RESEARCH CENTER

FIELD
Algorithmics, Programming, Software and Architecture

Activity Report 2013

Section New Results

Edition: 2014-03-20

ARIC Project-Team

6. New Results

6.1. Cryptography and lattices

6.1.1. Group signatures

Group signatures are cryptographic primitives where users can anonymously sign messages in the name of a population they belong to. Gordon et al. (Asiacrypt 2010) suggested the first realization of group signatures based on lattice assumptions in the random oracle model. A significant drawback of their scheme is its linear signature size in the cardinality N of the group. A recent extension proposed by Camenisch et al. (SCN 2012) suffers from the same overhead.

F. Laguillaumie, A. Langlois, B. Libert (Technicolor), and D. Stehlé described in [24] the first lattice-based group signature schemes where the signature and public key sizes are essentially logarithmic in N (for any fixed security level). Their basic construction only satisfies a relaxed definition of anonymity (just like the Gordon et al. system) but readily extends into a fully anonymous group signature (i.e., that resists adversaries equipped with a signature opening oracle). They proved the security of their schemes in the random oracle model under the SIS and LWE assumptions.

6.1.2. Classical hardness of learning with errors

Z. Brakerski (Stanford U.), A. Langlois, C. Peikert (Georgia Institute of Technology), O. Regev (Courant Institute, New York U.), and D. Stehlé showed in [16] that the Learning with Errors (LWE) problem is classically at least as hard as standard worst-case lattice problems, even with polynomial modulus. Previously this was only known under quantum reductions. Their techniques capture the tradeoff between the dimension and the modulus of LWE instances, leading to a much better understanding of the landscape of the problem. The proof is inspired by techniques from several recent cryptographic constructions, most notably fully homomorphic encryption schemes.

6.1.3. Improved Zero-knowledge Proofs of Knowledge for the ISIS Problem, and Applications

In all existing efficient proofs of knowledge of a solution to the infinity norm Inhomogeneous Small Integer Solution ISIS_∞ problem, the knowledge extractor outputs a solution vector that is only guaranteed to be $\widetilde{O}(n)$ times longer than the witness possessed by the prover. As a consequence, in many cryptographic schemes that use these proof systems as building blocks, there exists a gap between the hardness of solving the underlying ISIS_∞ problem and the hardness underlying the security reductions. Together with S. Ling, K. Nguyen, and H. Wang (Nanyang Technological University, Singapore), D. Stehlé generalized in [26] Stern's protocol to obtain two statistical zero-knowledge proofs of knowledge for the ISIS_∞ problem that remove this gap. Their result yields the potential of relying on weaker security assumptions for various lattice-based cryptographic constructions. As applications of their proof system, they introduced a concurrently secure identity-based identification scheme based on the worst-case hardness of the $\text{SIVP}_{\widetilde{O}(n^{1.5})}$ problem (in the L2 norm) in general lattices in the random oracle model, and an efficient statistical zero-knowledge proof of plaintext knowledge with small constant gap factor for Regev's encryption scheme.

6.1.4. Decoding by Embedding: Correct Decoding Radius and DMT Optimality

In lattice-coded multiple-input multiple-output (MIMO) systems, optimal decoding amounts to solving the closest vector problem (CVP). Embedding is a powerful technique for the approximate CVP, yet its remarkable performance is not well understood. In [8], C. Ling (Imperial College, London), L. Luzzi (ENSEA, U. Cergy Pontoise), and D. Stehlé analyzed the embedding technique from a bounded distance decoding (BDD) viewpoint. They proved that the Lenstra, Lenstra and Lovász (LLL) algorithm can achieve $1/(2\gamma)$-BDD for $\gamma \approx O(2^{n/4})$, yielding a polynomial-complexity decoding algorithm performing exponentially better than Babai's which achieves $\gamma = O(2^{n/2})$. This substantially improves the existing result $\gamma = O(2^n)$ for embedding decoding. They also proved that BDD of the regularized lattice is optimal in terms of the diversity-multiplexing gain tradeoff (DMT).

6.1.5. A New View on HJLS and PSLQ: Sums and Projections of Lattices

The HJLS and PSLQ algorithms are the de facto standards for discovering non-trivial integer relations between a given tuple of real numbers. In [19], J. Chen, D. Stehlé, and G. Villard provided a new interpretation of these algorithms, in a more general and powerful algebraic setup: they view them as special cases of algorithms that compute the intersection between a lattice and a vector subspace. Further, they extracted from them the first algorithm for manipulating finitely generated additive subgroups of a Euclidean space, including projections of lattices and finite sums of lattices. They adapted the analyses of HJLS and PSLQ to derive correctness and convergence guarantees. They also investigated another approach based on embedding the input in a higher dimensional lattice and calling the LLL lattice reduction algorithm.

