call-by-need calculus.

6.1.2. The computational contents of completeness proofs

Hugo Herbelin worked on the computational content of Gödel's completeness theorem, developing a proof with side-effects suitable for normalisation-by-evaluation.

6.1.3. Gödel's functional interpretation

Pierre-Marie Pédrot extended the proof-as-program interpretation of Gödel's Dialectica translation to the fully dependent setting, including dependent elimination [17].

6.1.4. Logical foundations of call-by-need evaluation

Alexis Saurin and Pierre-Marie Pédrot extended their reconstruction of call-by-need based on linear head reduction with control. They showed how linear head reduction could be adapted to the $\lambda\mu$-calculus. This classical linear head reduction lifts the usual properties of the intuitionistic one (with respect to σ-equivalence) to the $\lambda\mu$-calculus (and its σ-equivalence already formulated by Olivier Laurent in his PhD thesis). Moreover, they showed that substitution sequences of the $\lambda\mu$-calculus linear head reduction are in correspondence with the classical Krivine abstract machine substitution sequences, validating the known fact that the KAM implements linear head reduction. In a second step, they could lift to the $\lambda\mu$-calculus their three-step transformation from linear head reduction to call-by-need, and study the correspondence with Ariola, Herbelin and Saurin's classical call-by-need. This work appeared as one of the chapters of Pierre-Marie Pédrot's thesis and has been accepted for publication at ESOP'16 [30].

6.1.5. Call-by-name forcing

Pierre-Marie Pédrot studied variants of the forcing construction by decomposing it through call-by-push-value. In particular, the by-name decomposition behaves much more nicely w.r.t. the computational content of proofs and is a candidate for a dependently-typed extension. This work is partially reported on in his PhD [17].

6.1.6. A theory of effects and resources

In joint work with Marcelo Fiore and Guillaume Munch-Maccagnoni, Pierre-Louis Curien considered the Curry-Howard-Lambek correspondence for effectful computation and resource management, specifically proposing polarised calculi together with presheaf-enriched adjunction models as the starting point for a comprehensive semantic theory relating logical systems, typed calculi, and categorical models in this context. Our thesis is that the combination of effects and resources should be considered orthogonally. Model theoretically, this leads to an understanding of our categorical models from two complementary perspectives: (i) as a linearisation of CBPV (Call-by-Push-Value) adjunction models, and (ii) as an extension of linear/non-linear adjunction models with an adjoint resolution of computational effects. When the linear structure is cartesian and the resource structure is trivial, we recover Levy's notion of CBPV adjunction model, while when the effect structure is trivial, we have Benton's linear/non-linear adjunction models. Further instances of our model theory include the dialogue categories with a resource modality of Melliès and Tabareau, and the Enriched Effect Calculus models of Egger, Møgelberg and Simpson. Our development substantiates the approach by providing a lifting theorem of linear models into cartesian ones. To each of our categorical models we systematically associate a typed term calculus, each of which corresponds to a variant of the sequent calculi LJ (Intuitionistic Logic) or ILL (Intuitionistic Linear Logic). The adjoint resolution of effects corresponds to polarisation whereby, syntactically, types locally determine a strict or lazy evaluation order and, semantically, the associativity of cuts is relaxed. In particular, our results show that polarisation provides a computational interpretation of CBPV in direct style. Further, we characterise depolarised models: those where the cut is associative, and where the evaluation order is unimportant. This work will be presented at POPL 2016 [26].

6.1.7. Coq as a programming language with effects

As part of his PhD thesis, Guillaume Claret defined a notion of effectful interactive computation as an embedded DSL in Coq (in the spirit of the works on algebraic effects), and used it to implement a web server. It is equipped with a dual notion of effectful interactive execution context. Using these two notions together, Guillaume Claret is able to specify and reason about interactive programs inside Coq. He submitted several papers about this line of work: one has been published [32], others will be part of his PhD manuscript.

6.2. Reasoning and programming with infinite data

Participants: Amina Doumane, Alexis Saurin, Pierre-Marie Pédrot, Yann Régis-Gianas.

This theme is part of the ANR project Rapido (see the National Initiatives section).

6.2.1. Interactive semantics for logic fixed-points and infinitary logics.

Amina Doumane and Alexis Saurin, in a joint work with David Baelde published at CSL 2015 [24], developed a game-semantics of $\mu MALL$ (Multiplicative Additive Linear Logic with least and greatest fixpoints).

This interactive semantics was worked out in computational ludics, benefitting from both the work by Clairambault on a HO style game semantics for an intuitionistic logic with least and greatest fixpoints and from the flexibility of Terui's computational ludics (in particular its ability to consider designs with cuts).

This framework is built around the notion of design, which can be seen as an analogue of the strategies of game semantics. The infinitary nature of designs makes them particularly well suited for representing computations over infinite data. We provided $\mu MALL$ with a denotational semantics (that is invariant by cut-elimination), interpreting proofs by designs and formulas by particular sets of designs called behaviours. Then a completeness result for a specific class of designs is proved, the class of "essentially finite designs", which are those designs performing a finite computation followed by a copycat. On the way to the previous completeness result, we investigate semantic inclusion, proving its decidability (given two formulas A and B, one can decide whether the semantics of A is included in the semantics of B) and completeness (if semantic inclusion holds, the corresponding implication is provable in $\mu MALL$).

6.2.2. Proof theory of circular proofs

In a collaboration with David Baelde, Amina Doumane and Alexis Saurin developed further the theory of infinite proofs. Studying the proof theory of circular proofs on MALL, they established a result of focalisation for these infinite proofs. The usual result of focalisation for linear logic can actually be extended to circular proofs but, contrarily to $\mu MALL$ where fixed-points operators can be given an arbitrary polarity, the least fixed-point must be set to be a positive construction and the greatest fixed-points to be negative, which is consistent with intuition from programming with inductive and co-inductive datatypes. An interesting phenomenon arising with focalisation is that some infinite but regular proofs may not have any regular focused proofs. This is similar to what happens for cut-elimination of regular proofs.

Works on cut-elimination for circular proofs are still ongoing.

6.2.2.1. Automata theory meets proof theory: proof certificates for Büchi inclusion

In a joint work with David Baelde and Lucca Hirschi, Amina Doumane and Alexis Saurin carried out a proof-theoretical investigation of the linear-time μ-calculus, proposing well-structured proof systems and showing constructively that they are complete for inclusions of Büchi automata suitably encoded as formulas.

They do so in a way that combines the advantages of two lines of previous work: Kaivola gave a proof of completeness for an axiomatisation that amounts to a finitary proof system, but his proof is non-constructive and yields no reasonable procedure. On the other hand, Dax, Hofmann and Lange recently gave a deductive system that is appropriate for algorithmic proof search, but their proofs require a global validity condition and do not have a well understood proof theory.

They work with well-structured proof systems, effectively constructing proofs in a finitary sequent calculus that enjoys local correctness and cut elimination.

This involves an intermediate circular proof system in which one can obtain proofs for all inclusions of parity automata, by adapting Safra's construction. In order to finally obtain finite proofs of Büchi inclusions, a translation result from circular to finite proofs is designed.

6.3. Effective higher dimensional algebra

Participants: Cyrille Chenavier, Pierre-Louis Curien, Yves Guiraud, Maxime Lucas, Philippe Malbos, Jovana Obradović.

6.3.1. Rewriting methods for Artin monoids

With Stéphane Gaussent (ICJ, Univ. Saint-Étienne), Yves Guiraud and Philippe Malbos have used higher-dimensional rewriting methods for the study of Artin monoids, a class of monoids that is fundamental in algebra and geometry. This work formulates in a common language several known results in combinatorial group theory: one by Tits about the fundamental group of a graph associated to an Artin monoid [76], and one by Deligne about the actions of Artin monoids on categories [58], both originally proved by geometrical and non-constructive methods. An improved completion procedure, called the homotopical completion-reduction procedure (see also [8]), is formalised and used to give constructive proofs of (improved versions of) both theorems. This work has been published in Compositio Mathematica [19] and has been implemented in a Python library (http://www.pps.univ-paris-diderot.fr/~guiraud/cox/cox.zip).

6.3.2. Rewriting and Garside theory

Yves Guiraud has collaborated with Patrick Dehornoy (LNO, Univ. Caen) to develop an axiomatic setting for monoids with a special notion of quadratic normalisation map with good computational properties. This theory generalises the normalisation procedure known for monoids that admit a special family of generators called a Garside family [57] to a much wider class that also includes the plactic monoids. It is proved that good quadratic normalisation maps correspond to quadratic convergent presentations, together with a sufficient condition for this to happen, based on the shape of the normalisation paths on length-three words. This work has been submitted for publication to the Journal de l'École Polytechnique — Mathématiques [44].

Building on this last article, Yves Guiraud currently collaborates with Matthieu Picantin (Automates team, LIAFA, Univ. Paris 7) to generalise the main results of [19] to monoids with a Garside family. This will allow an extension of the field of application of the rewriting methods to other geometrically interesting classes of monoids, such as the dual braid monoids.

6.3.3. Higher-dimensional linear rewriting

With Eric Hoffbeck (LAGA, Univ. Paris 13), Yves Guiraud and Philippe Malbos have introduced in [64] the setting of linear polygraphs to formalise a theory of linear rewriting, generalising Gröbner bases. They have adapted the computational method of [7] to compute polygraphic resolutions of associative algebras, with applications to the decision of the Koszul homological property. They are currently engaged into a major overhaul of this work, whose main goal is to ease the adaptation of the results to other algebraic varieties, like commutative algebras or Lie algebras.

6.3.4. Theory of reduction operators

Cyrille Chenavier, supervised by Yves Guiraud and Philippe Malbos, explores the use of Berger's theory of reduction operators [50] to design new rewriting methods in algebra. In [42], he proposed a construction of a contracting homotopy for the Koszul complex of an algebra (a complex characterising the homological property of Koszulness): when an algebra admits a side-confluent presentation (a strong hypothesis of confluence), he gave a candidate for the contracting homotopy, built using specific representations of confluence algebras; when the presentation satisfies an additional condition, called the extra-condition, it turns out that this candidate works.

6.3.5. Rewriting methods for coherence

In [45], Maxime Lucas, supervised by Yves Guiraud and Pierre-Louis Curien, has applied the rewriting techniques of [65] to prove coherence theorems for bicategories and pseudofunctors. He obtained a coherence theorem for pseudonatural transformations thanks to a new theoretical result, improving on the former techniques, that relates the properties of rewriting in 1- and 2-categories.

6.3.6. Wiring structure of operads and operad-like structures

Building on recent ideas of Marcelo Fiore on the one hand, and of François Lamarche on the other hand, Pierre-Louis Curien and Jovana Obradović developed a syntactic approach, using some of the kit of Curien-Herbelin's duality of computation and its polarised versions by Munch and Curien, to the definition of various structures that have appeared in algebra under the names of operads, cyclic operads, dioperads, properads, modular and wheeled operads, permutads, etc. These structures are defined in the literature in different flavours. The goal is to formalise the proofs of equivalence between these different styles of definition. This work is completed for cyclic operads and was presented at the conference Category Theory 2015 in Aveiro [43]. Further work will be to make these proofs modular, so as not to repeat them for each variation of the notion of operad.

6.3.7. A graphical proof of the Bénabou-Roubaud theorem

As a substantial development of reasoning with string diagrams, Jovana Obradović gave a complete proof of the Bénabou-Roubaud monadic descent theorem in [47]. One of the essential points concerning Grothendieck's original approach to descent theory consists of identifying the class of effective descent morphisms for a given fibration. In the special case of a bifibration satisfying Beck-Chevalley condition, Bénabou and Roubaud have given such a characterisation by means of monadicity. Due to the technically complicated calculations involving Grothendieck's cocycle condition, the categorical equivalence which reflects the comparison of the descent in fibered categories with monadic descent is usually not worked out in complete detail in the literature. Jovana Obradović linked the monadic and the original viewpoint via another possible definition of the category of descent data. This intermediate step, due to Janelidze and Tholen, involves constructions in internal categories and it provides an example on how one can stay in the world of string diagrams even when dealing with morphisms which do not have an explicit string diagram definition.

6.4. Incrementality

Participants: Yann Régis-Gianas, Lourdes Del Carmen González Huesca, Thibaut Girka.

An optimisation to perform incremental computations was developed by Lourdes del Carmen González Huesca and Yann Régis-Gianas, providing a mechanism to achieve efficiency. Incrementality as a way to propagate an input change into a corresponding output change is guided by formal change descriptions over terms and dynamic differentiation of functions. The data-changes are represented by displaceable types, a general framework to displace terms directed by types. An extension of the simply-typed lambda-calculus with differentials and partial derivatives offers a language to reason about incrementality. The basic system, λ-diff, was enriched with expressions for fixed-points and data-types together with their corresponding derivatives to analyse incrementality over them. The above results are reported in the second part of Lourdes González Huesca PhD thesis [16].

In collaboration with Paolo Giarrusso and Yufei Cai (Univ Marburg, Allemagne), Yann Régis-Gianas developed a new method to incrementalise higher-order programs using formal derivatives and static caching. A paper is in preparation.

In collaboration with David Mentré (Mitsubishi), Thibaut Girka and Yann Régis-Gianas designed and certified a new algorithm for correlating program generation: such a program is used to characterise the differences between two close programs. (Therefore, a correlating program is a good input for an incremental static analyser.) Before their work, only one algorithm existed in the literature and it was unsound. The new algorithm is sound and certified in Coq. This work has been published in the ATVA conference. Thibaut Girka has presented this work [33] at ATVA 2015.

In collaboration with David Mentré (Mitsubishi), Thibaut Girka and Yann Régis-Gianas are developing a theoretical framework to define a notion of differential operational semantics: a general mathematical object to characterise the difference of behavior of two close programs.

6.5. Metatheory and development of Coq

Participants: Pierre-Louis Curien, Hugo Herbelin, Pierre Letouzey, Yann Régis-Gianas, Matthieu Sozeau.

6.5.1. Models of type theory

Simplicial sets and their extensions as Kan complexes can serve as models of homotopy type theory. Hugo Herbelin extended his concrete type-theoretic formalisation of semi-simplicial sets [20] to simplicial sets.

6.5.2. Unification

Matthieu Sozeau is working in collaboration with Beta Ziliani (PhD at MPI-Saarbrücken, now assistant professor at Cordoba, Argentina) on formalising the unification algorithm used in Coq, which is central for working with advanced type inference features like Canonical Structures. This is the first precise formalisation of all the rules of unification including the ones used for canonical structure resolution and universes. The presentation includes a careful study of the heuristics used in the existing Coq algorithms, which can be added or removed from the new implementation modularly. This work has been presented at the ICFP'15 conference [31].

6.5.3. Nominal techniques

Matthieu Sozeau cosupervised the internship of Gabriel Lewertowski with Nicolas Tabareau (Ascola team, Nantes), on the development of a library for nominal reasoning in Coq/Ssreflect. The goal of this internship was to study the use of nominal sets to ease the formalisation of programming language (meta-)theory. A library based on the Mathematical Components formalisation of finite sets and effective quotients was built, providing generic definitions of substitution and elimination operators for simple descriptions of programming language syntax as a grammar. This work was done in collaboration with Assia Mahboubi (Specfun) and Cyril Cohen (Marelle). It forms the basis for the formalisation of cubical type theory, a new type theory using name abstraction that implements an axiom-free version of Homotopy Type Theory.

SUMO Project-Team

7. New Results

7.1. Model expressivity and quantitative verification

7.1.1. Diagnosability of stochastic systems

Participants: Nathalie Bertrand, Engel Lefaucheux.

Diagnosis of partially observable stochastic systems prone to faults was introduced in the late nineties. Diagnosability, i.e. the existence of a diagnoser, may be specified in different ways: (1) exact diagnosability (called A-diagnosability) requires that almost surely a fault is detected and that no fault is erroneously claimed while (2) approximate diagnosability (called ϵ-diagnosability) allows a small probability of error when claiming a fault and (3) accurate approximate diagnosability (called AA-diagnosability) requires that this error threshold may be chosen arbitrarily small. In [32] we mainly focus on approximate diagnoses. We first refine the almost sure requirement about finite delay introducing a uniform version and showing that while it does not discriminate between the two versions of exact diagnosability this is no more the case in approximate diagnosis. Then we establish a complete picture for the decidability status of the diagnosability problems: (uniform) ϵ-diagnosability and uniform AA-diagnosability are undecidable while AA-diagnosability is decidable in PTIME, answering a longstanding open question.

7.1.2. Probabilistic model checking

Participants: Blaise Genest, Ocan Sankur.

In [16], we considered the verification of Markov chains against properties talking about distributions of probabilities. Even though a Markov chain is a very simple formalism, by discretizing in a finite number of classes the space of distributions through some symbols, we proved that the language of trajectories of distributions (one for each initial distribution) is not regular in general, even with 3 states. We then proposed a parametrized algorithm which approximates what happens to infinity, such that each symbolic block in the approximate language is at most ϵ away from the concrete distribution. We proved in [26] that if the eigenvalues of the Markov chain are distinct positive real numbers, then the trajectory is effectively regular. This is however not the case anymore if the eigenvalues can be distinct roots of real numbers.

Markov decision processes (MDPs) with multi-dimensional weights are useful to analyze systems with multiple objectives that may be conflicting and require the analysis of trade-offs. In [40], we study the complexity of percentile queries in such MDPs and give algorithms to synthesize strategies that enforce such constraints. Given a multi-dimensional weighted MDP and a quantitative payoff function f, thresholds v_i (one per dimension), and probability thresholds α_i, we show how to compute a single strategy to enforce that for all dimensions i, the probability of outcomes ρ satisfying $f_i(\rho) \geq v_i$ is at least α_i. We consider classical quantitative payoffs from the literature (sup, inf, lim sup, lim inf, mean-payoff, truncated sum, discounted sum). Our work extends to the quantitative case the multi-objective model checking problem studied by Etessami et al. [48] in unweighted MDPs.

In the invited contribution [25], we revisit the stochastic shortest path problem, and show how recent results allow one to improve over the classical solutions: we present algorithms to synthesize strategies with multiple guarantees on the distribution of the length of paths reaching a given target, rather than simply minimizing its expected value. The concepts and algorithms that we propose here are applications of more general results that have been obtained recently for Markov decision processes and that are described in a series of recent papers, including [40].

7.1.3. Stochastic modeling of biological systems

Participants: Blaise Genest, Éric Fabre, Sucheendra Palaniappan, Matthieu Pichené.

In [47], we model a population of Hela cells with non deterministic behavior, subject to the drug TRAIL. TRAIL kills a large fraction of cancerous Hela cells by triggering the apoptosis pathway. Modelling this survival is important to perform *in silico* computations helping designing treatments killing the largest fraction of cancerous cells. We model this system using the stochastic class of Dynamic Bayesian Networks. We maintain large conditional probability tables which are represented by sparse datastructure, and perform simulations by looking ahead one time step and factoring this information to avoid empty probability entries. This considerably improves the simulation based inference of DBNs, getting a 100 times improvement in its efficiency.