6.2. Certified computing and computer algebra

6.2.1. Polynomial system solving

Polynomial system solving is a core topic of computer algebra. While the worst-case complexity of this problem is known to be hopelessly large, the practical complexity for large families of systems is much more reasonable. Progress has been made in assessing precise complexity estimates in this area.

First, M. Bardet (U. Rouen), J.-C. Faugère (PolSys team), and B. Salvy studied the complexity of Gröbner bases computations, in particular in the generic situation where the variables are in simultaneous Noether position with respect to the system. They gave a bound on the number of polynomials of each degree in a Gröbner basis computed by Faugère's F5 algorithm in this generic case for the grevlex ordering (which is also a bound on the number of polynomials for a reduced Gröbner basis), and used it to bound the exponent of the complexity of the F5 algorithm [35].

Next, a fundamental problem in computer science is to find all the common zeroes of m quadratic polynomials in n unknowns over F_2. The cryptanalysis of several modern ciphers reduces to this problem. Up to now, the best complexity bound was reached by an exhaustive search in $4\log_2 n\, 2^n$ operations. In [1], M. Bardet (U. Rouen), J.-C. Faugère (PolSys team), B. Salvy, and P.-J. Spaenlehauer (CARAMEL team) gave an algorithm that reduces the problem to a combination of exhaustive search and sparse linear algebra. This algorithm has several variants depending on the method used for the linear algebra step. Under precise algebraic assumptions, they showed that the deterministic variant of their algorithm has complexity bounded by $O(2^{0.841n})$ when $m = n$, while a probabilistic variant of the Las Vegas type has expected complexity $O(2^{0.792n})$. Experiments on random systems showed that the algebraic assumptions are satisfied with probability very close to 1. They have also given a rough estimate for the actual threshold between their method and exhaustive search, which is as low as 200, and thus very relevant for cryptographic applications.

6.2.2. Linear differential equations

Creative telescoping algorithms compute linear differential equations satisfied by multiple integrals with parameters. Together with A. Bostan and P. Lairez (SpecFun team), B. Salvy described a precise and elementary algorithmic version of the Griffiths–Dwork method for the creative telescoping of rational functions. This leads to bounds on the order and degree of the coefficients of the differential equation, and to the first complexity result which is simply exponential in the number of variables. One of the important features of the algorithm is that it does not need to compute certificates. The approach is vindicated by a prototype implementation [15].

In [2], B. Salvy proved with A. Bostan (SpecFun team) and K. Raschel (U. Tours) that the sequence $(e_n^{\mathfrak{S}})_{n\geq 0}$ of excursions in the quarter plane corresponding to a nonsingular step set $\mathfrak{S} \subseteq \{0, \pm 1\}^2$ with infinite group does not satisfy any nontrivial linear recurrence with polynomial coefficients. Accordingly, in those cases, the trivariate generating function of the numbers of walks with given length and prescribed ending point is not D-finite. Moreover, they displayed the asymptotics of $e_n^{\mathfrak{S}}$. This completes the classification of these walks.

With F. Johansson and M. Kauers (RISC, Linz, Austria), M. Mezzarobba presented in [23] a new algorithm for computing hyperexponential solutions of ordinary linear differential equations with polynomial coefficients. The algorithm relies on interpreting formal series solutions at the singular points as analytic functions and evaluating them numerically at some common ordinary point. The numerical data is used to determine a small number of combinations of the formal series that may give rise to hyperexponential solutions.

6.2.3. Exact linear algebra

Transforming a matrix over a field to echelon form, or decomposing the matrix as a product of simpler matrices that reveal the rank profile, is a fundamental building block of computational exact linear algebra. For such tasks the best previously available algorithms were either rank sensitive (i.e., of complexity expressed in terms of the exponent of matrix multiplication and the rank of the input matrix) or in place (i.e., using essentially no more memory that what is needed for matrix multiplication). In [6] C.-P. Jeannerod, C. Pernet, and A. Storjohann (U. Waterloo, Canada) have proposed algorithms that are both rank sensitive and in place. These algorithms required to introduce a matrix factorization of the form $A = CUP$ with C a column echelon form giving the row rank profile of the input matrix A, U a unit upper triangular matrix, and P a permutation matrix.

6.2.4. Certified multiple-precision evaluation of the Airy Ai function

The series expansion at the origin of the Airy function $Ai(x)$ is alternating and hence problematic to evaluate for $x > 0$ due to cancellation. S. Chevillard (APICS team) and M. Mezzarobba showed in [20] how an arbitrary and certified accuracy can be obtained in that case. Based on a method recently proposed by Gawronski, Müller, and Reinhard, they exhibited two functions F and G, both with nonnegative Taylor expansions at the origin, such that $Ai(x) = G(x)/F(x)$. The sums are now well-conditioned, but the Taylor coefficients of G turn out to obey an ill-conditioned three-term recurrence. They then used the classical Miller algorithm to overcome this issue. Finally, they bounded all errors and proposed an implementation which, by allowing an arbitrary and certified accuracy, can be used for example to provide correct rounding in arbitrary precision.

6.2.5. Standardization of interval arithmetic

The IEEE 1788 working group is devoted to the standardization of interval arithmetic. V. Lefèvre and N. Revol are very active in this group. This year is the last year granted by IEEE for the preparation of a draft text of the standard. 2014 will be devoted to a ballot on the whole text, first by the standardization working group and then by a group of experts appointed by IEEE. In 2013, the definition of interval literals, of constructors, and of input and output has been adopted. The work now concentrates on portions of the final text [42].

6.2.6. Parallel product of interval matrices

The problem considered here is the multiplication of two matrices with interval coefficients. Parallel implementations by N. Revol and Ph. Théveny [10] compute results that satisfy the inclusion property, which is the fundamental property of interval arithmetic, and offer good performances: the product of two interval matrices is not slower than 15 times the product of two floating-point matrices.

6.2.7. Numerical reproducibility

What is called *numerical reproducibility* is the problem of getting the same result when the scientific computation is run several times, either on the same machine or on different machines. In [43], the focus is on interval computations using floating-point arithmetic: N. Revol identifies implementation issues that may invalidate the inclusion property, and presents several ways to preserve this inclusion property. This work has also been presented at several conferences [30], [29], [31].

6.3. Floating-point arithmetic

6.3.1. Improved error bounds for complex floating-point arithmetic with a fused-multiply add

Assuming that a fused multiply-add (FMA) instruction is available, C.-P. Jeannerod, N. Louvet, and J.-M. Muller [22] obtained sharp error bounds for various alternatives to Kahan's FMA-based algorithm for 2 x

2 determinants (which they had analyzed in [5]). They showed how to combine such variants with Kahan's original scheme in order to derive componentwise-accurate algorithms for complex floating-point division. Finally, they established sharp or reasonably sharp error bounds for each of these division algorithms.

C.-P. Jeannerod, P. Kornerup (U. of Southern Denmark), N. Louvet, and J.-M. Muller [36] studied the impact of the FMA on the normwise relative accuracy of complex floating-point multiplication. They showed that the classical normwise relative error bound $\sqrt{5}\,u$ (with u the unit roundoff) can be decreased further to $2u$, and that this new constant is best possible for several FMA-based multiplication algorithms.

J.-M. Muller analyzed in [41] another 2 x 2 determinant algorithm, due to Cornea, Harrison, and Tang, and showed that for radix 2 it admits a sharp relative error bound of the form $2u + O(u^2)$.

6.3.2. Improved error bounds for numerical linear algebra

C.-P. Jeannerod and S. M. Rump (Hamburg University of Technology) [7] showed that when evaluating sums of n real numbers in standard floating-point arithmetic, the usual fraction $\gamma_n = nu/(1-nu)$, which has the form $nu + O(u^2)$ and requires $nu < 1$, can be replaced by nu without any restriction on n. Applications include simpler and more general error bounds for inner products, matrix-vector multiplication, and classical matrix multiplication.

In [45] they extended these results to LU and Cholesky factorizations as well as to triangular linear system solving by showing that the constants γ_n that appear classically in the backward error bounds for such problems can all be replaced by $O(u^2)$-free and unconditional constants nu. To get these new bounds the main ingredient is a general framework for bounding expressions of the form $|\rho - s|$, where s is the exact sum of a floating-point number and $n-1$ real numbers, and where ρ is a real number approximating the computed sum $\widehat{s}$.

6.3.3. On Ziv's rounding test

F. de Dinechin, J.-M. Muller and S. Torres studied with C. Lauter (Univ. Paris 6) the rounding test introduced by Ziv in its libultim software [4]. This test determines if an approximation to the value $f(x)$ of an elementary function at a given point x suffices to return the floating-point number nearest to $f(x)$. They showed that the same test may be used for efficient implementation of floating-point operations with input and output operands of different formats. That test depends on a "magic constant" e and they also showed how to choose that constant to make the test reliable and efficient. Various cases are considered, depending on the availability of an FMA instruction, and on the range of $f(x)$.