7.1.4. Robustness of timed models

Participants: Ocan Sankur, Loïc Hélouët.

Robustness of timed systems aims at studying whether infinitesimal perturbations in clock values can result in new discrete behaviors. A model is robust if the set of discrete behaviors is preserved under arbitrarily small (but positive) perturbations. This year we tackled this problem both for Timed Automata and time Petri Nets.

Timed automata are an extension of finite automata with clock variables that can conveniently model real-time systems. In [42], we study the robustness analysis problem for timed automata under guard imprecisions which consists in computing a timing imprecision bound under which a given specification holds. This is a particular kind of parameter synthesis problems specialized for analyzing robustness. We give a symbolic semi-algorithm for the problem based on a parametric data structure, and evaluate its performance in comparison with a recently published one, and with a binary search on the imprecision bound. We show that a safe bound on imprecision can be computed efficiently, and a performance close to that of exact model checking can be obtained thanks to the use of the parametric data structure and cycle acceleration techniques.

Another related problem is that of robust controller synthesis for timed automata where the goal is to choose actions and their timings so as to ensure a given state is reached when the chosen time delays are adversarially perturbed within a bound. In [21], we are interested in synthesizing “robust” strategies for ensuring reachability of a location in timed automata. We model this problem as a game between the controller and its environment, and solve the parameterized robust reachability problem: we show that the existence of an upper bound on the perturbations under which there is a strategy reaching a target location is EXPTIME-complete. We also extend our algorithm, with the same complexity, to turn-based timed games, where the successor state is entirely determined by the environment in some locations.

We also tackled the robustness problem for time Petri nets (TPNs, for short) in [17] by considering the model of parametric guard enlargement which allows time-intervals constraining the firing of transitions in TPNs to be enlarged by a (positive) parameter. We show that TPNs are not robust in general and checking if they are robust with respect to standard properties (such as boundedness, safety) is undecidable. We then extend the marking class timed automaton construction for TPNs to a parametric setting, and prove that it is compatible with guard enlargements. We apply this result to the (undecidable) class of TPNs which are robustly bounded (i.e., whose finite set of reachable markings remains finite under infinitesimal perturbations): we provide two decidable robustly bounded subclasses, and show that one can effectively build a timed automaton which is timed bisimilar even in presence of perturbations. This allows us to apply existing results for timed automata to these TPNs and show further robustness properties.

7.1.5. Verification for classes of Petri Nets with time

Participants: Blaise Genest, Loïc Hélouët.

We have considered verification problems for classes of Petri Nets with time. We have introduced the first, up to our knowledge, decidability result on reachability and boundedness for Petri Net variants that combine unbounded places, time, and urgency (the ability to enforce actions to happen within some delay). For this, we introduce the class of Timed-Arc Petri Nets with Urgency, which extends Timed-Arc Petri Nets [58] to allow urgency constraints, a feature from Timed-transition Petri Nets (TPNs) [54]. In order to avoid (straightforward) undecidability, we have considered restricted urgency: urgency can be used only on transitions consuming tokens from bounded places. For Timed-Arc Petri Nets with restricted urgency, we extend decidability results from Timed-Arc Petri Nets: control-state reachability and boundedness are decidable. Our main result concerns (marking) reachability, which is undecidable for both TPNs (because of unrestricted urgency) [52] and Timed-Arc Petri Nets (because of infinite number of clocks) [57]. We have obtained decidability of reachability for (unbounded) TPNs with restricted urgency under a new, yet natural, timed-arc semantics presenting them as Timed-Arc Petri Nets with restricted urgency. Decidability of reachability under the original semantics of TPNs was also obtained for a restricted subclass of unbounded nets. This work is under submission.

7.1.6. Non-interference in partial order models

Participant: Loïc Hélouët.

In [36] we have proposed a new definition of interference for partial order models. Non-interference (NI) is a property of systems stating that confidential actions should not cause effects observable by unauthorized users. Several variants of NI have been studied for many types of models, but rarely for true concurrency or unbounded models. In [36] we have investigated NI for High-level Message Sequence Charts (HMSC), a scenario language for the description of distributed systems, based on composition of partial orders. We firstly have proposed a general definition of security properties in terms of equivalence among observations, and shown that these properties, and in particular NI are undecidable for HMSCs. We hence have considered weaker local properties, describing situations where a system is attacked by a single agent, and show that local NI is decidable in this context. We then have proposed a refinement of local NI to obtain a finer notion of causal NI that emphasizes causal dependencies between confidential actions and observations. This causal NI has then been extended to causal NI with (selective) declassification of confidential events. Finally, we have shown that checking whether a system satisfies local and causal NI and their declassified variants are PSPACE-complete problems. Decidability seems to extend to other classes of partial order models which partially ordered observations can be represented by partial order models that exhibit some forms of regularity such as graph grammars or partial order automata. This conjecture will be explored next year.

7.1.7. Synthesis and games

Participants: Ocan Sankur, Engel Lefaucheux.

In [33], we investigate compositional algorithms to solve safety games described succinctly by synchronous circuits (given by AND and inverter gates). We show how the safety specification can be decomposed, in most cases, into a set of simpler specifications, each defining a safety game depending on less inputs and state variables. We give several algorithms which consist in solving the subgames, and aggregating them in order to find strategies for the global game. We present results of extensive experiments done on around five hundred benchmarks used in the synthesis competition SYNTCOMP 2014 and show that the compositional approach improves the performence on several classes of benchmarks.

In [35] we investigate priced timed games. Priced timed games are two-player zero-sum games played on priced timed automata (whose locations and transitions are labeled by weights modeling the costs of spending time in a state and executing an action, respectively). The goals of the players are to minimise and maximise the cost to reach a target location, respectively. We consider priced timed games with one clock and arbitrary (positive and negative) weights and show that, for an important subclass (the so-called simple priced timed games), one can compute, in exponential time, the optimal values that the players can achieve, with their associated optimal strategies. As side results, we also show that one-clock priced timed games are determined and that we can use our result on simple priced timed games to solve the more general class of so-called reset-acyclic priced timed games (with arbitrary weights and one-clock).

In [34], we introduce a novel rule for synthesis of reactive systems, applicable to systems made of n components which have each their own objectives. This rule is based on the notion of admissible strategies. Intuitively, a strategy σ is dominated by σ' if against all strategies of other players, σ' is as good as σ, and against at least one strategy σ' is strictly better than σ. Admissible strategies are those that are not dominated by any other strategy. The assume-admissible synthesis consists in restricting the space of strategies to admissible ones, and to look for strategy profiles which satisfy given specifications. We compare this rule with previous rules defined in the literature, and show that contrary to the previous proposals, it defines sets of solutions which are rectangular. This property leads to solutions which are robust and resilient, and allows one to synthesize strategies separately for each agent. We provide algorithms with optimal complexity and also an abstraction framework compatible with the new rule.

7.2. Management of large distributed systems

7.2.1. Parameterized verification in parameterized networks

Participants: Nathalie Bertrand, Paulin Fournier.

We study the problems of reaching a specific control state, or converging to a set of target states, in networks with a parameterized number of identical processes communicating via broadcast. To reflect the distributed aspect of such networks, we restrict our attention to executions in which all the processes must follow the same local strategy that, given their past performed actions and received messages, provides the next action to be performed. We show that the reachability and target problem under such local strategies are NP-complete, assuming that the set of receivers is chosen non-deterministically at each step. On the other hand, these problems become undecidable when the communication topology is a clique. However, decidability can be regained with the additional assumption that all processes are bound to receive the broadcast messages. This is a joint work with Arnaud Sangnier [31].

7.2.2. Runtime enforcement of untimed and timed properties

Participants: Thierry Jéron, Hervé Marchand, Srinivas Pinisetty.

Runtime enforcement is a powerful technique to ensure that a running system satisfies some desired properties. Using an enforcement monitor, an (untrustworthy) input execution (in the form of a sequence of events) is modified into an output sequence that complies with a property. Over the last decade, runtime enforcement has been mainly studied in the context of untimed properties. For several years, and in particular in the context of the PhD thesis of Srinivas Pinisetty [15] we elaborated the theory of runtime enforcement of timed properties. This year we also continued our work on the subject in several directions.

In [38] we describe the TiPEX tool that implements the enforcement monitoring algorithms for timed properties proposed in our previous papers . Enforcement monitors are generated from timed automata specifying timed properties. Such monitors correct input sequences by adding extra delays between events. Moreover, TiPEX also provides modules to generate timed automata from patterns, compose them, and check the class of properties they belong to in order to optimize the monitors. This paper also presents the performance evaluation of TiPEX within some experimental setup.

With coleagues from LaBRI (M. Renard, A. Rollet) and LIG (Y. Falcone) we investigate runtime enforcement of (timed and untimes) properties with uncontrollable events. In [41], we introduce a framework that takes as input any regular (timed) property over an alphabet of events, with some of these events being uncontrollable. An uncontrollable event cannot be delayed nor intercepted by an enforcement mechanism. Enforcement mechanisms satisfy important properties, namely soundness and compliance, meaning that enforcement mechanisms output correct executions that are close to the input execution. We discuss the conditions for a property to be enforceable with uncontrollable events, and we define enforcement mechanisms that modify executions to obtain a correct output, as soon as possible. Moreover, we synthesize sound and compliant descriptions of runtime enforcement mechanisms at two levels of abstraction to facilitate their design and implementation.

With colleagues from the Aalto University (S. Pinisetty, S. Tripakis and V. Preoteasa) and LIG (Y. Falcone) we investigate predictive runtime enforcement. In [39] we introduce predictive runtime enforcement, where the system is not entirely black-box, but we know something about its behavior. This a-priori knowledge about the system allows to output some events immediately, instead of delaying them until more events are observed, or even blocking them permanently. This in turn results in better enforcement policies. We also show that if we have no knowledge about the system, then the proposed enforcement mechanism reduces to a classical non-predictive RE framework. All our results are formalized and proved in the Isabelle theorem prover. We are also currently extending this work to the timed setting.

7.2.3. Discrete controller synthesis

Participants: Nicolas Berthier, Hervé Marchand.

In [29] we investigate the opportunities given by recent developments in the context of Discrete Controller Synthesis algorithms for infinite, logico-numerical systems. To this end, we focus on models employed in previous work for the management of dynamically partially reconfigurable hardware architectures. We extend these models with logico-numerical features to illustrate new modeling possibilities, and carry out some benchmarks to evaluate the feasibility of the approach on such models.

In [30] we elaborate on our former work for the safety control of infinite reactive synchronous systems modeled by arithmetic symbolic transition systems. By using abstract interpretation techniques involving disjunctive polyhedral overapproximations, we provide effective symbolic algorithms allowing to solve the deadlock-free safety control problem while overcoming previous limitations regarding the non-convexity of the set of states violating the invariant to enforce.

The ever growing complexity of software systems has led to the emergence of automated solutions for their management. The software assigned to this work is usually called an Autonomic Management System (AMS). It is ordinarily designed as a composition of several managers, which are pieces of software evaluating the dynamics of the system under management through measurements (e.g., workload, memory usage), taking decisions, and acting upon it so that it stays in a set of acceptable operating states. However, careless combination of managers may lead to inconsistencies in the taken decisions, and classical approaches dealing with these coordination problems often rely on intricate and ad hoc solutions. To tackle this problem, we take a global view and underscore that AMSs are intrinsically reactive, as they react to flows of monitoring data by emitting flows of reconfiguration actions. Therefore in [19] we propose a new approach for the design of AMSs, based on synchronous programming and discrete controller synthesis techniques. They provide us with high-level languages for modeling the system to manage, as well as means for statically guaranteeing the absence of logical coordination problems. Hence, they suit our main contribution, which is to obtain guarantees at design time about the absence of logical inconsistencies in the taken decisions. We detail our approach, illustrate it by designing an AMS for a realistic multi-tier application, and evaluate its practicality with an implementation.

In the invited paper [24] we make an overview of our works addressing discrete control-based design of adaptive and reconfigurable computing systems, also called autonomic computing. They are characterized by their ability to switch between different execution modes w.r.t. application and functionality, mapping and deployment, or execution architecture. The control of such reconfigurations or adaptations is a new application domain for control theory, called feedback computing. We approach the problem with a programming language supported approach, based on synchronous languages and discrete control synthesis. We concretely use this approach in FPGA-based reconfigurable architectures, and in the coordination of administration loops.

7.2.4. Computing knowledge at runtime

Participant: Blaise Genest.

In [37] we compare three notions of knowledge in concurrent system: memoryless knowledge, knowledge of perfect recall, and causal knowledge. Memoryless knowledge is based only on the current state of a process, knowledge of perfect recall can take into account the local history of a process, and causal knowledge depends on the causal past of a process, which comprises the information a process can obtain when all processes exchange the information they have when performing joint transitions. We compare these notions in terms of knowledge strength, number of bits required to store this information, and the complexity of checking if a given process has a given knowledge. We show that all three notions of knowledge can be implemented using finite memory. Causal knowledge proves to be strictly more powerful than knowledge with perfect recall, which in turn proves to be strictly more powerful than memoryless knowledge. We show that keeping track of causal knowledge is cheaper than keeping track of knowledge of perfect recall.

7.2.5. Distributed optimal planning

Participant: Éric Fabre.

Planning problems consist in organizing actions in a system in order to reach one of some target states. The actions consume and produce resources, can of course take place concurrently, and may have costs. We have a collection of results addressing this problem in the setting of distributed systems. This takes the shape of a network of components, each one holding private actions operating over its own resources, and shared/synchronized actions that can only occur in agreement with its neighbors. The goal is to design in a distributed manner a tuple of local plans, one per component, such that their combination forms a consistent global plan of minimal cost.

Our previous solutions to this problem modeled components as weighted automata [22]. In collaboration with Loig Jezequel (TU Munich) and Victor Khomenko (Univ. of Newcastle), we have extended this approach to the case of components modeled as safe Petri nets[23]. This allows one to benefit from the internal concurrency of actions within a component. Benchmarks have shown that this method can lead to significant time reductions to find feasible plans, in good cases. In the least favorable cases, performances are comparable to those obtained with components modeled as automata. The method does not apply to all situations however, as computations require to perform ϵ-reductions on Petri-nets (our work also contains a contribution to this difficult question).

7.2.6. Regulation of urban train systems

Participants: Éric Fabre, Loïc Hélouët, Karim Kecir, Hervé Marchand, Christophe Morvan.

A part of the SUMO team is involved in a collaboration with Alstom transports on regulation techniques. The role of regulation algorithms is to observe train trajectories and delays with respect to an expected ideal schedule, and then compute commands that are sent to trains to meet some quality of service (punctuality, regularity, ...) The objective of this collaboration is to study regulation techniques that are currently in use in urban train systems and compare their performances, and in the future to be able to compute optimal regulation strategies.

This year, we have proposed models inspired from stochastic Petri nets and from closed loop controllers to simulate regulated railways systems. The Petri net model led to the design of a tool called SIMSTORS, that was sucessfully used to model a real case study (line 1 of Santiago's subway). The simulator relies on event-based symbolic techniques: the time elapsed between two steps of the simulation is the time between two event occurrences (arrival, departure of a train, incident,...). This simulation scheme relying on an abstract model allowed a dramatic speed up of simulation with respect to existing solutions in use at Alstom Transport.

A second line of work has also been explored, in order to design and evaluate new regulation strategies for subway lines. The underlying model is inspired from event-based control theory, in a stochastic and timed setting. It abstracts away several significant topological features of a subway line, and focuses on the optimal command of train speeds in order to achieve high-level objectives such as the equal spacing of trains, or the efficient insertion/extraction of trains. This approach has allowed us to design new distributed regulation policies, which are remarquably stable and efficiently mitigate known instabilities of subway lines, like the bunching phenomenon. We are currently working on an extension of this approach for the management of time-tables and of forks and joins in the topology of subway lines.

7.3. Data driven systems

7.3.1. A model of large-scale distributed collaborative system

Participants: Éric Badouel, Loïc Hélouët, Christophe Morvan, Robert Nsaibirni.

We have presented in [27] and [18] a purely declarative approach to artifact-centric collaborative systems, a model which we introduced in two stages. First, we assume that the workspace of a user is given by a mindmap, shortened to a map, which is a tree used to visualize and organize tasks in which he or she is involved, together with the information used for the resolution of these tasks. We introduce a model of guarded attribute grammar, or GAG, to help the automation of updating such a map. A GAG consists of an underlying grammar, that specifies the logical structure of the map, with semantic rules which are used both to govern the evolution of the tree structure (how an open node may be refined to a subtree) and to compute the value of some of the attributes (which derives from con-textual information). The map enriched with this extra information is termed an active workspace. Second, we define collaborative systems by making the various user's active workspaces communicate with each other. The communication uses message passing without shared memory thus enabling convenient distribution on an asynchronous architecture. A case study on a disease surveillance system is under development in the PhD thesis of Robert Nsaibirni and a first prototype of the model of active workspaces was written by Eric Badouel.

7.3.2. Petri Nets with semi-structured data

Participants: Éric Badouel, Loïc Hélouët, Christophe Morvan.

In [28], we have proposed an extension of Petri nets with data called Structured Data Nets (StDN). This extension allows for the description of transactional systems with data. In StDNs, tokens are structured documents. Each transition is attached to a query, guarded by patterns, (logical assertions on the contents of its preset) and transforms tokens. In [28], we have proposed a semantics for StDNs, and then considered their formal properties: coverability of a marking, termination and soundness of transactions. Unrestricted StDNs are Turing complete, so these properties are undecidable. However, we have proposed an order on structured documents, and shown that under reasonable restrictions on documents and on the ex- pressiveness of patterns and queries, StDNs are well-structured transition systems, for which coverability, termination and soundness are decidable. This work has then been extended to consider properties of sets of configurations described as upward closed sets satisfying patterns, and should appear in a journal paper in 2016.