6.3.4. Various issues related to double roundings

Double rounding is a phenomenon that may occur when different floating-point precisions are available on the same system. Although double rounding is, in general, innocuous, it may change the behavior of some useful floating-point algorithms. G. Melquiond (Toccata team), E. Martin-Dorel (then in the Marelle team), and J.-M. Muller analyzed in [9] the potential influence of double rounding on the Fast2Sum and 2Sum algorithms, on some summation algorithms, and Veltkamp's splitting. When performing divisions using Newton-Raphson (or similar) iterations on a processor with a floating-point fused multiply-add instruction, one must sometimes scale the iterations, to avoid over/underflow and/or loss of accuracy. This may lead to double-roundings, resulting in output values that may not be correctly rounded when the quotient falls in the subnormal range. J.-M. Muller showed in [13] how to avoid this problem.

6.3.5. Comparison between binary and decimal floating-point numbers

The IEEE 754-2008 standard for floating-point arithmetic arithmetic specifies binary as well as decimal formats. N. Brisebarre, C. Lauter (Univ. Paris 6), M. Mezzarobba, and J.-M. Muller introduced in [17] an algorithm that allows one to quickly compare a binary64 floating-point number and a decimal64 floating-point number, assuming the "binary encoding" of the decimal formats specified by the IEEE-754 standard is used. It is a two-step algorithm: a first pass, based on the exponents only, makes it possible to quickly eliminate most cases; then, when the first pass does not suffice, a more accurate second pass is required. They provide an implementation of several variants of their algorithm, and compare them.

6.3.6. Conversions between binary and decimal floating-point numbers

Conversion between binary and decimal floating-point representations is ubiquitous. Floating-point radix conversion means converting both the exponent and the mantissa. O. Kupriianova and C. Lauter (Univ. Paris 6) and J.-M. Muller developed in [38] an atomic operation for floating-point radix conversion with simple straight-line algorithm, suitable for hardware design. Exponent conversion is performed with a small multiplication and a lookup table. It yields the correct result without error. Mantissa conversion uses a few multiplications and a small lookup table that is shared amongst all types of conversions. The accuracy changes by adjusting the computing precision.

6.3.7. Table-maker's dilemma

Computing hardest-to-round cases of elementary functions is a key issue when one wants to develop an efficient and reliable implementation of such a function. The algorithms developed until now required a large amount of computation and produced a simple yes/no answer. In [40], G. Hanrot developed together with E. Martin-Dorel (Toccata team), M. Mayero (IUT Villetaneuse, LIPN), and L. Théry (Marelle team) a certificate-based approach of the SLZ algorithm where the execution produces certificates which can then be validated using Coq. This allows one to validate a posteriori the fact that for a given function, a given input precision p and bound p', there is no pair (x, y) of floating-point representable numbers in precision p such that $2^{-e_p(f(x))}|f(x) - y| \leq 2^{-p'}$. This approach has been tested on the exponential function over $[1/2, 1]$, with an input precision of 53 bits and $p' = 300$.

6.4. Hardware and FPGA arithmetic

6.4.1. Reconfiguring arithmetic

With B. Pasca (Altera), F. de Dinechin contributed a book chapter about of the opportunities and challenges of computer arithmetic for reconfigurable/FPGA computing [32]. The main point of this chapter is to look beyond the heritage of processor arithmetic. Using many examples from the FloPoCo project and others, it shows the benefits of merging and fusing standard operators, it introduces an open-ended space of non-standard operators, and illustrates the power of machine-generation of such arithmetic cores.

6.4.2. The bit heap framework for fixed-point arithmetic

N. Brunie, F. de Dinechin, and M. Istoan, with students G. Sergent, K. Illyes, and B. Popa, extended FloPoCo with a versatile framework for manipulating sums of weighted bits [28], [18]. Such bit heaps may be used to express and optimize at the bit level a wide range of operators (from adders and multipliers to polynomials, filters, and other coarse arithmetic cores). A single piece of code can then be used to generate an architecture for any of these operators.

6.4.3. Elementary functions

F. de Dinechin, with P. Echeverria and M. Lopez-Vallejo (U. Madrid) and B. Pasca (Altera), published a hardware architecture for the floating-point pow and powr functions of the IEEE-754-2008 standard [3]. These functions compute x^y, and differ only in the specification of special cases. The implementation, distributed in FloPoCo, is parameterized in exponent and significand size. It combines suitably modified exponential and logarithm units.

F. de Dinechin and M. Istoan, with student G. Sergent, compared several hardware algorithms for the implementation of sine, cosine, and combined sine/cosine [21]: unrolled CORDIC in two variants with several minor improvements, polynomial approximation, and an ad-hoc architecture based on trigonometric identities. A surprising result is that the ad-hoc architecture betters CORDIC even when its multipliers and tables are synthesized as logic.

6.4.4. Contributions to processor architecture

S. Collange (ALF team) and N. Brunie with G. Diamos (Nvidia) suggested improvements for the architecture of general-purpose graphical processing units [11]. As threads take different paths across the control-flow graph, SIMD lockstep execution is partially lost, and must be regained whenever possible in order to maximize the occupancy of SIMD units. Two techniques are described to handle SIMT control divergence and identify reconvergence points. The most advanced one operates in constant space and handles indirect jumps and recursion. In terms of performance, this solution is at least as efficient as state-of-the-art techniques in use in current GPUs.

N. Brunie and F. de Dinechin studied with B. de Dinechin (Kalray) the integration of a tightly coupled reconfigurable accelerator in a massively parallel multiprocessor [27]. For this purpose, they described an architecture exploration framework that produces an architecture along with the relevant compilation software. This framework was demonstrated on AES, SHA2, and a FIR filter.

CARAMEL Project-Team

6. New Results

6.1. Computation of Discrete Logarithms in $\mathrm{GF}(2^{809})$

Participants: Razvan Barbulescu, Cyril Bouvier, Jérémie Detrey, Pierrick Gaudry, Hamza Jeljeli, Emmanuel Thomé [contact], Marion Videau, Paul Zimmermann.

In the context of the CATREL ANR project, most team members contributed to the achievement of a new record computation for discrete logarithms in $\mathrm{GF}(2^{809})$, with the Function Field Sieve (FFS) algorithm. This is, to date, the largest computation in a binary field of prime extension degree. Beyond the experimental data and the improvements related to "what it takes" to beat such a record, this work provides very useful basis information towards the assessment of the cut-off with the novel quasi-polynomial algorithm discussed below.

This work has been reported in the article [15], accepted for publication in the conference PKC 2014 (Public Key Cryptography). It was the occasion to illustrate several contributions of members of the teams to various phases of the algorithm: Răzvan Bărbulescu [21] analyzed the polynomial selection step for FFS; Jérémie Detrey, Pierrick Gaudry and Marion Videau [17] improved the practical implementation of the relation collection; Cyril Bouvier [23] studied the filtering step; and Hamza Jeljeli [28] proposed to use the Residue Number System representation for the linear algebra step on GPU and CPU.

6.2. A Quasi-polynomial Algorithm for the Computation of Discrete Logarithms in Finite Fields of Small Characteristic

Participants: Razvan Barbulescu, Pierrick Gaudry, Emmanuel Thomé [contact].

In collaboration with Antoine Joux (Université Pierre et Marie Curie), Răzvan Bărbulescu, Pierrick Gaudry, and Emmanuel Thomé designed a new algorithm of quasi-polynomial complexity for computing discrete logarithms in finite fields $\mathrm{GF}(p^n)$, under the constraint that the characteristic p is small: it must not grow faster than a polynomial in the input size $n \log p$. This constraint accomodates for instance the cryptographically relevant case of finite fields of fixed characteristic $\mathrm{GF}(2^n)$ and $\mathrm{GF}(3^n)$.

This new algorithm dramatically changes the complexity landscape of the computation of discrete logarithms in finite fields. This has in particular an immense impact on the small characteristic pairing-based cryptography proposals. As it turns out, the field of definition of the Weil pairing for curves over small characteristic fields lends itself incredibly well to the new algorithm, to the point that the key sizes which are necessary to claim a sufficient security suddenly become unacceptably large. The newly proposed algorithm practically kills such cryptosystems.

This work has been published in preprint form in June 2013 [22] and was immediately acclaimed as a breakthrough, receiving also some external publicity. Pending the submission outcome, a first publication is expected in 2014.

6.3. Computation of CM Class Polynomials for Genus 2 Jacobians

Participant: Emmanuel Thomé [contact].

In collaboration with Andreas Enge, Emmanuel Thomé has developed software for computing class polynomials, in the context of complex multiplication theory in genus 2. The current computations set new records which are well above the previous state of the art, as Igusa class polynomials for class number above 20,000 have been computed in december 2013 using this software. An article describing this work has been accepted for publication in *Experimental Mathematics* [11].

Using similar underlying tools and theory, and based on work by Sorina Ionica [13], Sorina Ionica and Emmanuel Thomé have worked on the analysis of isogeny graphs in genus 2, when certain properties of the endomorphism ring are satisfied. A publication is being worked on, and is expected to be submitted in early 2014.

6.4. Binary to Decimal Conversion

Participants: Cyril Bouvier, Paul Zimmermann.

Cyril Bouvier and Paul Zimmermann designed a new algorithm to convert a large binary integer to decimal (or more generally any non-power-of-two radix). Compared to the reference implementation in GNU MP, this algorithm replaces divisions by multiplications, and exhibits a speedup of up to a factor of two (or more) in some cases [24].

6.5. Fast Change of Ordering for Gröbner Bases

Participant: Pierrick Gaudry.