TOCCATA Project-Team

7. New Results

7.1. Deductive Verification

- M. Clochard, J.-C. Filliâtre, and A. Paskevich proposed a novel method to prove the relative safety of operations over bounded integers in a large class of programs. Their approach consists of introducing dedicated abstract types for the bounded integers and restricting the set of allowed operations over these types in such a way that it is impossible to reach the bound during a realistic execution of the program: for example, it would take several hundred years to overflow a 64-bit integer. This technique is aimed at integer variables that serve essentially as counters or size measures. It can be used alongside the traditional methods of proving the absence of overflows for other integer values in the same program. The proposed approach is implemented in Why3 and was presented at VSTTE 2015 [26].
- J.-C. Filliâtre and M. Pereira proposed a new way to specify the behavior of a cursor data structure, with the objective of being able to verify both the implementation of a cursor and its use by client code. The approach is modular, which means that a program using a cursor can be verified independently of the way the cursor is implemented. An experimental evaluation has been conducted with Why3, with several implementations and client codes being verified. This work will be presented at JFLA 2016 [26].
- C. Fumex and C. Marché developed a new library for bit-vectors, in Why3 and SPARK [30]. This library is rich enough for the formal specification of functional behavior of programs that operate at the level of bits. It is also designed to exploit efficiently the support for bit-vectors built-in in some SMT solvers. This work is done in the context of the ProofInUse joint laboratory. The SPARK front-end of Why3, for the verification of Ada programs, is extended to exploit this new bit-vector theory. Several cases studies are conducted: efficient search for rightmost bit of a bit-vector, efficient computation of the number of bits set to 1, efficient solving of the n-queens problem. At the level of SPARK, a program inspired from some industrial code (originally developed in C par J. Gerlach, Fraunhofer FOKUS Institute, Germany and partially proved with Frama-C and Coq) is specified in SPARK and proved with automatic solvers only. The support for bit-vectors is already distributed with SPARK, and SPARK users already reported that several verification conditions, that couldn't be proved earlier, are now proved automatically.
- D. Hauzar and C. Marché worked on counterexample generation from failed proof attempts. They designed a new approach for generating potential counterexamples in the deductive verification setting, and implemented in Why3. When the logic goal generated for a given verification condition is not shown unsatisfiable by an SMT solvers, some solver can propose a model. By carefully reverting the transformation chain (from an input program through the VC generator and the various translation steps to solvers), this model is turned into a potential counterexample that the user can exploit to analyze why its original code is not proved. The approach is implemented in the chain from Ada programs through SPARK, Why3, and SMT solvers CVC4 and Z3. This implementation is robust enough to be distributed in the next release Pro 16 of SPARK. A research report on this subject will appear in January 2016.
- A. Charguéraud and F. Pottier (Inria Paris-Rocquencourt) obtained new results in the machine-checked verification of asymptotic complexity bounds, in addition to program correctness properties. Verifying the time usage of a program is very important, because otherwise a program might be proved to be functionally correct but may appear to run into an infinite loop for particular input data. More specifically, A. Charguéraud and F. Pottier started from the extension of CFML with *time credits* (encoding of time resources in Separation Logic), developed last year by A. Charguéraud, and

they used it to formally produce a machine-checked proof of the correctness and time complexity of a Union-Find data structure, implemented as an OCaml module. They thereby demonstrate that the approach scales up to difficult complexity analyses, and applies to actual executable code (as opposed to pseudo-code). This work was presented at ITP 2015 [24]. Furthermore, A. Charguéraud and F. Pottier co-advised the M2 internship of Armaël Guéneau, who extended the time credits approach so as to allow working conveniently with the big-O notation. He extended the CFML library and verified the time complexity of a binary random access list data structure due to Okasaki. This work has not been published yet.

- A. Charguéraud described a method for reasoning about mutable data structures that own their elements. In Separation Logic, representation predicates describe the ownership of a mutable data structure, by establishing a relationship between the entry point of the structure, the piece of heap over which this structure spans, and the logical model associated with the structure. When a data structure is polymorphic, such as in the case of a container, its representation predicate needs to be parameterized not just by the type of the items stored in the structure, but also by the representation predicates associated with these items. Such higher-order representation predicates can be used in particular to control whether containers should own their items. A. Charguéraud wrote a paper describing, through a collection of practical examples, solutions to the challenges associated with reasoning about accesses into data structures that own their elements. This paper will appear at CPP 2016 [23].

7.2. Automated Reasoning

- C. Dross, A. Paskevich, J. Kanig and S. Conchon published a journal paper [16] about integration of first-order axiomatizations with triggers as decision procedures in an SMT solver. This work extends a part of C. Dross PhD thesis [79]. A formal semantics of the notion of trigger is presented, with a general setting to show how a first-order axiomatization with triggers can be proved correct, complete, and terminating. An extended DPLL(T) algorithm can then integrate such an axiomatization with triggers, as a decision procedure for the theory it defines.

7.3. Certification of Languages, Tools and Systems

- M. Clochard and L. Gondelman developed a formalization of a simple compiler in *Why3*. It compiles a simple imperative language into assembler instructions for a stack machine. This case study was inspired by a similar example developed using Coq and interactive theorem proving. The aim is to improve significantly the degree of automation in the proofs. This is achieved by the formalization of a Hoare logic and a Weakest Precondition Calculus on assembly programs, so that the correctness of compilation is seen as a formal specification of the assembly instructions generated. This work was presented at the JFLA conference in 2015 [25].
- S. Boldo, C. Lelay, and G. Melquiond worked on the Coquelicot library, designed to be a user-friendly Coq library about real analysis. An easier way of writing formulas and theorem statements is achieved by relying on total functions in place of dependent types for limits, derivatives, integrals, power series, and so on. To help with the proof process, the library comes with a comprehensive set of theorems and some automation. We have exercised the library on several use cases: in an exam at university entry level, for the definitions and properties of Bessel functions, and for the solution of the one-dimensional wave equation. These results are published in the journal *Mathematics in Computer Science* [14].
- C. Lelay developed a new formalization of convergence with a focus on usability and genericity for the Coquelicot library. This formalization covers various parts of analysis: sequences, real functions, complex functions, vector functions, and so on. This work was presented at the 7th Coq Workshop [27].
- C. Paulin wrote a gentle introduction to the Calculus of Inductive Construction, the formalism on which the Coq proof assistant is based [28], discussing both theoretical and pragmatic aspects of the design.

7.4. Floating-Point and Numerical Programs

- É. Martin-Dorel and G. Melquiond worked on integrating the CoqInterval and CoqApprox libraries into a single package. The CoqApprox library is dedicated to computing verified Taylor models of univariate functions so as to compute approximation errors. The CoqInterval library reuses this work to automatically prove bounds on real-valued expressions. A large formalization effort took place during this work, so as to get rid of all the holes remaining in the formal proofs of CoqInterval. It was also the chance to perform a comparison between numerous decision procedures dedicated to proving nonlinear inequalities involving elementary functions. This work has been published in the *Journal of Automated Reasoning* [18].
- S. Boldo and G. Melquiond, with J.-H. Jourdan and X. Leroy (Gallium team, Inria Paris - Rocquencourt) extended the CompCert compiler to get the first formally verified C compiler that provably preserves the semantics of floating-point programs This work, published in the *Journal of Automated Reasoning* [13], also covers the formalization of numerous algorithms of conversion between integers and floating-point numbers.
- S. Boldo worked on the fact that $a/\sqrt{(a^2+b^2)}$ is always in the interval $[-1,1]$ even when operations are done using floating-point arithmetic. This reduces to taking the square root of the square of a floating-point number as it is the worst case. Results in radix 2 (where $\sqrt{(a^2)} = |a|$) and other radices (where it might not hold) have been published at the 8th International Workshop on Numerical Software Verification [22].
- S. Boldo worked on programs computing the average of two floating-point numbers. As we want to take exceptional behaviors into account, we cannot use the naive formula `(x+y)/2`. Based on hints given by Sterbenz, she first wrote an accurate program and formally proved its properties. She also developed and formally proved a new algorithm that computes the correct rounding of the average of two floating-point numbers [21]. This was published at the 17th International Conference on Formal Engineering Methods.
- P. Roux formalized a theory of numerical analysis for bounding the round-off errors of a floating-point algorithm. This approach was applied to the formal verification of a program for checking that a matrix is semi-definite positive. The challenge here is that testing semi-definiteness involves algebraic number computations, yet it needs to be implemented using only approximate floating-point operations. This work has been published in the *Journal of Automated Reasoning* [19].
- C. Lelay and G. Melquiond worked on formalizing in Coq a numerical domain for the Verasco abstract interpreter built upon the CompCert verified compiler. This abstract domain is a relational domain based on affine forms (zonotopes). It is meant to help verifying floating-point programs and it is expected to perform faster (but less accurately) than a more generic domain based on polyhedrons.

7.5. Miscellaneous

- A. Charguéraud worked together with Umut Acar, Mike Rainey, and Filip Sieczkowski, as part of the ERC project *DeepSea*, on the development of efficient data structures and algorithms targeting modern, shared memory multicore architectures. A. Charguéraud was involved in two major results obtained this year.

  The first result is the development of fast and robust parallel graph traversal algorithms based on depth-first-search. This algorithm leverages a new sequence data structure for representing the set of edges remaining to be visited. This sequence itself builds on prior work on bootstrapped chunked sequences [35]. In particular, the edge sequence structure uses a balanced split operation for partitioning the edges of a graph among the several processors involved in the computation. Compared with prior work, the new algorithm is designed and proved to be efficient not just for particular classes of graphs, but for all input graphs. This work has been published in the ACM/IEEE Conference on High Performance Computing (SC) [20].

Another result by A. Charguéraud and his co-authors is the development of a calculus for parallel computing on shared memory computers. Many languages for writing parallel programs have been developed. These languages offer several different abstractions for parallelism, such as fork-join, async-finish, futures, etc. While they may seem similar, these abstractions lead to different semantics, language design and implementation decisions. In this work, we consider the question of whether it would be possible to unify different approaches to parallelism. To this end, we propose a calculus, called *DAG-calculus* that can encode existing approaches to parallelism based on fork-join, async-finish, and futures paradigms and possibly others. We have shown that the approach is realistic by presenting an implementation in C++ and by performing an empirical evaluation. This work has been submitted for publication.

- A. Charguéraud developed a patch to the OCaml compiler for improving type error messages, in particular to make the language more accessible to beginners. The problem of improving type error messages in ML has received quite a bit of attention over the past two decades, and many different strategies have been considered. The challenge is not only to produce error messages that are both sufficiently concise and systematically useful to the programmer, but also to handle a full-blown programming language and to cope with large-sized programs efficiently. A. Charguéraud's novel approach consists of a slight modification to the traditional ML type inference algorithm implemented in OCaml that, by significantly reducing the left-to-right bias, produces error messages that are more helpful to the programmer. This work was published this year in the journal Electronic Proceedings in Theoretical Computer Science [15].

VERIDIS Project-Team

7. New Results

7.1. Automated and Interactive Theorem Proving

Participants: Gabor Alági, Haniel Barbosa, Jasmin Christian Blanchette, Martin Bromberger, Simon Cruanes, Pablo Dobal, Mathias Fleury, Pascal Fontaine, Maximilian Jaroschek, Marek Košta, Stephan Merz, Martin Riener, Thomas Sturm, Hernán Pablo Vanzetto, Uwe Waldmann, Daniel Wand, Christoph Weidenbach.

7.1.1. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen from the CASSIS project-team at Inria Nancy – Grand Est, and Paula Chocron, a student at the University of Buenos Aires.

A satisfiability problem is often expressed in a combination of theories, and a natural approach consists in solving the problem by combining the satisfiability procedures available for the component theories. This is the purpose of the combination method introduced by Nelson and Oppen. However, in its initial presentation, the Nelson-Oppen combination method requires the theories to be signature-disjoint and stably infinite (to ensure the existence of an infinite model). The design of a generic combination method for non-disjoint unions of theories is clearly a hard task, but it is worth exploring simple non-disjoint combinations that appear frequently in verification. An example is the case of shared sets, where sets are represented by unary predicates. Another example is the case of bridging functions between data structures and a target theory (e.g., a fragment of arithmetic).

We defined [24] a sound and complete combination procedure à la Nelson-Oppen for the theory of absolutely free data structures (including lists and trees) connected to another theory via bridging functions. This combination procedure has also been refined for standard interpretations. The resulting theory has a nice politeness property, enabling combinations with arbitrary decidable theories of elements. We also investigated [25] other theories amenable to similar combinations: this class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between.

7.1.2. Adapting Real Quantifier Elimination Methods for Conflict Set Computation

The satisfiability problem in real closed fields is decidable. In the context of satisfiability modulo theories, the problem restricted to conjunctive sets of literals, that is, sets of polynomial constraints, is of particular importance. One of the central problems is the computation of good explanations of the unsatisfiability of such sets, i.e. obtaining a small subset of the input constraints whose conjunction is already unsatisfiable. We have adapted two commonly used real quantifier elimination methods, cylindrical algebraic decomposition and virtual substitution, to provide such conflict sets and demonstrate the performance of our method in practice [27].

7.1.3. Codatatypes and Corecursion

Joint work with Andrei Popescu and Dmitriy Traytel (Technische Universität München) and Andrew Reynolds (EPFL).

Datatypes and codatatypes are useful for specifying and reasoning about (possibly infinite) computational processes. The Isabelle/HOL proof assistant is being extended with flexible and convenient support for (co)datatypes and (co)recursive functions on them. We extended the emergent framework for (co)codatatypes with automatic generation of nonemptiness witnesses [22], nonemptiness being a proviso for introducing types in many logics, including Isabelle's higher-order logic. As a theoretical step towards a definitional mechanism in Isabelle, we formalized a framework for defining corecursive functions safely, based on corecursion up-to and relational parametricity [21]. The end product is a general corecursor that allows corecursive (and even recursive) calls under "friendly" operations—an improvement over the inflexible syntactic criteria of systems such as Agda and Coq.

In a related line of work, we improved the automation of the SMT solver CVC4 by designing, implementing, and evaluating a combined decision procedure for datatypes and codatatypes [31]. The procedure decides universal problems and is composable via the Nelson–Oppen method, as implemented in SMT solvers. The decision procedure for (co)datatypes is useful both for proving and for model finding. We have commenced work on a higher-order model finder based on CVC4, called Nunchaku, that relies heavily on the decision procedure.

7.1.4. Analysis and Generation of Structured Proofs

Joint work with Sascha Böhme (QAware GmbH), Maximilian Haslbeck and Tobias Nipkow (Technische Universität München), Daniel Matichuk (NICTA), and Steffen J. Smolka (Cornell University).

Isabelle/HOL is probably the most widely used proof assistant besides Coq. The Archive of Formal Proofs is a vast collection of computer-checked proofs developed using Isabelle, containing nearly 65 000 lemmas. We performed an in-depth analysis of the archive, looking at various properties of the proof developments, including size, dependencies, and proof style [18]. This give some insights into the nature of formal proofs.

In the context of the Sledgehammer bridge between automatic theorem provers and proof assistants, we designed a translation of machine-generated proofs into (semi-)intelligible Isabelle proofs that users can simply insert into their proof texts to discharge proof obligations [16]. While the output is designed for certifying the machine-generated proofs, it also has a pedagogical value: Unlike Isabelle's automatic tactics, which are black boxes, the proofs delivered by Sledgehammer can be inspected and understood. The direct proofs also form a good basis for manual tuning.

7.1.5. Encoding Set-Theoretic Formulas in Many-Sorted First-Order Logic

TLA$^+$ is a language for the formal specification of systems and algorithms whose first-order kernel is a variant of untyped Zermelo-Fraenkel set theory. Typical proof obligations that arise during the verification of TLA$^+$ specifications mix reasoning about sets, functions, arithmetic, tuples, and records. Encoding such formulas in the input languages of standard first-order provers (SMT solvers or superposition-based provers for first-order logic) is paramount for obtaining satisfactory levels of automation. For set theory, the basic idea is to represent membership as an uninterpreted predicate for the backend provers, and to reduce set-theoretic expressions to this basic predicate. This is not straightforward for formulas involving set comprehension or for proofs that rely on extensionality for inferring equality of sets. Moreover, a full development of set-theoretic expressions may lead to large formulas that can overwhelm backend provers. We describe a technique that transforms set-theoretic formulas by successively applying rewriting and abstraction until a fixed point is reached. The technique is extended to handling functions, records, and tuples, and it is the kernel of the SMT backend of the TLA$^+$ proof system (section 6.3). A paper describing our technique has been presented at the SETS workshop 2015 [46].

Although the approach was mainly intended to support proofs, we have also started work on adapting it for constructing models of formulas in set theory. Being able to construct (counter-)models can help users understand why proof attempts fail. During his internship, Glen Mével from ENS Rennes designed translation rules for a core fragment of TLA$^+$ set theory. He validated them by using the finite model finding functionality of the SMT solver CVC4 for constructing models, with encouraging preliminary results.

7.1.6. Linear Constraints in Integer Arithmetic

We have investigated linear integer constraint solving. Many existing algorithms rely on solving the rational relaxation and transferring the results to an integer branch and bound approach. This algorithm eventually terminates due to the well-known a priori exponential bounds of an integer solution. De Moura and Jovanović proposed the first model-driven terminating algorithm where the termination relies on the structure of the problem itself but not on a priori bounds [62]. However, the algorithm contained some bugs, in particular it did not terminate. We fixed the bugs by introducing the notion of Weak Cooper elimination. Termination requires adding more rules to the algorithm and refining some existing ones [23].

7.1.7. Decidability of First-Order Clause Sets

Recursion is a necessary source for first-order undecidability of clause sets. If there are no cyclic, i.e., recursive definitions of predicates in such a clause set, (ordered) resolution terminates, showing decidability. In this work we present the first characterization of recursive clause sets enabling non-constant function symbols and depth increasing clauses but still preserving decidability. For this class called BDI (Bounded Depth Increase) we present a specialized superposition calculus. This work was published in the Journal of Logic and Computation [63]. Recursive clause sets also become decidable in the context of finite domain axioms. For this case we developped a new calculus that incorporates explicit partial model assumptions guiding the search [19].

7.1.8. Building Blocks for Automated Reasoning

There are automated reasoning building blocks shared between today's prime calculi for propositional logic (CDCL), propositional logic modulo theories (CDCL(T)), and first-order logic with equality (superposition). Underlying all calculi is a partial model assumption guiding inferences that are not redundant. Deciding the abstract redundancy notion is basically as difficult as the overall satisfiability problem for the respective logic, but for well-chosen partial model assumptions inferences can be guaranteed to be non-redundant at much lower cost. For example, for SAT it is possible to computed inferences in linear time [40] that are guaranteed to be non-redundant.

7.1.9. Beagle – A Hierarchic Superposition Prover

Joint work with Peter Baumgartner and Joshua Bax from NICTA, Canberra, Australia.

Hierarchic superposition is a calculus for automated reasoning in first-order logic extended by some background theory. In [20] we describe an implementation of hierarchic superposition within the Beagle theorem prover, and report on Beagle's performance on the TPTP problem library. Currently implemented background theories are linear integer and linear rational arithmetic. Beagle features new simplification rules for theory reasoning and implements calculus improvements like weak abstraction and determining (un)satisfiability w.r.t. quantification over finite integer domains.

7.1.10. Modal Tableau Systems with Blocking and Congruence Closure

Joint work with Renate A. Schmidt from the University of Manchester, UK.