When solving polynomial systems, the usual approach is to compute a Gröbner basis for a monomial order that is compatible with the degree with the F4 or F5 algorithm, and then compute a Gröbner basis for the lexicographical order using the FGLM algorithm. In collaboration with Jean-Charles Faugère, Louise Huot and Guénaël Renault, Pierrick Gaudry designed another approach [27] for this second step, leading to a better asymptotic complexity: the cubic complexity is replaced by the complexity of the linear algebra where the exponent can theoretically be as small as 2.37.

CASCADE Project-Team (section vide)

CRYPT Team (section vide)

GEOMETRICA Project-Team

6. New Results

6.1. Mesh Generation and Geometry Processing

6.1.1. Splat-based Surface Reconstruction from Defect-Laden Point Sets.

Participant: Mariette Yvinec.

In collaboration with Pierre Alliez (EPI Titane), Ricard Campos (University of Girona), Raphael Garcia (University of Girona)

We introduce a method for surface reconstruction from point sets that is able to cope with noise and outliers. First, a splat-based representation is computed from the point set. A robust local 3D RANSAC-based procedure is used to filter the point set for outliers, then a local jet surface – a low-degree surface approximation – is fitted to the inliers. Second, we extract the reconstructed surface in the form of a surface triangle mesh through Delaunay refinement. The Delaunay refinement meshing approach requires computing intersections between line segment queries and the surface to be meshed. In the present case, intersection queries are solved from the set of splats through a 1D RANSAC procedure. [14].

6.1.2. Constructing Intrinsic Delaunay Triangulations of Submanifolds

Participants: Jean-Daniel Boissonnat, Ramsay Dyer.

In collaboration with Arijit Ghosh (Indian Statistical Institute)

We describe an algorithm to construct an intrinsic Delaunay triangulation of a smooth closed submanifold of Euclidean space [42]. Using results established in a companion paper on the stability of Delaunay triangulations on δ-generic point sets, we establish sampling criteria which ensure that the intrinsic Delaunay complex coincides with the restricted Delaunay complex and also with the recently introduced tangential Delaunay complex. The algorithm generates a point set that meets the required criteria while the tangential complex is being constructed. In this way the computation of geodesic distances is avoided, the runtime is only linearly dependent on the ambient dimension, and the Delaunay complexes are guaranteed to be triangulations of the manifold.

6.1.3. Delaunay Triangulation of Manifolds

Participants: Jean-Daniel Boissonnat, Ramsay Dyer.

In collaboration with Arijit Ghosh (Indian Statistical Institute)

We present an algorithmic framework for producing Delaunay triangulations of manifolds [44]. The input to the algorithm is a set of sample points together with coordinate patches indexed by those points. The transition functions between nearby coordinate patches are required to be bi-Lipschitz with a constant close to 1. The primary novelty of the framework is that it can accommodate abstract manifolds that are not presented as submanifolds of Euclidean space. The output is a manifold simplicial complex that is the Delaunay complex of a perturbed set of points on the manifold. The guarantee of a manifold output complex demands no smoothness requirement on the transition functions, beyond the bi-Lipschitz constraint. In the smooth setting, when the transition functions are defined by common coordinate charts, such as the exponential map on a Riemannian manifold, the output manifold is homeomorphic to the original manifold, when the sampling is sufficiently dense.

6.1.4. Anisotropic Delaunay Meshes of Surfaces

Participants: Jean-Daniel Boissonnat, Mariette Yvinec.

In collaboration with Jane Tournois (GeometryFactory) and Kan-Le Shi (Tsing Hua University)

Anisotropic simplicial meshes are triangulations with elements elongated along prescribed directions. Anisotropic meshes have been shown to be well suited for interpolation of functions or solving PDEs. They can also significantly enhance the accuracy of a surface representation. Given a surface S endowed with a metric tensor field, we propose a new approach to generate an anisotropic mesh that approximates S with elements shaped according to the metric field [13], [47]. The algorithm relies on the well-established concepts of restricted Delaunay triangulation and Delaunay refinement and comes with theoretical guarantees. The star of each vertex in the output mesh is Delaunay for the metric attached to this vertex. Each facet has a good aspect ratio with respect to the metric specified at any of its vertices. The algorithm is easy to implement. It can mesh various types of surfaces like implicit surfaces, polyhedra or isosurfaces in 3D images. It can handle complicated geometries and topologies, and very anisotropic metric fields.

6.2. Topological and Geometric Inference

6.2.1. An Efficient Data Structure for Computing Persistent Cohomology

Participants: Jean-Daniel Boissonnat, Clément Maria.

In collaboration with Tamal Dey (Ohio State University)

Persistent homology with coefficients in a field F coincides with the same for cohomology because of duality. We propose an implementation of a recently introduced algorithm for persistent cohomology that attaches annotation vectors with the simplices. We separate the representation of the simplicial complex from the representation of the cohomology groups, and introduce a new data structure for maintaining the annotation matrix, which is more compact and reduces substancially the amount of matrix operations. In addition, we propose a heuristic to further simplify the representation of the cohomology groups and improve both time and space complexities. The paper provides a theoretical analysis, as well as a detailed experimental study of our implementation and comparison with state-of-the-art software for persistent homology and cohomology [41], [29].

6.2.2. Multi-Field Persistent Homology

Participants: Jean-Daniel Boissonnat, Clément Maria.

In [46], we introduce the *multi-field persistence diagram* for the persistence homology of a filtered complex. It encodes compactly the *superimposition* of the persistence diagrams of the complex with several field coefficients, and provides a substantially more precise description of the topology of the filtered complex. Specifically, the multi-field persistence diagram encodes the Betti numbers of integral homology and the prime divisors of the torsion coefficients of the underlying shape. Moreover, it enjoys similar stability properties as the ones of standard persistence diagrams, with the appropriate notion of distance. These properties make the multi-field persistence diagram a useful tool in computational topology. The multi-field algorithms are, in practice, as fast as algorithms that compute persistent homology in a single field.

6.2.3. Zigzag Zoology: Rips Zigzags for Homology Inference

Participants: Steve Oudot, Donald Sheehy.

For points sampled near a compact set X, the persistence barcode of the Rips filtration built from the sample contains information about the homology of X as long as X satisfies some geometric assumptions. The Rips filtration is prohibitively large, however zigzag persistence can be used to keep the size linear. We present several species of Rips-like zigzags and compare them with respect to the signal-to-noise ratio, a measure of how well the underlying homology is represented in the persistence barcode relative to the noise in the barcode at the relevant scales. Some of these Rips-like zigzags have been available as part of the Dionysus library for several years while others are new. Interestingly, we show that some species of Rips zigzags will exhibit less noise than the (non-zigzag) Rips filtration itself. Thus, Rips zigzags can offer improvements in both size complexity and signal-to-noise ratio. Along the way, we develop new techniques for manipulating and comparing persistence barcodes from zigzag modules. In particular, we give methods for reversing arrows and removing spaces from a zigzag while controlling the changes occurring in its barcode. We also discuss

factoring zigzags and a kind of interleaving of two zigzags that allows their barcodes to be compared. These techniques were developed to provide our theoretical analysis of the signal-to-noise ratio of Rips-like zigzags, but they are of independent interest as they apply to zigzag modules generally [33].

6.2.4. *Efficient and Robust Topological Data Analysis on Metric Spaces*

Participants: Mickaël Buchet, Frédéric Chazal, Steve Oudot, Donald Sheehy.

We extend the notion of the distance to a measure from Euclidean space to probability measures on general metric spaces as a way to perform topological data analysis in a way that is robust to noise and outliers. We then give an efficient way to approximate the sub-level sets of this function by a union of metric balls and extend previous results on sparse Rips filtrations to this setting. This robust and efficient approach to topological data analysis is illustrated with several examples from an implementation [54].

6.2.5. *Noise-Adaptive Shape Reconstruction from Raw Point Sets*

Participant: David Cohen-Steiner.

In collaboration with Pierre Alliez (EPI Titane), Simon Giraudot (EPI Titane)

We propose a noise-adaptive shape reconstruction method specialized to smooth, closed hypersurfaces. Our algorithm takes as input a defect-laden point set with variable noise and outliers, and comprises three main steps. First, we compute a novel type of robust distance function to the data. As a robust distance function, its sublevel-sets have the correct homotopy type when the data is a sufficiently good sample of a regular shape. The new feature is a built-in scale selection mechanism that adapts to the local noise level, under the assumption that the inferred shape is a smooth submanifold of known dimension. Second, we estimate the sign and confidence of the function at a set of seed points, based on estimated crossing parities along the edges of a uniform random graph. That component is inspired by the classical MAXCUT relaxation, except that we only require a linear solve as opposed to an eigenvector computation. Third, we compute a signed implicit function through a random walker approach with soft constraints chosen as the most confident seed points computed in previous step. The resulting pipeline is scalable and offers excellent behavior for data exhibiting variable noise levels [19].

6.2.6. *Optimal Rates of Convergence for Persistence Diagrams in Topological Data Analysis*

Participants: Frédéric Chazal, Marc Glisse, Bertrand Michel.

In collaboration with Catherine Labruère (Université de Bourgogne).

Computational topology has recently known an important development toward data analysis, giving birth to the field of topological data analysis. Topological persistence, or persistent homology, appears as a fundamental tool in this field. In this paper [57] (to appear in proc. ICML 2014), we study topological persistence in general metric spaces, with a statistical approach. We show that the use of persistent homology can be naturally considered in general statistical frameworks and persistence diagrams can be used as statistics with interesting convergence properties. Some numerical experiments are performed in various contexts to illustrate our results.