For many common modal and description logics there are ways to avoid the explicit use of equality in a tableau calculus. For more expressive logics, e.g., with nominals as in hybrid modal logics and description logics, avoiding equality becomes harder, though, and for modal logics where the binary relations satisfy frame conditions expressible as first-order formulae with equality, explicit handling of equations is the easiest and sometimes the only known way to perform equality reasoning. In [32] we describe an approach for efficient handling of equality in tableau systems. We combine Smullyan-style tableaux with a congruence closure algorithm, and demonstrate that this method also permits the use of common blocking restrictions such as ancestor blocking.

7.1.11. Subtropical Real Root Finding

This research is motivated by a series of studies of Hopf bifurcations [60], [59] for reaction systems in chemistry and gene regulatory networks in systems biology. The relevant systems are originally given in terms of ordinary differential equations, for which Hopf bifurcations can be described algebraically [54], [74], [58], [57], typically resulting in one very large multivariate polynomial equation $f = 0$ subject to a few much simpler polynomial side conditions $g_1 > 0$, ..., $g_n > 0$. For these algebraic systems one is interested in feasibility over the reals and, in the positive case, in at least one feasible point. It turns out that, generally, scientifically meaningful information can be obtained already by checking only the feasibility of $f = 0$, which is the focus of this project. For further details on the motivating problems, we refer to our earlier publications [72], [71], [56], [55].

With one of our models, viz. *Mitogen-activated protein kinase (MAPK)*, we obtain and solve polynomials of considerable size. Our currently largest instance `mapke5e6` contains 863,438 monomials in 10 variables. One of the variables occurs with degree 12, all other variables occur with degree 5. Such problem sizes are clearly beyond the scope of classical methods in symbolic computation. To give an impression, the size of an input file with `mapke5e6` in infix notation is 30 MB large. LaTeX-formatted printing of `mapke5e6` would fill more than 5000 pages in this report.

We have developed an incomplete but terminating algorithm for finding real roots of large multivariate polynomials [33]. The principal idea is to take an abstract view of the polynomial as the set of its exponent vectors supplemented with sign information on the corresponding coefficients. To that extent, out approach is quite similar to tropical algebraic geometry [73]. However, after our abstraction we do not consider tropical varieties but employ linear programming to determine certain suitable points in the Newton polytope, which somewhat resembles successful approaches to sum-of-square decompositions [67].

We have implemented our approach in Reduce [61] using direct function calls to the dynamic library of the LP solver Gurobi [48]. In practical computations on several hundred examples originating from the work within an interdisciplinary research group our method has failed due to its incompleteness in only 10 percent of the cases. The longest computation time observed was around 16 s for the above-mentioned `mapke5e6`. With a publication of our computational results in a physics journal, our research had considerable impact beyond computer science [17].

7.1.12. Standard Answers for Virtual Substitution

Joint work with A. Dolzmann from Leibniz-Zentrum für Informatik in Saarbrücken, Germany.

We consider existential problems over the reals. Extended quantifier elimination generalizes the concept of regular quantifier elimination by additionally providing answers which are descriptions of possible assignments for the quantified variables. Implementations of extended quantifier elimination via virtual substitution have been successfully applied to various problems in science and engineering.

So far, the answers produced by these implementations included infinitesimal and infinite numbers, which are hard to interpret in practice. This has been explicitly criticized in the scientific literature. In our article [44], we introduce a complete post-processing procedure to convert, for fixed values of parameters, all answers into standard real numbers. We furthermore demonstrate the successful application of an implementation of our method within Redlog to a number of extended quantifier elimination problems from the scientific literature including computational geometry, motion planning, bifurcation analysis for models of genetic circuits and for mass action, and sizing of electrical networks.

7.1.13. A Generalized Framework for Virtual Substitution

We generalize the framework of virtual substitution for real quantifier elimination to arbitrary but bounded degrees [45]. We make explicit the representation of test points in elimination sets using roots of parametric univariate polynomials described by Thom codes. Our approach follows an early suggestion by Weispfenning, which has never been carried out explicitly.

We give necessary and sufficient conditions for the existence of a root with a given test point representation. These conditions are used to rule out redundant test points. Our encoding allows us to distinguish between test points that represent lower bounds and test points representing upper bounds of a satisfying interval for a given input formula. Furthermore, we show how to reduce the size of elimination sets by generalizing a well-known idea from linear virtual substitution, namely to consider only test points representing lower bounds of a satisfying interval.

Our framework relies on some external algorithm $\mathcal{A}$, which is used to eliminate a single existential quantifier from a finite set of generic formulas. The existence of $\mathcal{A}$ is guaranteed by the fact that $\mathbb{R}$ admits quantifier elimination. We briefly refer to experiments which compared the performance of our framework—when Cylindrical Algebraic Decomposition is used as the external algorithm—to other quantifier elimination algorithms. Unfortunately, our approach is not yet able to compete with other state-of-the-art quantifier elimination algorithms. However, currently ongoing research suggests the possibility for drastic improvements in practice. Investigating this is left for future work.

7.2. Formal Methods for Developing Algorithms and Systems

Participants: Manamiary Andriamiarina, Noran Azmy, Gabriel Corona, Marie Duflot-Kremer, Marion Guthmuller, Souad Kherroubi, Dominique Méry, Stephan Merz, Martin Quinson, Christoph Weidenbach.

7.2.1. Incremental Development of Distributed Algorithms

Joint work with Mike Poppleton, University of Southampton, UK, and with Neeraj Kumar Singh from the Department of Computing and Software, McMaster University, Hamilton, Canada.

The development of distributed algorithms and, more generally, of distributed systems, is a complex, delicate, and challenging process. The approach based on refinement helps to gain formality by using a proof assistant, and proposes to apply a design methodology that starts from the most abstract model and leads, in an incremental way, to the most concrete model, for producing a distributed solution. Our work helps formalizing pre-existing algorithms, developing new algorithms, as well as developing models for distributed systems.

More concretely, we aim at an integration of the correct-by-construction refinement-based approach for distributed algorithms. Our main results during 2015 are:

- An integrated formal method for verification of liveness properties in distributed systems is introduced [43], and the verification of a self-stabilizing leader election protocol for population protocols illustrates the proposed methodology.
- Manamiary Andriamiarina completed his PhD, illustrating a method for developing distributed algorithms based on a combination of Event-B and fragment of temporal logic TLA.
- The methodology has been applied to take into account resilience in distributed systems. We describe a fully mechanized proof of correctness of self-☆ systems [42] along with an interesting case study related to P2P-based self-healing protocols.

7.2.2. Modeling Medical Devices

Joint work with Neeraj Kumar Singh from the Department of Computing and Software, McMaster University, Hamilton, Canada.

Formal modeling techniques and tools have attained sufficient maturity for formalizing highly critical systems in view of improving their quality and reliability, and the development of such methods has attracted the interest of industrial partners and academic research institutions. Building high quality and zero-defect medical software-based devices is a particular domain where formal modelling techniques can be applied effectively. Medical devices are very prone to showing unexpected system behaviour in operation when traditional methods are used for system testing. Device-related problems have been responsible for a large number of serious injuries. Officials of the US Food and Drug Administration (FDA) found that many deaths and injuries related to these devices are caused by flaws in product design and engineering. Cardiac pacemakers and implantable cardioverter-defibrillators (ICDs) are among the most critical medical devices and require closed-loop modelling (integrated system and environment modelling) for verification purposes before obtaining a certificate from the certification bodies.

Clinical guidelines systematically assist practitioners in providing appropriate health care in specific clinical circumstances. Today, a significant number of guidelines and protocols are lacking in quality. Indeed, ambiguity and incompleteness are likely anomalies in medical practice. The analysis of guidelines using formal methods is a promising approach for improving them.

Analyzing requirements is a major challenge in the area of safety- critical software, where the quality of requirements is an important issue for building a dependable critical system. Many projects fail due to lack of understanding of user needs, missing functional and non-functional system requirements, inadequate methods and tools, and inconsistent system specifications. This often results from the poor quality of system requirements. Based on our experience and knowledge, an environment model has been recognized to be a promising approach to support the requirements engineering to validate a system specification. It is crucial to get an approval and feedback at an early stage of the system development to guarantee the completeness and correctness of the requirements. In [29], we propose a method for analyzing the system requirements using closed-loop modelling technique. The closed-loop model in an integration of system model and environment model, where both the system and environment models are formalized using formal techniques. Formal verification of this closed-loop model helps to identify hidden or missing system requirements and peculiar behaviours, which are not covered earlier during requirements elicitation process. Moreover, the environment model assists in the construction, clarification, and validation of a given system requirements.

7.2.3. Verification of the Pastry routing protocol

In his PhD thesis at Saarbrücken University in 2013, Tianxiang Lu had studied the routing protocol of the Pastry algorithm [69] for maintaining a distributed hash table in a peer-to-peer network. He had discovered several problems in the published algorithm and proposed a modification of the protocol, together with a correctness proof under the hypothesis that no node ever disconnects. The proof had been checked using TLAPS, but it made many assumptions on the underlying data structures that were left unchecked. In particular, support for (modulus) arithmetic in TLAPS was too weak at the time when the proof was written.

As part of her PhD thesis, Noran Azmy studied the assumptions that had been left unproved, and found that several of them were not valid. As a consequence, she was able to find a counter-example to one of the invariants underlying the correctness proof. She corrected the assumptions, proved all of the ones that were needed for the proof using the current version of TLAPS, and also introduced higher-level abstractions that allowed her to rewrite the specification and the correctness proof of the routing protocol in a way that avoids low-level arithmetic reasoning throughout the proof. As a result, she obtained a complete machine-checked proof of Lu's variant of Pastry, still under the assumption that no node leaves the network. A paper describing the result is being submitted.

7.2.4. Proof of Determinacy of PharOS

Joint work with Selma Azaiez and Matthieu Lemerre (CEA Saclay), and Damien Doligez (Inria Paris).

The main contribution of our team to the ADN4SE project (section 8.1), in cooperation with colleagues from CEA, was to write a high-level specification of the real-time operating system PharOS in the TLA$^+$ language, and to prove a determinacy property of the model using TLAPS. Roughly speaking, determinacy means that the sequence of local states of each process during a computation does not depend on the order in which processes are scheduled, as long as there are no missed deadlines. This property simplifies the analysis and verification of programs that run on PharOS. It relies on the fact that every instruction is associated with a time window of execution, and a message can only be received by an instruction if the earliest possible execution time of that instruction is later than the latest possible execution time of the instruction sending the message. The model and proof are based on Lemerre et al. [65]. However, the underlying assumptions are made fully explicit in the formal model, and the proof is carried out in assertional rather than behavioral style. The proof was completed in 2015, and a paper describing the result is being submitted.

7.2.5. Formal Development of Component Semantics in B

Joint work with David Déharbe of Universidade Federal do Rio Grande de Norte (UFRN), Brazil.

We develop a formal model in Isabelle/HOL of the behavioral semantics of software components designed with the B method. We formalize semantic objects, based on labeled transition systems, notions of internal and externally visible behavior, and simulation. In particular, we study a variant of simulation that corresponds to refinement in the B method. We also formally represent the composition of components in the B method.

This work was presented at an invited talk at FACS 2015 in Rio de Janeiro, and an article will be published in LNCS.

7.2.6. Analysis of Distributed Legacy Applications

SimGrid is a toolkit for the study of Large-Scale Distributed Systems. It contains both a simulator with sound and validated performance models for the network, CPUs, and disks, but also an explicit model checker exploring all possible message interleavings in the application, and searching for states violating some properties specified by the user.

We recently added the ability to assess liveness properties over arbitrary and legacy codes, thanks to a system-level introspection tool that provides a detailed view of the running application to the model checker. This can for example be leveraged to verify both safety and liveness properties, on arbitrary MPI code written in C, C++ or Fortran. This work has been published in the Workshop on Formal Approaches to Parallel and Distributed Systems (4PAD) [26], while the full details appear in Guthmuller's PhD thesis [12].

In his master project, Gabriel Rodrigues Santos investigated the feasibility of implementing algorithms for statistical model checking within SimGrid. The basic idea is to sample sufficiently many executions of a program, based on probabilistic parameters associated with the execution platform, for quantifying correctness and reliability properties. By construction, the answers obtained in this way are not exact, but their imprecision can be bounded by an interval of confidence. The results are very encouraging, and we intend to pursue this approach in further work.

7.2.7. Evaluating and Verifying Probabilistic Systems

Joint work with colleagues at ENS Cachan, University Paris Est Créteil, and Ecole Centrale Paris.

Since its introduction in the 1980s, model checking has become a prominent technique for the verification of complex systems. The aim was to decide whether or not a system fulfills its specification. With the rise of probabilistic systems, new techniques have been designed to verify this new type of systems, and appropriate logics have been proposed to describe more subtle properties to be verified. However, some characteristics of such systems fall outside the scope of model checking. In particular, it is often of interest not to decide wether a property is satisfied but how well the system performs with respect to a certain measure. We have designed a statistical tool for tackling both performance and verification issues. Following several conference talks, two journal papers have been published. The first one [14] presents the approach in details together with illustrative applications to flexible manufacturing systems, and to the study of a biological mechanism knwon as circadian clock. The second one [15] focuses on biological applications, and more precisely the use of statistical model checking to detect and measure several indicators of oscillating biological systems.

CARTE Project-Team

7. New Results

7.1. Computability and Complexity

- **Complexity of stream functions and higher-order complexity.** We have pursued our works on higher-order complexity and the complexity of stream functions. Both notions are closely related as any function from natural numbers to natural numbers can be seen as a stream (an infinite list) of natural numbers:
  - A characterization of the class of Basic Feasible Functionals using term rewrite systems on streams and interpretation methods has been proposed in [13]. This result is part of Hugo Férée's PhD thesis for which he has obtained the Ackermann award.
  - In [14], we have provided some interpretation criteria useful to ensure two kinds of stream properties: space upper bounds and input/output upper bounds. Our space upper bounds criterion ensures global and local upper bounds on the size of each output stream element expressed in term of the maximal size of the input stream elements. The input/output upper bounds criterion considers instead the relations between the number of elements read from the input stream and the number of elements produced on the output stream.
  - The paper [21] has extended the light affine lambda calculus with inductive and coinductive data types using the category theory notions of (weak) initial algebra and coalgebra.
- **Complexity analysis of Object-Oriented programs.** We have proposed a type system based on non-interference and data ramification (tiering) principles in [22]. We have captured the set of functions computable in polynomial time on OO programs. The studied language is general enough to capture most OO constructs and the characterization is quite expressive as it allows the analysis of a combination of imperative loops and of data ramification scheme based on Bellantoni and Cook's safe recursion using function algebra.
- **Rice-like theorem for primitive recursive functions.** We have studied the following question: what are the properties of primitive recursive functions that are decidable (by a Turing machine), given a primitive recursive presentation of the function. We give a complete characterization of these properties. We show that they can be expressed as unions of elementary properties of being compressible. If $h : \mathbb{N} \to \mathbb{N}$ is a computable increasing unbounded function (like $\log(n)$ or 2^n), we say that a function $f : \mathbb{N} \to \mathbb{N}$ is h-compressible if for each n there is a primitive recursive program of size at most $h(n)$ computing a function that coincides with f on entries $0, 1, ..., n$. Whether f is h-compressible is decidable given a primitive recursive program for f, and every decidable property can be obtained as a combination of such elementary properties. This result actually holds for any class of total functions that admits a sound and complete programming language. An article is currently in preparation.
- **Parametrization of geometric figures.** During the master internship of Diego Nava Saucedo, we have studied the semi-computability of geometric figures. A figure is semi-computable if there is a program that semi-decides whether a pixel intersects the figure. Our goal is to understand the semi-computability of a figure in terms of the parameters describing the figure. It turns out that the usual ways of parameterizing simple figures such as triangles, squares or disks do not behave well in terms of semi-computability. We have actually proved that no *finite* parametrization behaves well.
- **Symbolic Dynamics on Groups.** In an effort to better understand the interplay of geometry and computability in tiling theory, E. Jeandel has studied tiling problems on general Cayley graphs, and has obtained a significant number of new results. He has proven that groups with an (strongly) aperiodic tiling system have decidable word problem [30], and provided examples of new groups (in particular monster groups) with such tiling systems, and proved that all nontrivial nilpotent groups

have an aperiodic tiling system and an undecidable domino problem [31]. He also showed how the new concept of translation-like actions from geometric group theory can be used to prove that many groups, in particular the Grigorchuk groups and most groups with a nontrivial center, have an undecidable domino problem [33].

- **The smallest aperiodic tileset.** In a joint work with Michael Rao, E. Jeandel has proven that there exists an aperiocic set of 11 Wang tiles [34], and furthermore that this number is optimal.

7.2. Quantum Computing

- **On Weak Odd Domination and Graph-based Quantum Secret Sharing.** In this work published in the journal Theoretical Computer Science [15], Simon Perdrix and his co-authors Sylvain Gravier, Jérôme Javelle and Mehdi Mhalla study weak odd domination in graphs and its application in quantum secret sharing. A weak odd dominated (WOD) set in a graph is a subset B of vertices for which there exists a distinct set of vertices C such that every vertex in B has an odd number of neighbors in C. They point out the connections of weak odd domination with odd domination, [σ,ρ]-domination, and perfect codes. They introduce bounds on κ(G), the maximum size of WOD sets of a graph G, and on κ´(G), the minimum size of non-WOD sets of G. Moreover, they prove that the corresponding decision problems are NP-complete. The study of weak odd domination is mainly motivated by the design of graph-based quantum secret sharing protocols: a graph G of order n corresponds to a secret sharing protocol whose threshold is κQ(G)=max(κ(G),n-κ´(G)). These graph-based protocols are very promising in terms of physical implementation, however all such graph-based protocols studied in the literature have quasi-unanimity thresholds (i.e. κQ(G)=n-o(n) where n is the order of the graph G underlying the protocol). In this paper, they show using probabilistic methods the existence of graphs with smaller κQ (i.e. κQ(G)$\leq$0.811n where n is the order of G). They also prove that deciding for a given graph G whether κQ(G)$\leq$k is NP-complete, which means that one cannot efficiently double check that a graph randomly generated has actually a κQ smaller than 0.811n.
- **Minimum Degree up to Local Complementation: Bounds, Parameterized Complexity, and Exact Algorithms.** In this work presented at ISAAC [25], David Cattaneo and Simon Perdrix introduce new upper bounds and exact algorithms for the local minimum degree. The author also prove the W[2]-membership of the corresponding decision problem. The local minimum degree of a graph is the minimum degree that can be reached by means of local complementation. For any n, there exist graphs of order n which have a local minimum degree at least $0.189n$, or at least $0.110n$ when restricted to bipartite graphs. Regarding the upper bound, they show that the local minimum degree is at most $3/8n + o(n)$ for general graphs and $n/4 + o(n)$ for bipartite graphs, improving the known $n/2$ upper bound. They also prove that the local minimum degree is smaller than half of the vertex cover number (up to a logarithmic term). The local minimum degree problem is NP-Complete and hard to approximate. They show that this problem, even when restricted to bipartite graphs, is in W[2] and FPT-equivalent to the EvenSet problem, whose W[1]-hardness is a long standing open question. Finally, they show that the local minimum degree is computed by a $O_*(1.938n)$-algorithm, and a $O_*(1.466n)$-algorithm for the bipartite graphs.
- **The ZX Calculus is incomplete for Clifford+T quantum mechanics.** The ZX calculus is a diagrammatic language for quantum mechanics and quantum information processing. In this paper[17], Simon Perdrix and Harny Wang prove that the ZX-calculus is not complete for the Clifford+T quantum mechanics. The completeness for this fragment has been stated as one of the main current open problems in categorical quantum mechanics. The ZX calculus was known to be incomplete for quantum mechanics, on the other hand, it has been proved complete for Clifford quantum mechanics (a.k.a. stabilizer quantum mechanics), and for single-qubit Clifford+T quantum mechanics. The question of the completeness of the ZX calculus for Clifford+T is a crucial step in the development of the ZX calculus because of its (approximate) universality for quantum mechanics (i.e. any unitary evolution can be approximated using Clifford and T gates only). They exhibit a property which is know to be true in Clifford+T quantum mechanics and prove that this equation cannot be derived

in the ZX calculus, by introducing a new sound interpretation of the ZX calculus in which this particular property does not hold. Finally, we propose to extend the language with a new axiom. This result has been presented as invited speakers in the conferences "Quantum Theory: from foundations to technologies" in Vaxjo Sweden, and "Higher TQFT and categorical quantum mechanics" at the Scrounger Institute in Vienna. The authors also presented these results at the workshop of the CNRS groupe de travail Informatique Quantique du GDR IM, in Grenoble.

- **Block Representation of Reversible Causal Graph Dynamics.** In this work presented at the conference on Foundation of computer science (FCT'15) [18], Pablo Arrighi, Simon Martiel and Simon Perdrix, consider a reversible version of the causal graph dynamics. Causal Graph Dynamics extend Cellular Automata to arbitrary, bounded-degree, time-varying graphs. The whole graph evolves in discrete time steps, and this global evolution is required to have a number of physics-like symmetries: shift-invariance (it acts everywhere the same) and causality (information has a bounded speed of propagation). We study a further physics-like symmetry, namely reversibility. More precisely, we show that Reversible Causal Graph Dynamics can be represented as finite-depth circuits of local reversible gates.
- **Reversibility in the Extended Measurement-based Quantum Computation.** In this work by Nidal Hamrit and Simon Perdrix has been presented at the conference on Reversible Computation in Grenoble [23]. When applied on some particular quantum entangled states, measurements are universal for quantum computing. In particular, despite the fondamental probabilistic evolution of quantum measurements, any unitary evolution can be simulated by a measurement-based quantum computer (MBQC). They consider the extended version of the MBQC where each measurement can occur not only in the X,Y-plane of the Bloch sphere but also in the X,Z- and Y,Z-planes. The existence of a gflow in the underlying graph of the computation is a necessary and sufficient condition for a certain kind of determinism. They extend the focused gflow (a gflow in a particular normal form) defined for the X,Y-plane to the extended case, and provide necessary and sufficient conditions for the existence of such normal forms.
- **Quantum Circuits for the Unitary Permutation Problem.** In this paper presented at TAMC'15 [20] Stefano Facchni and Simon Perdrix consider the *Unitary Permutation* problem which consists, given n quantum gates $U_1, ..., U_n$ and a permutation σ of $\{1, ..., n\}$, in applying the quantum gates in the order specified by σ, i.e., in performing $U_{\sigma(n)} \circ ... \circ U_{\sigma(1)}$. This problem has been introduced and investigated in [47] where two models of computations are considered. The first is the (standard) model of query complexity: the complexity measure is the number of calls to any of the quantum gates U_i in a quantum circuit which solves the problem. The second model is roughly speaking a model for higher order quantum computation, where quantum gates can be treated as objects of second order. In both model the existing bounds are improved, in particular the upper and lower bounds for the standard quantum circuit model are established by pointing out connections with the *permutation as substring* problem introduced by Karp.

CASSIS Project-Team

7. New Results

7.1. Automated Deduction

We develop general techniques which allow us to re-use available tools in order to build a new generation of solvers offering a good trade-off between expressiveness, flexibility, and scalability. We focus on the careful integration of combination techniques and rewriting techniques to design decision procedures for a wide range of verification problems.

7.1.1. Building and Verifying decision procedures

Participants: Alain Giorgetti, Olga Kouchnarenko, Christophe Ringeissen.

In the context of the PhD thesis by Elena Tushkanova (defended in 2013), we have developed a methodology to build decision procedures specified by using a superposition calculus [20] which is at the core of all equational theorem provers. This calculus is refutation complete: it provides a semi-decision procedure that halts on unsatisfiable inputs but may diverge on satisfiable ones. Fortunately, it may also terminate for some theories of interest in verification, and thus it becomes a decision procedure. To reason on the superposition calculus, a schematic superposition calculus has been developed to build the schematic form of the saturations allowing to automatically prove decidability of single theories and of their combinations. We have proposed a rule-based logical framework and a tool implementing a complete many-sorted schematic superposition calculus for arbitrary theories. By providing results for unit theories, arbitrary theories, and also for theories with counting operators, we show that this tool is very useful to derive decidability and combinability of theories of practical interest in verification.

7.1.2. Combination of Satisfiability Procedures

Participant: Christophe Ringeissen.

We have continued our work started with Paula Chocron (IIIA-CSIC, U. Barcelona) and Pascal Fontaine (project-team Veridis) on the extension of the Nelson-Oppen combination method to non-disjoint unions of theories. We investigate the case of theories connected via bridging functions [28]. The motivation is, e.g., to solve verification problems expressed in a combination of data structures connected to arithmetic with bridging functions such as the length of lists and the size of trees. We present a sound and complete combination procedure à la Nelson-Oppen for the theory of absolutely free data structures, including lists and trees. This combination procedure is then refined for standard interpretations. The resulting theory has a nice politeness property, enabling combinations with arbitrary decidable theories of elements.

To go beyond the case of absolutely free data structures, we study in [29] the problem of determining the data structure theories for which this combination method is sound and complete. Our completeness proof is based on a rewriting approach where the bridging function is defined as a term rewrite system, and the data structure theory is given by a basic congruence relation. Our contribution is to introduce a class of data structure theories that are combinable with a disjoint target theory via an inductively defined bridging function. This class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between. Hence, our non-disjoint combination method applies to many classical data structure theories admitting a rewrite-based satisfiability procedure.

7.1.3. Unification Modulo Equational Theories

Participant: Christophe Ringeissen.

We investigate a hierarchical combination approach to the unification problem in non-disjoint unions of equational theories. In this approach, the idea is to extend a base theory with some additional axioms given by rewrite rules in such way that the unification algorithm known for the base theory can be reused without loss of completeness. Additional techniques are required to solve a combined problem by reducing it to a problem in the base theory. In [33] we show that the hierarchical combination approach applies successfully to some classes of syntactic theories, such as shallow theories since the required unification algorithms needed for the combination algorithm can always be obtained. We also consider the matching problem in syntactic extensions of a base theory. Due to the more restricted nature of the matching problem, we obtain several improvements over the unification problem.

7.1.4. Enumeration of Planar Proof Terms

Participant: Alain Giorgetti.

By the Curry-Howard isomorphism, simply typed lambda terms correspond to natural deduction proofs in minimal logic. Noam Zeilberger and Alain Giorgetti proved that normal planar lambda terms are in size-preserving bijection with rooted planar maps [21]. Although the formal aspect is not emphasized in the paper, the use of formal representations of both normal planar lambda terms and rooted planar maps, of logic programming and a proof assistant software helped much in more quickly finding the bijection.

7.1.5. Rewriting-based Mathematical Model Transformations

Participants: Walid Belkhir, Alain Giorgetti.

Since 2011 we collaborate with the Department “Temps-Fréquence” of the FEMTO-ST institute (Franche-Comté Electronique Mécanique Thermique et Optique - Sciences et Technologies, CNRS UMR 6174) on the formalization of asymptotic methods (based on two-scale convergence). The goal is to design a software, called *MEMSALab*, for the automatic derivation of multiscale models of arrays of micro- and nanosystems. In this domain a model is a partial differential equation. Multiscale methods approximate it by another partial differential equation which can be numerically simulated in a reasonable time. The challenge consists in taking into account a wide range of different physical features and geometries e.g. thin structures, periodic structures, multiple nested scales etc. In [24], we propose a method called "*by-extension-combination*", in which the asymptotic models are constructed incrementally so that model features can be included step by step. More precisely, a model derivation is formalised as a rewriting strategy, and its extension is formalised as a second-order rewriting strategy. Thus, our method amounts to defining an operation of combination over a class of second-order rewriting strategies. We illustrate the method by an example of an asymptotic model for the stationary heat equation in a Micro-Mirror Array developed for astrophysics.

7.2. Security Protocol Verification

The design of cryptographic protocols is error-prone. Without a careful analysis, subtle flaws may be discovered several years after the publication of a protocol, yielding potential harmful attacks. In this context, formal methods have proved their interest for obtaining good security guarantees. Many analysis techniques have been proposed in the literature [66]. We have edited a book [62] where each chapter presents an important and now standard analysis technique. We develop new techniques for richer primitives, wider classes of protocols and higher security guarantees. In Section 7.4.3 we consider derived testing techniques for verifying protocol implementations.

7.2.1. Design of Voting Protocols

Participants: Véronique Cortier, Stéphane Glondu, Steve Kremer, Peter Rønne.

Voting is a cornerstone of democracy and many voting systems have been proposed so far, from old paper ballot systems to purely electronic voting schemes. Although many works have been dedicated to standard protocols, very few address the challenging class of voting protocols.

One famous e-voting protocol is Helios, an open-source web-based end-to-end verifiable electronic voting system, used e.g., by UCL and the IACR association in real elections. One main advantage of Helios is its verifiability, up-to the ballot box (a dishonest ballot box may add ballots). We have defined a variant of Helios, named Belenios, that prevents from ballot stuffing, even against a dishonest ballot box. Our approach consists in introducing an additional authority that provides credentials that the ballot box can verify but not forge. Belenios[0] has been implemented by Stéphane Glondu (cf Section 6.1.3).

Helios as well as Belenios are not receipt-free, that is, a (malicious) voter can *prove* how they voted to any third party. Building upon a scheme proposed by G. Fuschbauer and David Pointcheval, we have enhanced Belenios with a receipt-free variant, called BeleniosRF. Now, the ballot box can re-randomize any (signed) ballot it receives. This way, a voter can no longer exhibit the randomness they used to build their ballot.

End-to-end verifiable voting schemes typically involves voters handling an encrypted ballot in order to confirm that their vote is accurately included in the tally. While this may be technically valid, from a public acceptance standpoint it may be problematic: many voters may not really understand the purpose of the encrypted ballot and the various checks that they can perform. In [61] we take a different approach and revisit an old idea: to provide each voter with a private tracking number. Votes are posted on a bulletin board in the clear along with their associated tracking number. This is appealing in that it provides voters with a very simple, intuitive way to verify their vote, in the clear. However, there are obvious drawbacks: we must ensure that no two voters are assigned the same tracker and we need to keep the trackers private. We propose a new scheme, called Selene, that addresses both of these problems: we ensure that voters get unique trackers and we close off the coercer's window of opportunity by ensuring that the voters only learn their tracking numbers after votes have been posted. The resulting scheme provides receipt-freeness, and indeed a good level of coercion-resistance while also providing a more immediately understandable form of verifiability. The cryptography is under the bonnet as far as the voter is concerned.

In 2010 Hao, Ryan and Zielinski proposed a simple decentralised e-voting protocol that only requires 2 rounds of communication. Thus, for k elections their protocol needs $2k$ rounds of communication. Observing that the first round of their protocol is aimed to establish the public-keys of the voters, we propose in [60] an extension of the protocol as a non-interactive e-voting scheme in the public-key setting (NIVS) in which the voters, after having published their public-keys, can use the corresponding secret-keys to participate in an arbitrary number of one-round elections. We first construct a NIVS with a standard tally function where the number of votes for each candidate is counted. Further, we present constructions for two alternative types of elections. Specifically in the first type (dead or alive elections) the tally shows if at least one voter cast a vote for the candidate. In the second one (elections by unanimity), the tally shows if all voters cast a vote for the candidate. Our constructions are based on bilinear groups of prime order. As definitional contribution we provide formal computational definitions for privacy and verifiability of NIVSs. We conclude by showing intriguing relations between our results, secure computation, electronic exams and conference management systems.

7.2.2. Analysis of Voting Protocols

Participants: Véronique Cortier, Catalin Dragan, Steve Kremer, Peter Rønne.

Properties. Even a basic property like ballot secrecy is difficult to define formally and several definitions co-exist. We studied all game-based privacy definitions of the literature and discovered that none of them was satisfactory: they were either limited (not fully modeling e-voting protocols), or too strong (incompatible with verifiability), or even flawed for a few of them [25]. Based on our findings, we have proposed a new game-based privacy definition BPRIV, proved that it implies simulation-based privacy and showed that it is realized by the Helios protocol [25].

Proof. Such a proof of privacy for Helios is done by hand and is error-prone. Moreover, there is not a single version of Helios. Instead, many slight variants of Helios may be considered (e.g. early and late weeding, weeding based on the identity or on the ciphertexts, mixnet or homomorphic tally, etc.). Each of these variants would require a new proof. Therefore, we are conducting a proof of Helios and Belenios through the Easycrypt framework. This first fully formal proof will cover most existing variants of Helios and Belenios.

[0] https://belenios.loria.fr

Analysis. Existing automated analysis techniques are inadequate to deal with commonly used cryptographic primitives, such as homomorphic encryption and mix-nets, as well as some fundamental security properties, such as verifiability. In collaboration with Matteo Maffei and Fabienne Eigner (Saarland University) we propose a novel approach based on refinement type systems for the automated analysis of two fundamental properties of e-voting protocols, namely, vote privacy and verifiability. We demonstrate the effectiveness of our approach by developing the first automated analysis of Helios using an off-the-shelf type-checker [32].

A challenging problem in e-voting is to provide guarantees when the voting platform itself is corrupted. Du-Vote [73] is a recently presented remote electronic voting scheme that aims to be malware tolerant, i.e., provide security even in the case where the platform used for voting has been compromised by dedicated malware. For this it uses an additional hardware token, similar to tokens distributed in the context of online banking. Du-Vote aims at providing vote privacy as long as either the vote platform or the vote server is honest. For verifiability, the security guarantees are even higher, as even if the token's software has been changed, and the platform and the server are colluding, attempts to change the election outcome should be detected with high probability. In recent work [41] we provide an extensive security analysis of Du-Vote and show several attacks on both privacy as well as verifiability. We also propose changes to the system that would avoid many of these attacks.

7.2.3. Other Families of Protocols

Participants: Véronique Cortier, Jannik Dreier, Alicia Filipiak, Steve Kremer, Ludovic Robin.

Secure Mobile Applications. There is a growing development of Secure Elements for Mobile Phone and Tablets. These Secure Elements are hosted in the SIM for example and can perform cryptographic operations. This opens the way for a much higher level of security in such environnements. However, how to use these secure elements is still very unclear. How keys will be registered in Secure Elements? Which applications may access to the keys and how is this enforced? Which part of the application should be deployed in a Secure Element? It is of course not possible to host an entire application in a Secure Element for size and performance issues. Alicia Filipiak has started a PhD in March 2015 to propose a model for secure mobile applications that make use of Secure Elements. This is a collaboration with Orange Labs (CIFRE). She has proposed a light and secure paiement application which is compatible with standard paiement systems (EMV). The proof of security is conducted in Tamarin, in order to cope with global states.

Protocols using low-entropy secrets. Many two factor authentication protocols consider an additional authentic, but low bandwidth channel to send a confirmation code. A typical example is to send such a code by SMS to a user's mobile phone. Given that such codes need to be copied by users they are short and therefore vulnerable to offline brute-force attacks. Ludovic Robin has started a PhD thesis in October 2014 and proposed a model to take into account an attacker's capability to run such brute-force attacks. While the problem is reminiscent to guessing attacks in password-based protocols, several subtle differences make this problem more difficult. Ludovic is adapting the decision procedure implemented in *Akiss* in order to decide protocol security in the presence of such an attacker.

Auction protocols. Auctions have a long history, having been recorded as early as 500 B.C.. Nowadays, electronic auctions have been a great success and are increasingly used in various applications. Many cryptographic protocols have been proposed to address the various security requirements of these electronic transactions, in particular to ensure privacy. Jannik Dreier, in collaboration with Pascal Lafourcade from Université d'Auvergne and Jean-Guillaume Dumas from Université Grenoble Alpes, recently performed a detailed analysis [15] of Brandt's auction protocol that computes the winner using homomorphic operations on a distributed ElGamal encryption of the bids. Jannik and his coauthors were able to show that this protocol – when using malleable interactive zero-knowledge proofs – is vulnerable to attacks by dishonest bidders. Such bidders can manipulate the publicly available data in a way that allows the seller to deduce all participants' bids. They developed an efficient parallelized implementation of the protocol and the attack to show its practicality.

7.2.4. Automated Verification of Indistinguishability Properties

Participants: Vincent Cheval, Rémy Chrétien, Véronique Cortier, Antoine Dallon, Jannik Dreier, Steve Kremer.

New emerging classes of protocols such as voting protocols often require to model less classical security properties, such as anonymity properties, strong versions of confidentiality and resistance to offline guessing attacks. Many of these properties can be modelled using the notion of indistinguishability by an adversary, which can be conveniently modeled using process equivalences.

Active case, bounded number of sessions. We previously proposed a procedure for approximating trace equivalence in the case of a bounded number of sessions, i.e., for a replication free fragment of a cryptographic process calculus. The procedure is implemented in the *Akiss* tool. While we proved soundness and correctness for any convergent rewrite system that has the finite variant property, termination of the procedure was still an open question. We have recently shown that the procedure indeed terminates for the class of subterm convergent rewrite systems. We are currently also working on an extension of *Akiss* in order to verify protocols that may use the exclusive or operator. This extensions requires us to reason modulo associativity and commutativity. While proving soundness and completeness of a naive extension of the existing procedure is a rather straightforward, the resulting procedure faces directly non-termination. We therefore adapt the resolution strategy to ensure termination on practical examples. While soundness is preserved we need to prove the completeness of the new resolution strategy.

When considering the equational theory corresponding to the standard primitives, Vincent Cheval has proposed a decision procedure for checking equivalence of set constraints, which yields a procedure for checking trace equivalence [69]. We have extended this decision procedure to the case where the attacker can observe the *time* of executions [27], capturing what is called *timing attacks*. To obtain decidability, we have shown how to reduce to a previous result to decide length trace equivalence, where the attacker no longer has access to execution times but can still compare the length of messages. As an application, we study several protocols that aim for privacy. In particular, we (automatically) detect an existing timing attack against the biometric passport and new timing attacks against the Private Authentication protocol.

Active case, unbounded number of sessions.

We have shown that for some classes of protocols, decidability of trace equivalence can be reduced to equivalence of deterministic pushdown automata [13]. Equivalence of deterministic pushdown automata is decidable [79] and the corresponding decision procedure has been recently implemented by Géraud Senizergues. Based on his tool, we have developed a tool for automatically checking equivalence, for an unbounded number of sessions.

For trace properties such as secrecy and authentication, it has been shown that it is sufficient to consider typically three agents, two honest and one dishonest agents [70]. This result no longer holds for equivalence properties. Antoine Dallon has recently started a PhD thesis on deciding equivalence properties. He has shown that it is sufficient to consider two honest agents and two dishonest agents for equivalence properties, for deterministic processes with standard primitives and without else branches. More generally, he shows how to bound the number of agents for arbitrary constructor theories and for protocols with simple else branches. These hypotheses are tight, and counter-examples are provided for non action-deterministic processes, non constructor theories, or protocols with complex else branches.

When proving security in symbolic settings for an unbounded number of sessions, a typical technique (like in the aforementioned result) consists in abstracting away fresh nonces and keys by a bounded set of constants. While this abstraction is clearly sound in the context of secrecy properties (for protocols without else branches), this is no longer the case for equivalence properties. We have shown how to soundly get rid of nonces in the context of equivalence properties [30]. We show that nonces can be replaced by constants provided that each nonce is associated to two constants (instead of typically one constant for secrecy properties). Our result holds for deterministic (simple) protocols and a large class of primitives that includes e.g. standard primitives, blind signatures, and zero-knowledge proofs.

Of course, our abstraction of nonces may introduce false attacks. To avoid this, it is necessary to consider protocols *with* nonce. We have provide the first decidability result for trace equivalence of security protocols, for an unbounded number of sessions and unlimited fresh nonces [31]. Our class encompasses most symmetric key protocols of the literature, in their tagged variant.

Decomposing equivalence. Unique decomposition has been a subject of interest in process algebra for a long time (for example in BPP or CCS in the 1980s), as it provides a normal form and useful cancellation properties. In recent work [16] Jannik Dreier, together with Cristian Ene and Yassine Lakhnech from Université Grenoble Alpes as well as Pascal Lafourcade from Université d'Auvergne, proved two parallel decomposition results for subsets of the applied π-calculus. They showed that every closed normed (i.e. with a finite shortest complete trace) process P can be decomposed uniquely into prime factors P_i with respect to strong labeled bisimilarity, i.e. such that $P \sim_l P_1|...|P_n$. Moreover, they proved that closed finite processes can be decomposed uniquely with respect to weak labeled bisimilarity. They also investigated whether efficient algorithms that compute the unique decompositions exist, which would be useful for the verification of equivalences. It turned out that the simpler problem of deciding whether a process is in its unique decomposition form is undecidable in general in both cases, due to potentially undecidable equational theories. Moreover, the unique decomposition remains undecidable even given an equational theory with a decidable word problem.

7.2.5. Securely Composing Protocols

Participants: Vincent Cheval, Véronique Cortier, Éric Le Morvan.

Protocols are often built in a modular way. For example, authentication protocols may assume pre-distributed keys or may assume secure channels. However, when an authentication protocol has been proved secure assuming pre-distributed keys, there is absolutely no guarantee that it remains secure when executing a real protocol for distributing the keys. During his PhD thesis, Éric Le Morvan has shown how to securely realize the three main types of channels: secure (unreadable and untappable), confidential (unreadable), and authenticated (untappable) channels [54].

7.3. Model-based Verification

We have investigated extensions of regular model-checking to new classes of rewrite relations on terms. We have studied specification and proof of modular imperative programs, as well as of modal workflows.

7.3.1. Tree Automata with Constraints

Participants: Pierre-Cyrille Héam, Olga Kouchnarenko.

Tree automata with constraints are widely used to tackle data base algorithmic problems, particularly to analyse queries over XML documents. The model of Tree Automata with Global Constraints (TAGED) has been introduced for these purposes. The membership problem for TAGED is known to be NP-complete. The emptiness problem for TAGED is known to be decidable and the best known algorithm in the general case is non elementary. Following our NP-hardness result [74], we are still working in collaboration with Vincent Hugot on the complexity of the emptiness problem.

7.3.2. Random Generation of Finite Automata

Participant: Pierre-Cyrille Héam.

Developing new algorithms and heuristics raises crucial evaluation issues, as improved worst-case complexity upper-bounds do not always transcribe into clear practical gains. A classical way for software performance evaluation is to randomly generate inputs.

In collaboration with Jean-Luc Joly, we investigate the problem of randomly and uniformly generating deterministic pushdown automata [40]. Based on a recursive counting approach, we propose a polynomial time algorithm for this purpose. The influence of the accepting condition on the generated automata is also experimentally studied.

Partially ordered automata are finite automata where simple loops have length one. We have used a Markov chain based approach [75] to randomly - and uniformly - generate deterministic partially ordered automata.

In [39] we address the problem of the uniform random generation of non deterministic automata (NFA) up to isomorphism. We show how to use a Monte-Carlo approach to uniformly sample a NFA. The main result is to show how to use the Metropolis-Hastings Algorithm to uniformly generate NFAs up to isomorphism. Using labeling techniques, we show that in practice it is possible to move into the modified Markov Chain efficiently, allowing the random generation of NFAs up to isomorphism with dozens of states. This general approach is also applied to several interesting subclasses of NFAs (up to isomorphism), such as NFAs having a unique initial states and a bounded output degree. Finally, we prove that for these interesting subclasses of NFAs, moving into the Metropolis Markov chain can be done in polynomial time.

7.3.3. Verification of Linear Temporal Patterns over Finite and Infinite Traces

Participants: Pierre-Cyrille Héam, Olga Kouchnarenko.

In the regular model-checking framework, reachability analysis can be guided by temporal logic properties, for instance to achieve the counter example guided abstraction refinement (CEGAR) objectives. A way to perform this analysis is to translate a temporal logic formula expressed on maximal rewriting words into a "rewrite proposition" – a propositional formula whose atoms are language comparisons, and then to generate semi-decision procedures based on (approximations of) the rewrite proposition. In collaboration with Vincent Hugot, we have investigated suitable semantics for LTL on maximal rewriting words and their influence on the feasibility of a translation. We have expended the work in [76] by providing a general translation scheme giving exact results for a fragment of LTL corresponding mainly to safety formulæ, and approximations for a larger fragment.

7.3.4. Constraint Solving for Verifying Modal Workflow Specifications

Participants: Hadrien Bride, Olga Kouchnarenko.

Workflow Petri nets are well suited for modelling and analysing discrete event systems exhibiting behaviours such as concurrency, conflict, and causal dependency between events. They represent finite or infinite-state processes, and several important verification problems, like reachability or soundness, are known to be decidable. Modal specifications introduced in [77] allow loose or partial specifications in a framework based on process algebras.

Our work in [26] aims at verifying modal specifications of coloured workflows with data assigned to the tokens and modified by transitions. To this end, executions of coloured workflow nets are modelled using constraint systems, and constraint solving is used to verify modal specifications specifying necessary or admissible behaviours. An implementation supporting the proposed approach and promising experimental results on an issue tracking system constitute a practical contribution.

7.4. Model-based Testing

Our research in Model-Based Testing (MBT) aims to extend the coverage of tests. The coverage refers to several artefacts: model, test scenario/property, and code of the program under test. The test generation uses various underlying techniques such as symbolic animation of models [71], or symbolic execution of programs by means of dedicated constraints, SMT solvers, or model-checkers.

7.4.1. Automated Test Generation from Behavioral Models

Participants: Fabrice Bouquet, Frédéric Dadeau, Elizabeta Fourneret, Jean-Marie Gauthier, Julien Lorrain, Alexandre Vernotte.

We have developed an original model-based testing approach that takes a behavioral view (modelled in UML) of the system under test and automatically generates test cases and executable test scripts according to model coverage criteria. We continue to extend this result to SysML specifications for validating embedded systems [35]. We apply this method on smartSurface [34]

We have investigated the use of a model-based testing approach for vulnerability testing in web applications. Our research prototype was able to detect vulnerabilities on already deployed web applications [80].

7.4.2. Scenario-Based Verification and Validation

Participants: Fabrice Bouquet, Frédéric Dadeau, Elizabeta Fourneret.

Test scenarios represent an abstract test case specification that aims at guiding the model animation in order to produce relevant test cases. Contrary to the previous section, this technique is not fully automated since it requires the user to design the scenario, in addition to the model.

We have proposed a dedicated formalism to express test properties. A test property is first translated into a finite state automaton which describes a monitor of its behaviors. We have also proposed dedicated property coverage criteria that can be used either to measure the property coverage of a given test suite, or to generate test cases, exercising nominal or robustness aspects of the property. This process has been fully tool-supported into an integrated software prototype[0]. This process has been designed during the ANR TASCCC project (2009-2012) and was continued during the ANR ASTRID OSEP project (2012-2013). The industrialization of this approach, and its integration within commercial test generation tools has started with the ANR ASTRID Maturation MBT_Sec project (2014-2015) in collaboration with the French DoD [46]. A technology transfer is currently in progress to integrate this technology into the Smartesting CertifyIt test generation environment.

Also, we have experimented the model approach to validate and to design Multi-Agent systems [51], [52].

7.4.3. Mutation-based Testing of Security Protocols

Participants: Frédéric Dadeau, Pierre-Cyrille Héam, Michaël Rusinowitch.

We have proposed a model-based penetration testing approach for security protocols [14]. This technique relies on the use of mutations of an original protocol, proved to be correct, for injecting realistic errors that may occur during the protocol implementation (e.g., re-use of existing keys, partial checking of received messages, incorrect formatting of sent messages, use of exponential/xor encryption, etc.). Mutations that lead to security flaws are used to build test cases, which are defined as a sequence of messages representing the behavior of the intruder. We have applied our technique on protocols designed in HLPSL, and implemented the protocol mutation tool jMuHLPSL that performs the mutations. The mutants are then analyzed by *CL-AtSe*.

7.4.4. Code and Contract-based Test Generation and Static Analysis

Participants: Fabrice Bouquet, Frédéric Dadeau, Alain Giorgetti.

With the CEA we have developed a test generation technique based on C code and formal specifications, to facilitate deductive verification, in a new tool named StaDy [43]. The tool integrates the concolic test generator PathCrawler within the static analysis platform Frama-C. StaDy is able to handle the ANSI C Specification Language (ACSL) of the framework and other Frama-C plug-ins are able to reuse results from the test generator. This tool is designed to be the foundation stone of modular static and dynamic analysis combinations in the Frama-C platform.

For bounded exhaustive unitary testing of functions on structured arrays we have designed and formally verified with Frama-C a library of sequential generators [43], [36]. A structured array is an array satisfying given constraints, such as being sorted or having no duplicate values. A sequential generator of structured arrays can be defined by two C functions: the first one computes an initial array, and the second one steps from one array to the next one according to some total order on the set of arrays. We formally specify with ACSL annotations that the generated arrays satisfy the prescribed structural constraints (soundness property) and that the generation is in increasing lexicographic order (progress property). We refine this specification into two programming and specification patterns: one for generation in lexicographic order and one for generation by filtering the output of another generator. After adding suitable loop invariants we automatically prove the soundness and progress properties of many generators with the Frama-C platform.

[0] A video of the prototype is available at: http://vimeo.com/53210102

7.5. Verification of Collaborative Systems

We investigate security problems occurring in decentralized systems. We develop general techniques to enforce read and update policies for controlling access to XML documents based on recursive DTDs (Document Type Definition). Moreover, we provide a necessary and sufficient condition for undoing safely replicated objects in order to enforce access control policies in an optimistic way. We investigate privacy issues for social networks in order to give more control to users over their personal data.

7.5.1. *Automatic Analysis of Web Services Security*

Participants: Walid Belkhir, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

Automatic composition of web services is a challenging task. Many works have considered simplified automata models that abstract away from the structure of messages exchanged by the services. For the domain of secured services (using e.g., digital signing or timestamping) we have proposed an original approach to automated orchestration of services under security constraints. Given a community of services and a goal service, we reduce the problem of generating a mediator between a client and a service community to a security problem where an intruder should intercept and redirect messages from the service community and a client service till reaching a satisfying state.

In [12] we develop an alternative approach based on *parametrized automata*, a natural extension of finite-state automata over infinite alphabet. In this model the transitions are labeled with constants or variables that can be refreshed in some specified states. We show the applicability of our model to Web services handling data from an infinite domain. We reduce the Web service composition problem to the construction of a simulation of a target service by the asynchronous product of existing services, and prove that this construction is computable. We also show expressive equivalence and succinctness of parametrized automata with respect to Finite Memory Automata in [47] We now work on synthesizing composed services that satisfy required security policies.

7.5.2. *Querying Security Views over XML Data*

Participant: Abdessamad Imine.

To enforce access control over XML data, virtual security views are commonly used in many many applications and commercial database systems. Querying these views raises some serious problems. More precisely, user XPath queries posed on recursive views cannot be rewritten to be evaluated on the underlying XML data. Existing rewriting solutions are based on the non-standard language “Regular XPath” enabling recursion operator. However, query rewriting under Regular XPath can be of exponential size. In [17], we show that query rewriting is always possible for arbitrary security views (recursive or not) by using only the expressive power of the standard XPath. We propose a more expressive language to specify XML access control policies as well as an efficient algorithm to enforce such policies. Finally, we present our system, called SVMAX, that implements our solutions and we show that it scales well through an extensive experimental study based on real-life DTD.

7.5.3. *Secure Computation in Social Networks*

Participants: Younes Abid, Bao Thien Hoang, Abdessamad Imine, Huu Hiep Nguyen, Michaël Rusinowitch.

Online social networks are increasingly exploited as real platforms for creating social links and sharing data. They are used from organizing public opinion polls about any societal theme to publish social graph data for achieving in-depth studies. To securely perform these large-scale computations, we need the design of reliable protocols to ensure the data privacy. In [44], [9], we address the polling problem in social networks where users want to preserve the confidentiality of their votes, obtain the correct final result, and hide, if any, their misbehaviors. We present EPol, a simple decentralized polling protocol that is deployed on a family of social graphs that satisfy a property based on topological ordering. Using these graphs, we show that their structures enable low communication cost, ensure vote privacy and limit the impact of dishonest users on the accuracy of the polling output.

The problem of private publication of social graphs has attracted a lot of attention recently. In [50], we tackle the problem about the upper bounds of privacy budgets related to differentially private release of graphs. We provide such a bound and we prove that, with a privacy budget of $O(\log n)$, there exists an algorithm capable of releasing a noisy output graph with edge edit distance of $O(1)$ against the true graph. At the same time, the complexity of our algorithm $Top-m$ Filter is linear in the number of edges m. This lifts the limits of the state-of-the-art, which incur a complexity of $O(n^2)$ where n is the number of nodes and runnable only on small graphs.

Anonymous use of Social network do not prevent users from privacy risks resulting from infering and cross-checking information published by themselves or their relationhips. In [57], we have conducted a survey in order to measure sensitiveness of personal data published on social media and to analyze the users behaviors. We have shown that 76% of internet users that have answered the survey are vulnerable to identity or sensitive data disclosure.

7.5.4. Safe Protocols for Collaborative Applications

Participant: Abdessamad Imine.

The Operational Transformation (OT) approach, used in many collaborative editors, allows a group of users to concurrently update replicas of a shared object and exchange their updates in any order. The basic idea is to transform any received update operation before its execution on a replica of the object. Designing transformation functions for achieving convergence of object replicas is a critical and challenging issue. In this work, we investigate the existence of transformation functions [19]. From the theoretical point of view, two properties, named TP1 and TP2, are necessary and sufficient to ensure convergence. Using controller synthesis technique, we show that there are some transformation functions, which satisfy TP1 for the basic signatures of insert and delete operations. But, there is no transformation function, which satisfies both TP1 and TP2. Consequently, a transformation function which satisfies both TP1 and TP2 must necessarily have additional parameters in the signatures of some update operations. Accordingly, we provide a new transformation function and show formally that it ensures convergence.

COMETE Project-Team

7. New Results

7.1. Foundations of information hiding

Information hiding refers to the problem of protecting private information while performing certain tasks or interactions, and trying to avoid that an adversary can infer such information. This is one of the main areas of research in Comète; we are exploring several topics, described below.

7.1.1. On the information leakage of differentially-private mechanisms

Differential privacy aims at protecting the privacy of participants in statistical databases. Roughly, a mechanism satisfies differential privacy if the presence or value of a single individual in the database does not significantly change the likelihood of obtaining a certain answer to any statistical query posed by a data analyst. Differentially-private mechanisms are often oblivious: first the query is processed on the database to produce a true answer, and then this answer is adequately randomized before being reported to the data analyst. Ideally, a mechanism should minimize leakage, i.e., obfuscate as much as possible the link between reported answers and individuals' data, while maximizing utility, i.e., report answers as similar as possible to the true ones. These two goals, however, are in conflict with each other, thus imposing a trade-off between privacy and utility.

In [12] we used quantitative information flow principles to analyze leakage and utility in oblivious differentially-private mechanisms. We introduced a technique that exploits graph symmetries of the adjacency relation on databases to derive bounds on the min-entropy leakage of the mechanism. We considered a notion of utility based on identity gain functions, which is closely related to min-entropy leakage, and we derived bounds for it. Finally, given some graph symmetries, we provided a mechanism that maximizes utility while preserving the required level of differential privacy.

7.1.2. Geo-indistinguishability: A Principled Approach to Location Privacy

With the increasing popularity of handheld devices, location-based applications and services have access to accurate and real-time location information, raising serious privacy concerns for their users. In [17] we reported on our ongoing project aimed at protecting the privacy of the user when dealing with location-based services. The starting point of our approach is the principle of geo-indistinguishability, a formal notion of privacy that protects the user's exact location, while allowing approximate information – typically needed to obtain a certain desired service – to be released. We then presented two mechanisms for achieving geo-indistinguishability, one generic to sanitize locations in any setting with reasonable utility, the other custom-built for a limited set of locations but providing optimal utility. Finally we extended our mechanisms to the case of location traces, where the user releases his location repeatedly along the day and we provide a method to limit the degradation of the privacy guarantees due to the correlation between the points. All the mechanisms were tested on real datasets and compared both among themselves and with respect to the state of the art in the field.

7.1.3. Constructing elastic distinguishability metrics for location privacy

The recently introduced notion of geo-indistinguishability tries to address the problem of accessing location-aware services in a privacy-friendly way by adapting the well-known concept of differential privacy to the area of location-based systems. Although geo-indistinguishability presents various appealing aspects, it has the problem of treating space in a uniform way, imposing the addition of the same amount of noise everywhere on the map.

In [13] we proposed a novel elastic distinguishability metric that warps the geometrical distance, capturing the different degrees of density of each area. As a consequence, the obtained mechanism adapts the level of noise while achieving the same degree of privacy everywhere. We also showed how such an elastic metric can easily incorporate the concept of a "geographic fence" that is commonly employed to protect the highly recurrent locations of a user, such as his home or work. We performed an extensive evaluation of our technique by building an elastic metric for Paris' wide metropolitan area, using semantic information from the OpenStreetMap database. We compared the resulting mechanism against the Planar Laplace mechanism satisfying standard geo-indistinguishability, using two real-world datasets from the Gowalla and Brightkite location-based social networks. The results showed that the elastic mechanism adapts well to the semantics of each area, adjusting the noise as we move outside the city center, hence offering better overall privacy.

7.1.4. Quantitative Information Flow for Scheduler-Dependent Systems

Quantitative information flow analyses measure how much information on secrets is leaked by publicly observable outputs. One area of interest is to quantify and estimate the information leakage of composed systems. Prior work has focused on running disjoint component systems in parallel and reasoning about the leakage compositionally, but has not explored how the component systems are run in parallel or how the leakage of composed systems can be minimised.

In [23] we considered the manner in which parallel systems can be combined or scheduled. This considers the effects of scheduling channels where resources may be shared, or whether the outputs may be incrementally observed. We also generalised the attacker's capability, of observing outputs of the system, to consider attackers who may be imperfect in their observations, e.g. when outputs may be confused with one another, or when assessing the time taken for an output to appear. Our main contribution was to present how scheduling and observation affect information leakage properties. In particular, that scheduling can hide some leaked information from perfect observers, while some scheduling may reveal secret information that is hidden to imperfect observers. In addition we presented an algorithm to construct a scheduler that minimises the min-entropy leakage and min-capacity in the presence of any observer.

7.2. Foundations of Concurrency

Distributed systems have changed substantially in the recent past with the advent of phenomena like social networks and cloud computing. In the previous incarnation of distributed computing the emphasis was on consistency, fault tolerance, resource management and related topics; these were all characterized by *interaction between processes*. Research proceeded along two lines: the algorithmic side which dominated the Principles Of Distributed Computing conferences and the more process algebraic approach epitomized by CONCUR where the emphasis was on developing compositional reasoning principles. What marks the new era of distributed systems is an emphasis on managing access to information to a much greater degree than before.

7.2.1. An Algebraic View of Space/Belief and Extrusion/Utterance for Concurrency/Epistemic Logic

The notion of constraint system (cs) is central to declarative formalisms from concurrency theory such as process calculi for concurrent constraint programming (ccp). Constraint systems are often represented as lattices: their elements, called constraints, represent partial information and their order corresponds to entailment. Recently a notion of n-agent spatial cs was introduced to represent information in concurrent constraint programs for spatially distributed multi-agent systems. From a computational point of view a spatial constraint system can be used to specify partial information holding in a given agent's space (local information). From an epistemic point of view a spatial cs can be used to specify information that a given agent considers true (beliefs). Spatial constraint systems, however, do not provide a mechanism for specifying the mobility of information/processes from one space to another. Information mobility is a fundamental aspect of concurrent systems.

In the poster paper [24] we discussed using constraint systems with an algebraic operator that correspond to moving information in-between spaces as to mimic the mobility of data of distributed systems such as posting opinions/lies to other spaces or publicly disclosing data. In the conference paper [22] we enriched spatial constraint systems with operators to specify information and processes moving from a space to another. We referred to these news structures as spatial constraint systems with extrusion. We investigated the properties of this new family of constraint systems and illustrated their applications. From a computational point of view the new operators provide for process/information extrusion, a central concept in formalisms for mobile communication. From an epistemic point of view extrusion corresponds to a notion we called utterance; a piece of information that an agent communicates to others but that may be inconsistent with the agent's beliefs. Utterances can then be used to express instances of epistemic notions, which are commonplace in social media, such as hoaxes or intentional lies. Spatial constraint systems with extrusion can be seen as complete Heyting algebras equipped with maps to account for spatial and epistemic specifications. In the journal paper [28] we extended our work in [22] by showing that spatial constraint systems can also express the epistemic notion of knowledge by means of a derived spatial operator that specifies global information.

7.2.2. A Labelled Semantics for Soft Concurrent Constraint Programming

In [21] we presented a labelled semantics for Soft Concurrent Constraint Programming (SCCP), a language where concurrent agents may synchronize on a shared store by either posting or checking the satisfaction of (soft) constraints. SCCP generalizes the classical formalism by parametrising the constraint system over an order-enriched monoid: the monoid operator is not required to be idempotent, thus adding the same information several times may change the store. The novel operational rules are shown to offer a sound and complete co-inductive technique to prove the original equivalence over the unlabelled semantics.

7.2.3. Verification methods for concurrent Constraint Programming

Concurrent Constraint Programming (CCP) is a well-established declarative framework from concurrency theory. Its foundations and principles e.g., semantics, proof systems, axiomatizations, have been thoroughly studied for over the last two decades. In contrast, the development of algorithms and automatic verification procedures for CCP have hitherto been far too little considered.

To the best of our knowledge there is only one existing verification algorithm for the standard notion of CCP program (observational) equivalence. In [16] we first showed that this verification algorithm has anexponential-time complexity even for programs from a representative sub-language of CCP; the summation-free fragment (CCP+). We then significantly improved on the complexity of this algorithm by providing two alternative polynomial-time decision procedures for CCP+ program equivalence. Each of these two procedures has an advantage over the other. One has a better time complexity. The other can be easily adapted for the full language of CCP to produce significant state space reductions. The relevance of both procedures derives from the importance of CCP+. This fragment, which has been the subject of many theoretical studies, has strong ties to first-order logic and an elegant denotational semantics, and it can be used to model real-world situations. Its most distinctive feature is that of confluence, a property we exploit to obtain our polynomial procedures.

Bisimilarity is a standard behavioral equivalence in concurrency theory. However, only recently a well-behaved notion of bisimilarity for CCP, and a CCP partition refinement algorithm for deciding the strong version of this equivalence have been proposed. Weak bisimilarity is a central behavioral equivalence in process calculi and it is obtained from the strong case by taking into account only the actions that are observable in the system. Typically, the standard partition refinement can also be used for deciding weak bisimilarity simply by using Milner's reduction from weak to strong bisimilarity; a technique referred to as saturation. In [15] we demonstrated that, because of its involved labeled transitions, the above-mentioned saturation technique does not work for CCP. We also gave an alternative reduction from weak CCP bisimilarity to the strong one that allows us to use the CCP partition refinement algorithm for deciding this equivalence. We also proved that due to distinctive nature of CCP, the new method does not introduce infinitely-branching in the resulting transition systems. Finally, we derived an algorithm to automatically verify the notion of weak bisimilarity in CCP.

The ntcc calculus extends CCP with the notion of discrete time-units for the specification of reactive systems. Moreover, ntcc features constructors for non-deterministic choices and asynchronous behavior, thus allowing for (1) synchronization of processes via constraint entailment during a time-unit and (2) synchronization of processes along time-intervals. In [20] we developed the techniques needed for the automatic verification of ntcc programs based on symbolic model checking. We showed that the internal transition relation, modeling the behavior of processes during a time-unit (1 above), could be symbolically represented by formulas in a suitable fragment of linear time temporal logic. Moreover, by using standard techniques as difference decision diagrams, we provided a compact representation of these constraints. Then, relying on a fixpoint characterization of the timed constructs, we obtained a symbolic model of the observable transition (2 above). We proved that our construction is correct with respect to the operational semantics. Finally, we introduced a prototypical tool implementing our method.

DECENTRALISE Team

6. New Results

6.1. Asyncronous Messaging

There are now a variety of end-to-end encrypted messaging platforms targeted at personal correspondences. Amongst these, only Pond and Ricochet provide meaningful resistance to traffic analysis by explicitly protecting the message metadata, although several can optionally operate over Tor to protect the user's location. Ricochet's design around Tor hidden services does not permit offline operation. Pond depends upon a centralized server.

In addition, there are messengers designed for academic research, like Vuvuzela, Dissent, and DP5. These employ information theoretically secure channels like dining cryptographers networks (DC-nets) and private information retrieval schemes (PIR) because they admit extremely simply proofs of security. As DC-nets and PIR schemes scale quadratically, these messaging research projects are effectively limited to a fixed maximum number of users, so they cannot realistically provide an alternative to modern email.

Instead, we have begun exploring the prospects of using mid-latency store-and-forward mixnets for asynchronous messaging. In fact, these are the amongst oldest anonymity systems, dating back to David Chaum, but they were normally restricted to anonymous email projects. At present, we remain in the early design phase, but our design scales linearly while providing many interesting properties desired by modern messengers.

We obtain provable security by basing our system on the Sphinx mixnet packet format, which is provably secure in the universal composability framework [7]. At first blush, Sphinx appears to be an overly restrictive format, but the restrictions are worth obtaining this degree of provable security along with a mixnet's scalability. After consideration, we have devised methods for adding entropy, and optimizing the location of entropy in Sphinx packet headers, without the need to use a larger and slower elliptic curve.

In Sphinx, there is a facility for single-use reply blocks (SURBs), as in other mixnets initially designed for anonymous remailers whose forward and backward messages look alike. We can store an SURB in the packet header, which enters use when the packet passes a fixed cross-over node, thereby allowing both sender and receiver remain anonymous to one another. We can orchestrate the usage of SURBs, and an authentication scheme using tokens, to provide optimal messaging propoerties that:

- Protect the identities of senders and recipients from each other and mixnet nodes, including the mailbox servers,
- Protect the identities of recipient's mailbox servers from even their contact to prevent denial of services attack,
- All redudancy through the usage of multiple mailbox servers.

We shall employ the Axolotl ratchet for long-term forward secrecy in messages, like Pond and Signal do. We can slightly improve upon the Axolotl ratchet by judiciously introducing side key material into the ratchet state. These side keys could be symmetric keys that take a different route through the mixnet, or travel outside the mixnet, thereby allowing the ratchet state to evolve based upon multiple concurrent paths. Side keys could also employ post-quantum public key cryptography, thus providing forward-secrecy against future attackers equipped with quantum computers.

We have also found another forward-secure ratchet inspired by Axolotl that integrates well with the Sphinx packet format. We believe this allows mixnet messages to be protected by long-term ratchets and posses a modicum of protection even against attackers with quantum-computers. At best, long-term ratchets themselves are only pseudonymous, not actually anonymous, so using the integrated ratchets requires considerable care.

6.2. Efficient Privacy-Preserving Scalar Product

We have designed, implemented and evaluated two variants of new privacy-preserving scalar product protocols. The first variant is based on an original idea of Ioannidis et al. [8] and was refined by Amirbekyan et al. [6]. Our first design improves on this by supporting signed values. A second design uses discrete logarithms over Elliptic curves instead of a homomorphic cipher, resulting in a substantially more efficient computation as long as the final result is numerically small.

In both protocols, Alice learns the scalar product $\sum a_i b_i$ of her private vector $\overrightarrow{a}$ with Bob's private vector $\overrightarrow{b}$. The protocol is privacy-preserving in that Alice cannot discern details about $\overrightarrow{b}$ other than what she can learn from $\overrightarrow{a}$ and the scalar product $\sum a_i b_i$, and Bob does not learn anything.

Table 1 summarizes our experimental results.

Table 1. Preliminary performance data for the SP algorithms, wall-clock time running on a single-core of an i7.

Length	RSA-2048	ECC-2^{20}	ECC-2^{28}
25	14 s	2 s	29 s
50	21 s	2 s	29 s
100	39 s	2 s	29 s
200	77 s	3 s	30 s
400	149 s	OOR	31 s
800	304 s	OOR	33 s
800	3846 kb	OOR	70 kb

6.3. GNS support for Tor

We have worked with the Tor community to understand how best to support integration of the GNU Name System with Tor via specialized Tor exit nodes. There are two components to this work:

At present, there are somewhat fragile configuration options to Tor that should allow Tor users to locate the specialized exit nodes, although a small patch to Tor itself would improve upon these.

There are security reasons why Tor should not interact with locally configured name resolution services. OnioNS created a method to make Tor use local services for some domain name lookups, but doing so is somewhat heavy [9]. If we're creating a GNS patch to Tor anyways, then we'll likely extend it to optimize this process.

DICE Team

6. New Results

6.1. The economy of intermediation and the anthropocene

Better understanding the economy, in a broad sense, of intermediation as it is performed by online platforms, is one of the major goals of the team. The paper [12] published in 1024, introduces the topics of algorihtmic intermediation and its social impact to a large audience.

Two contemporary revolutions are shaking the world. On one side, the digital revolution, which seems to introduce to a new economic era, allowing more sharing, and according to some authors the end of capitalism. On the other hand, the challenges of the preservation of our planet, and the limitation of resources that we are now facing. Clearly, there is an expectation that digital means will help face the challenges of the planet. In [14], we go one step further and analyse the possible relationship between the two phenomena, by drawing comparisons with biology where stress on ressources can lead to a horizontalisation of the species, much like what happens with digital technologies and intermediation platforms.

This later work is made in the framework of the study of the anthropocene, for which we are involved in the organisation of a workshop in the framework program of the HKW in Berlin on the technosphere

- URL: http://www.hkw.de/en/programm/projekte/2015/curriculum_campus_technosphere/campus_the_technosphere_issue.php

6.2. Geopolitics of intermediation platforms

Our study of the geopolitics of intermediation aims at grasping the balance of power between platforms, as well as between states - in their relation to platforms - and between platforms and states. We have designed coarse metrics [1] which capture the importance of a platform and the importance of a country in the digital landscape.

Our study focuses on the top 25 websites in a hundred countries. We emphasize the weight of intermedations platforms on the web. We also underline the imbalance between two digital powers - the United States and China - and the rest of the world. Indeed, most platforms belong to these two countries. We have extented our study to a deep analysis of the Asian case [8]. We develop our analysis in an interdisciplinary context as we collaborate with cartographers and economists. Two outcomes of our work are especially notable:

- We produce a set of maps and data vizualisations to illustrate the intermediation economy [11].
- We highlight the determinants of the imbalance in the intermediation landscape. National policies and incentives are of primer importance. The digita landscapes of Korea and Taiwan for instance, show that countries can still play a main role in their domestic web [8].

6.3. Public administration and intermediation platforms

Building on the sucess of platforms such as Uber and the analyses of their externalities, we study the potential role of platforms in public administration. Indeed, cities such as Boston exhibit the interest of a collaboration between administrations and platforms in city planning and maintenance. We also address the role of platforms at a wider level as we study cases such as the settlement of the right to be forgotten in Europe. Our work benefits from the collaboration with administrations, such as Lyon metropole and social scientists. In particular, we have designed three possible scenarii of collobation between platforms and institutions:

- Coexistence: platforms and institutions ignore themselves;
- Conflict: the services developped by platforms conflict with existing policies and institutional practices;
- Partenership: platforms and institutions partner around the development and promotion of services.

A working group has been established on digital sovereignty with CLTC, Center for Long Term Cybersecurity at UC Berkeley, Chaire Castex at Ecole Militaire, and Dice. This working group aims at getting a better understanding of the concept as well as the discrepancy of perception on both sides of the Atlantic. A first seminar was organised in Les Houches in december 2015.

This is work in progress with both academic and public administration actors.

6.4. Architecture design for intermediation platforms

Dice team designs software architectures for intermediation platforms. C3PO and BitBallot targets spontaneous and ephemeral social networks whereas Jumplyn focuses on pure central based system. All these architectures share a common JavaScript layout both at the client and the server sides. In the research context we validate state-of-the art technologies promoted by web leaders such as Google AngularJS, Facebook ReactJS and many others such as Netflix, Wallmart, or the Linux foundation for node.js. The Web environment raises many big issues since all equipments are basically connected to the Internet and the balance between end-user equipment cost and processing power is still a work in progress. Our main research track in such context is to find proper software toolkits hiding Web complexity. We mainly focus on time jitter, cornerstone of Web development, since it implies both end-user and network TCP indecisions. Due to this jitter combination the Web programming model has mutated toward the promises paradigm. It is a complex event based development model provided without external API help. It handles future execution whether successful or not, in a time jittered context. AngularJS, ReactJS, CoffeeScript, NodeJS, MongoDB, ElasticSearch are all time jitter compliant technologies designed for the Web constrains and revolutionising the way we build intermediation platforms.

C3PO explores network transport laziness with the use of a DTN that imposes a larger jitter than classical TCP/UDP. We build a JavaScript mockup [5] that uses a Java based DTN that stores, carries and forwards message from source to destination. C3PO is a software framework extending AngularJS through plugins, without central server, even during deployment phases. We use the dynamic nature of JavaScript to build application on the fly from network messages containing the application description. Our C3PO architecture enables us to build ephemeral and spontaneous social network, on demand and in a matter of days.

Our joint work with Worldline explores the promises paradigm model to enable automation extraction of independent micro-service. These micro-services called *fluxion* [9], from the contraction of flow and functions, may be dynamically and transparently moved over a cluster of servers. Our novelty resides in the fact that the original code is not redesigned for the cluster architecture. *Fluxion* are extracted from the initial code, and an equivalence is maintained between the initially *promissified* code and the *fluxionized* one. Code has two facets, a promise one, used to express software services and a *fluxion* one, used to express software bottlenecks.

Eventually our work with Jumplyn explores complex centralised social network. We want to design a software system to later support our technical research hot topics. The target theme is a software platform that helps students handle their projects. University depends more and more on external resources to teach students. Theses resources are both known by students and their teachers, and the pace and range of explored technologies leads to difficulties in teaching state-of-the-art subjects. The more dedicated a professor needs to be in his research activity, the more broad knowledge he has to teach. For instance 20 years ago one could cope software development teaching with one or two programming languages. Nowadays, a single code involves more then four programming languages to be fully understood. This technology spreading issue stands still in many teaching domains, since past technologies are still actives and future one are promising. We build Jumplyn to cope with this unbalanced game. To help student improving their project and avoid working with obsolete technologies, and to help teacher face the universal and inexpensive availability of knowledge. Jumplyn is a complex JavaScript development stack that collects resources for improving student work and providing services to help them from day to day activities. The current stack integrates the following technologies : MaterialDesign, AngularJS, CoffeeScript, NodeJs, MongoDb, ElasticSearch. Managing and developing software service above this stack is a complex research issue for a small sized development team. We do not have any publication on Jumplyn since our first goal is to build a support intermediation platform to study classical issues such as recommendation or web crawling, scraping and indexation with our own sources of raw data.

PRIVATICS Project-Team

6. New Results

6.1. Surveillance

Participants: Claude Castelluccia, Javier Parra Arnau.

In recent times, we are witnessing an increasing concern by governments and intelligence agencies to deploy mass-surveillance systems that help them fight terrorism. In [40], we conduct a formal analysis of the overall cost of such surveillance systems. Our analysis starts with a fairly-known result in statistics, namely, the false-positive paradox. We propose a quantitative measure of the total cost of a monitoring program, and study a detection system that is designed to minimize it, subject to a constraint in the number of terrorists the agency wishes to capture. In the absence of real, accurate behavioral models, we perform our analysis on the basis of several simple but insightful examples. With these examples, we illustrate the different parameters involved in the design of the detection system, and provide some indicative and representative figures of the cost of the monitoring program.

6.2. Security or privacy ?

Participants: Amrit Kumar, Cédric Lauradoux.

Security softwares such as anti-viruses, IDS or filters help Internet users to protect their privacy from hackers. The cost of this protection is that the users privacy is stripped away by the companies providing these security solutions. Currently, Internet users can choose between no security with the risk of being hacked or use security softwares and lose all personal data to security companies. As a example of this dilemma, we analyze the solution proposed by Google for Safe Browsing in [29] and shows that their privacy policies do not match the reality: Google can perform tracking.

6.3. Users characterization

Participants: Jagdish Achara, Gergely Acs, Claude Castelluccia.

Prior works have shown that the list of apps installed by a user reveal a lot about user interests and behavior. These works rely on the semantics of the installed apps and show that various user traits could be learnt automatically using off-the-shelf machine-learning techniques. In this work, we focus on the re-identifiability issue and thoroughly study the unicity of smartphone apps on a dataset containing 54,893 Android users collected over a period of 7 months. Our study finds that any 4 apps installed by a user are enough (more than 95% times) for the re-identification of the user in our dataset. As the complete list of installed apps is unique for 99% of the users in our dataset, it can be easily used to track/profile the users by a service such as Twitter that has access to the whole list of installed apps of users. As our analyzed dataset is small as compared to the total population of Android users, we also study how unicity would vary with larger datasets. This work emphasizes the need of better privacy guards against collection, use and release of the list of installed apps.

6.4. Data anonymization

Participants: Claude Castelluccia, Gergely Acs.

Set-valued dataset contains different types of items/values per individual, for example, visited locations, purchased goods, watched movies, or search queries. As it is relatively easy to re-identify individuals in such datasets, their release poses significant privacy threats. Hence, organizations aiming to share such datasets must adhere to personal data regulations. In order to get rid of these regulations and also to benefit from sharing, these datasets should be anonymized before their release. In this paper, we revisit the problem of anonymizing set-valued data. We argue that anonymization techniques targeting traditional k^m-anonymity model, which limits the adversarial background knowledge to at most m items per individual, are impractical for large real-world datasets. Hence, we propose in [25] a probabilistic relaxation of k^m-anonymity and present an anonymization technique to achieve it. This relaxation also improves the utility of the anonymized data. We also demonstrate the effectiveness of our scalable anonymization technique on a real-world location dataset consisting of more than 4 million subscribers of a large European telecom operator. We believe that our technique can be very appealing for practitioners willing to share such large datasets.

6.5. Wi-Fi and privacy

Participants: Jagdish Achara, Mathieu Cunche, Vincent Roca, Celestin Matte.

- **Geolocation spoofing attack** Our work at WiSec 2015 [17] shows how it is possible to manipulate the geolocation information of a single device and how to exploit this information as a side channel to identify the owner of the device on geottaged platforms such as social networks.
- **Extraction of sementical information from Wi-Fi network identifiers** Methods based on text similarity metrics can be used to infer user's interests based on the list of their preferred networks. We present in [23] a method identifying the physical entity (shop, restaurant, company ...) associated to Wi-Fi networks identifiers (SSID).

6.6. Formal and legal issues of privacy

Participants: Thibaud Antignac, Daniel Le Metayer.

- **Privacy by design** Privacy by design will become a legal obligation in the European Community when the Data Protection Regulation eventually gets adopted. However, taking into account privacy requirements in the design of a system is a challenging task. We have proposed an approach based on the specification of privacy architectures and illustrated our formal framework through several case studies. In collaboration with Morpho, we have applied it in the context of biometrics systems. The choice of particular techniques and the role of the components (central server, secure module, terminal, smart card, etc.) in the architecture have a strong impact on the privacy guarantees provided by a biometric system. However, existing proposals were made on a case by case basis, which makes it difficult to compare them and to provide a rationale for the choice of specific options. We have shown that a general framework for the definition of privacy architectures can be used to specify these options and to reason about them in a formal way. In 2015 the results on biometrics were presented at the conferences FM2015 [16] and ISC 2015 [15] (best paper award) and the general approach itself has led to Thibaud Antignac's PhD defense.
- **Verification of privacy properties**

  Electric vehicles are an up-and-coming technology that provides significant environmental benefits. A major challenge of these vehicles is their somewhat limited range, requiring the deployment of many charging stations. To effectively deliver electricity to vehicles and guarantee payment, a protocol was developed as part of the ISO 15118 standardization effort. A privacy-preserving variant of this protocol, POPCORN, has been proposed in recent work, claiming to provide significant privacy for the user, while maintaining functionality. We have proposed an approach for the verification of privacy properties of the protocol. We have provided a formal model of the expected privacy properties in the applied Pi-Calculus and used ProVerif to check them. We have identified weaknesses in the protocol in [11] and suggest improvements to address them.
- **Control over personal data**

More than ever the notion of control plays a pivotal and pervasive role in the discourses of privacy and data protection. Privacy scholarship and regulators propose to increase individual control over personal information as the ultimate prescriptive solution to tackle the issues raised by emergent data processing technologies. Conceived as the claim of individuals to determine for themselves when, how, and to what extent information about them is communicated to others, the notion of control is not new. It is often considered as the unique means of empowerment of the data subject. The mechanisms of this empowerment remain however surprisingly vague and understudied. What does it really mean to be in control of one's data in the context of contemporary socio-technical environments and practices? What are the characteristics, purposes and potential limits of such control and how can we guarantee data subjects effective control over their own data? We have carried out an interdisciplinary review of the concept of control to explore such questions in the fields of law and computer science and suggested conditions for the effective application of this concept (see [5]).

- **Accountability** The use of body-worn cameras by police forces around the world is spreading quickly. The resulting mobile and ubiquitous surveillance is often marketed as an instrument for accountability and an effective way of reducing violence. It also involves remarkable potential for intrusion into the privacy of both individuals and police agents. We have studied in [4] the deployment of police body-worn cameras in five countries, investigated their suitability as an accountability tool given the associated privacy threats, and analyzed the societal impact of their deployment as well as the risk of function creep.

6.7. Buidling blocks

Participant: Marine Minier.

- **Symmetric cryptography** During this year, a fruitful work in collaboration with Céline Blondeau from University of AAlto has appeared in FSE 2015 [8] concerning the equivalence between the key recovery parts of the three attacks (Zero-Correlation, impossible and integral) using the matrix method.

  With Thierry Berger, Julien Francq and also Gaël Thomas, we have proposed 2 new lightweight block ciphers : Lilliput and CubeCipher.

  Concerning symmetric cryptography, we obtain some results in both sides: on the one hand, we provide 2 new families of lightweight block ciphers: CubeCipher familiy and Lilliput; on the other hand, we work on the matrix method to simplify the representation of some attacks such as zero-correlation attack, impossible and integral attacks.

  We also published the extended version of our Secrypt 2013 paper in the journal Security and Communication Networks [2] concerning the performances on a dedicated platform.

- **Passwords Cracking** Passwords are widely used for user authentication, and will likely remain in use in the foreseeable future, despite several weaknesses. One important weakness is that human-generated passwords are far from being random, which makes them susceptible to guessing attacks. Understanding the adversaries' capabilities for guessing attacks is a fundamental necessity for estimating their impact and advising countermeasures. We develop OMEN [9], a new Markov model-based password cracker that extends ideas proposed by Narayanan and Shmatikov (CCS 2005). The main novelty of our tool is that it generates password candidates according to their occurrence probabilities, i.e., it outputs most likely passwords first. As shown by our extensive experiments, OMEN significantly improves guessing speed over existing proposals. In particular, we compare the performance of OMEN with the Markov mode of John the Ripper, which implements the password indexing function by Narayanan and Shmatikov. OMEN guesses more than 40% of passwords correctly with the first 90 million guesses, while JtR-Markov (for T = 1 billion) needs at least eight times as many guesses to reach the same goal, and OMEN guesses more than 80% of passwords correctly at 10 billion guesses, more than all probabilistic password crackers we compared against.

- **Time-memory trade-off** Cryptanalytic time-memory trade-offs (TMTO) are well-known tools available in any security expert toolbox. They have been used to break ciphers such as A5/1, but their efficiency to crack passwords made them even more popular in the security community. While symmetric keys are generated randomly according to a uniform distribution, passwords chosen by users are in practice far from being random, as confirmed by recent leakage of databases. Unfortunately, the technique used to build TMTOs is not appropriate to deal with non-uniform distributions. In [6], we introduce an efficient construction that consists in partitioning the search set into subsets of close densities, and a strategy to explore the TMTOs associated to the subsets based on an interleaved traversal. This approach results in a significant improvement compared to currently used TMTOs. We experimented our approach on a classical problem, namely cracking 7-character NTLM Hash passwords using an alphabet with 34 special characters, which resulted in a 16 × speedup over rainbow tables, which are considered as the most efficient variant of time-memory trade-offs.

PROSECCO Project-Team

7. New Results

7.1. Verification of Security Protocols in the Symbolic Model

Participants: Bruno Blanchet, Miriam Paiola.

The applied pi calculus is a widely used language for modeling security protocols, including as a theoretical basis of PROVERIF. However, the seminal paper that describes this language [24] does not come with proofs, and detailed proofs for the results in this paper were never published. This year, Martín Abadi, Bruno Blanchet, and Cédric Fournet finished the detailed proofs of all results of this paper, started last year, and added a new example on a symbolic analog of indifferentiability of hash functions. This work is submitted to a journal.

Previously [37], Bruno Blanchet and Miriam Paiola presented an automatic technique for proving secrecy and authentication properties for security protocols that manipulate lists of unbounded length, for an unbounded number of sessions. That work relies on an extension of Horn clauses, generalized Horn clauses, designed to support unbounded lists, and on a resolution algorithm on these clauses. However, in that previous work, they had to model protocols manually with generalized Horn clauses, which is unpractical. They recently extended the input language of ProVerif to model protocols with lists of unbounded length. They give the formal meaning of this extension, translate it automatically to generalized Horn clauses, and prove that this translation is sound. This work appears as a research report [21].

We implemented several extensions of ProVerif: Bruno Blanchet and Vincent Cheval improved the algorithm for proving observational equivalence between two processes, by merging them into a single biprocess that encodes the two processes. Bruno Blanchet also introduced a new construct **new** $a[x_1, ..., x_n]$ in ProVerif which allows to specify the arguments $x_1, \cdots, x_n$ used in the internal representation of the fresh name a. This extension allows one to tune the precision and speed of the analysis performed by ProVerif. The extended tool is available at http://proverif.inria.fr, and deposited to the APP (*Agence pour la Protection des Programmes*).

Stéphanie Delaune, Mark Ryan, and Ben Smyth [42] introduced the idea of swapping data in order to prove observational equivalence. For instance, ballot secrecy in electronic voting is formalized by saying that A voting a and B voting b is observationally equivalent to (indistinguishable from) A voting b and B voting a. Proving such an equivalence typically requires swapping the votes. However, Delaune et al's approach was never proved correct. Bruno Blanchet and Ben Smyth filled this gap by formalizing the approach and providing a detailed soundness proof. They plan to submit this work to a conference.

7.2. Verification of Security Protocols in the Computational model

Participant: Bruno Blanchet.

Bruno Blanchet implemented several extensions of his computational protocol verifier CryptoVerif. In particular, he improved the global dependency analysis, used in order to show that the result of all tests is independent from some random values. He improved the proof of secrecy properties, in particular to prove forward secrecy properties. He also improved the merging of branches of tests, in particular to be able to merge the two branches of **if**b**then**P_1**else**P_2 even when variables are renamed between P_1 and P_2. Finally, he added the display of an explanation of why a cryptographic transformation fails, to make the tool easier to use. The extended tool is available at http://cryptoverif.inria.fr.

Within the ANR project AnaStaSec, Bruno Blanchet verified an air-ground avionic security protocol (International Civil Aviation Organization (ICAO) Document 9880: Manual on Detailed Technical Specifications for the Aeronautical Telecommunication Network (ATN) using ISO/OSI standards and protocols, Part IV) using CryptoVerif. He proved entity authentication and message authenticity for the main protocol, in the computational model of cryptography, and made comments on some points that should be clarified in the protocol specification. He presented this work at a meeting of the secure dialog service working group of ICAO, in Toulouse, September 2015. The working group was strongly interested by the presentation and welcomed the proposal to apply these modelling and formal verification techniques as part of its validation activities.

7.3. The F* programming language

Participants: Nikhil Swamy [Microsoft Research], Catalin Hritcu, Chantal Keller [LRI], Aseem Rastogi [Univ of Maryland], Antoine Delignat-Lavaud, Simon Forest, Karthikeyan Bhargavan, Cedric Fournet [Microsoft Research], Pierre-Yves Strub [IMDEA], Markulf Kohlweiss [Microsoft Research], Jean Karim Zinzindohoue, Santiago Zanella Beguelin [Microsoft Research, MSR-Inria].

F* is a new higher order, effectful programming language (like ML) designed with program verification in mind. Its type system is based on a core that resembles System Fω (hence the name), but is extended with dependent types, refined monadic effects, refinement types, and higher kinds. Together, these features allow expressing precise and compact specifications for programs, including functional correctness properties. The F* type-checker aims to prove that programs meet their specifications using an automated theorem prover (usually Z3) behind the scenes to discharge proof obligations. Programs written in F* can be translated to OCaml, F#, or JavaScript for execution. We published a paper on the design, implementation, and formal core of F* at POPL 2016. F* is being developed as an open-source project at GitHub: https://github.com/FStarLang and the official webpage is at http://fstar-lang.org. We released several beta versions of the software this year.

7.4. Micro-Policies and Secure Compilation

Participants: Catalin Hritcu, Arthur Azevedo de Amorim, Zoi Paraskevopoulou, Nikolaos Giannarakis.

Following on from previous work on the *micro-policy* framework, Catalin Hritcu and his collaborators published new work on applications and efficient implementations of micro-policies. They published work on low-level implementations of micro-policies at ASPLOS 2015 [18]. At IEEE S&P, they published a paper how to write formally verified reference monitors using micro-policies [26].

Other than these published works, Hritcu and his colleagues also worked on using micro-policies to enforce secure information flow at the hardware level [25], and a secure compiler for a high-level language that relies on micro-policies to enforce programming language abstractions [45].

7.5. Dependable Property-Based Testing

Participants: Catalin Hritcu, Zoi Paraskevopoulou.

Catalin Hritcu and his student, Zoi Paraskevopoulou, worked on a methodology for formally verified property-based testing and implemented it as a foundational verification framework for QuickChick, a port of QuickCheck to Coq. This work was published at ITP 2015 [19]. Catalin Hritcu also worked with a number of co-authors on a new technique for creating random generators for property-based testing. This work is currently under submission [46].

7.6. Attacks and Proofs for Transport Layer Security

Participants: Benjamin Beurdouche, Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cedric Fournet [Microsoft Research], Markulf Kohlweiss [Microsoft Research], Alfredo Pironti, Pierre-Yves Strub [IMDEA], Jean Karim Zinzindohoue.

As a countermeasure to our earlier work on the triple handshake attack, we proposed a TLS extension called *session hash* which has now been published as an Internet standard (IETF RFC 7627). We also formally analyzed various protocols such as TLS, IKE, and SSH for key synchronization and triple handshake attacks, and proved that our session hash countermeasure prevents such attacks on TLS. This work appeared at NDSS 2015 [15].

We discovered and reported an important class of *state machine attacks* on implementations of the Transport Layer Security (TLS) protocol. These attacks appear when TLS implementations incorrectly accept messages which are forbidden by the TLS state machine. We built a test framework for such attacks and analyzed a number of open source implementations. Our analysis uncovered critical vulnerabilities such as the SKIP attack on Java and the FREAK attack on almost all mainstream web browsers. The research results were published at IEEE S&P where our paper won a distinguished paper award [14]. Our work also led to security updates and CVEs for many web browsers, TLS libraries, and web servers.

Along with colleagues at several other institutions, we discovered the Logjam vulnerability on protocols that still support weak Diffie-Hellman groups in their key exchange. We showed that the attack could be used for online and offline attacks on real-world TLS clients and servers. We also showed how the vulnerability could weaken the security of IPsec and SSH connections. Our research led to widespread changes to the configurations of web servers, mail servers, web browsers, and TLS libraries. The research was published at ACM CCS 2015 [12] where it won a Best Paper award.

Antoine Delignat-Lavaud showed how the unsafe sharing of certificates across multiple HTTPS websites could be exploited to fully compromise the same origin policy for websites, using a vulnerability called *virtual host confusion*. A research paper on these attacks appeared at WWW 2015 [17].

7.7. Privacy, Electronic Voting, and Auctions

Participants: Benjamin Smyth [correspondant], Elizabeth Quaglia.

Benjamin Smyth worked on a formal analysis of privacy in Direct Anonymous Attestation schemes [50]. He also showed how to verify commitment protocols in ProVerif without False attacks [39].

Apart from these published works, Benjamin Smyth and Elizabeth Quaglia worked on formal security analyses of electronic auction schemes based on existing models for electronic voting [48]. Benjamin Smyth worked on developing new formal definitions for secrecy and independence in election schemes [51], and on applying such definitions to the security analysis of real-world voting protocols such as Helios and JCJ [49].

7.8. Computationally Complete Symbolic Attacker Models

Participants: Gergei Bana, Hubert Comon-Lundh [ENS Cachan], Rohit Chadha [University of Missouri].

In previous work, Bana and Comon-Lunch proposed a new approach to computational verification of cryptographic protocols, by defining a *computationally complete* symbolic attacker, so that a symbolic proof against this attacker can be shown to imply a computational proof of security [27], [28].

Following on from this work, Bana and Chadha fully developed the core parts of the computationally complete symbolic attacker based on indistinguishability. This covers both trace properties and equivalence properties and can be proved partially complete. They evaluated their method by applying it to several classic protocols. This work is currently under submission.

Bana, Comon-Lundh, and Koutsos also worked on a decision procedure for the computationally complete symbolic attacker based on indistinguishability.