6.2.7. *Bootstrap and Stochastic Convergence for Persistence Diagrams and Landscapes*

Participant: Frédéric Chazal.

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

Persistent homology probes topological properties from point clouds and functions. By looking at multiple scales simultaneously, one can record the births and deaths of topological features as the scale varies. We can summarize the persistent homology with the persistence landscape, introduced by Bubenik, which converts a diagram into a well-behaved real-valued function. We investigate the statistical properties of landscapes, such as weak convergence of the average landscapes and convergence of the bootstrap. In addition, we introduce an alternate functional summary of persistent homology, which we call the silhouette, and derive an analogous statistical theory [55].

6.2.8. Gromov-Hausdorff Approximation of Metric Spaces with Linear Structure

Participant: Frédéric Chazal.

In collaboration with S. Jian (Tsinghua University).

In many real-world applications data come as discrete metric spaces sampled around 1-dimensional filamentary structures that can be seen as metric graphs. In this paper [58] we address the metric reconstruction problem of such filamentary structures from data sampled around them. We prove that they can be approximated, with respect to the Gromov-Hausdorff distance by well-chosen Reeb graphs (and some of their variants) and we provide an efficient and easy to implement algorithm to compute such approximations in almost linear time. We illustrate the performances of our algorithm on a few synthetic and real data sets.

6.2.9. Analysis and Visualization of Maps Between Shapes

Participants: Frédéric Chazal, Maks Ovsjanikov.

In collaboration with L. Guibas (Stanford University), M. Ben Chen (Technion).

In this work we propose a method for analyzing and visualizing individual maps between shapes, or collections of such maps [23]. Our method is based on isolating and highlighting areas where the maps induce significant distortion of a given measure in a multi-scale way. Unlike the majority of prior work which focuses on discovering maps in the context of shape matching, our main focus is on evaluating, analyzing and visualizing a given map, and the distortion(s) it introduces, in an efficient and intuitive way. We are motivated primarily by the fact that most existing metrics for map evaluation are quadratic and expensive to compute in practice, and that current map visualization techniques are suitable primarily for global map understanding, and typically do not highlight areas where the map fails to meet certain quality criteria in a multi-scale way. We propose to address these challenges in a unified way by considering the functional representation of a map, and performing spectral analysis on this representation. In particular, we propose a simple multi-scale method for map evaluation and visualization, which provides detailed multi-scale information about the distortion induced by a map, which can be used alongside existing global visualization techniques.

6.2.10. Map-Based Exploration of Intrinsic Shape Differences and Variability

Participants: Frédéric Chazal, Maks Ovsjanikov.

In collaboration with L. Guibas and Raif Rustamov (Stanford University), M. Ben Chen and O. Azencot (Technion).

We develop a novel formulation for the notion of shape differences, aimed at providing detailed information about the location and nature of the differences or distortions between the two shapes being compared [27]. Our difference operator, derived from a shape map, is much more informative than just a scalar global shape similarity score, rendering it useful in a variety of applications where more refined shape comparisons are necessary. The approach is intrinsic and is based on a linear algebraic framework, allowing the use of many common linear algebra tools (e.g, SVD, PCA) for studying a matrix representation of the operator. Remarkably, the formulation allows us not only to localize shape differences on the shapes involved, but also to compare shape differences across pairs of shapes, and to analyze the variability in entire shape collections based on the differences between the shapes. Moreover, while we use a map or correspondence to define each shape difference, consistent correspondences between the shapes are not necessary for comparing shape differences, although they can be exploited if available. We give a number of applications of shape differences, including parameterizing the intrinsic variability in a shape collection, exploring shape collections using local variability at different scales, performing shape analogies, and aligning shape collections.

6.2.11. An operator Approach to Tangent Vector Field Processing

Participants: Frédéric Chazal, Maks Ovsjanikov.

In collaboration with M. Ben Chen and O. Azencot (Technion).

In this work [34], we introduce a novel coordinate-free method for manipulating and analyzing vector fields on discrete surfaces. Unlike the commonly used representations of a vector field as an assignment of vectors to the faces of the mesh, or as real values on edges, we argue that vector fields can also be naturally viewed as operators whose domain and range are functions defined on the mesh. Although this point of view is common in differential geometry it has so far not been adopted in geometry processing applications. We recall the theoretical properties of vector fields represented as operators, and show that composition of vector fields with other functional operators is natural in this setup. This leads to the characterization of vector field properties through commutativity with other operators such as the Laplace-Beltrami and symmetry operators, as well as to a straight-forward definition of differential properties such as the Lie derivative. Finally, we demonstrate a range of applications, such as Killing vector field design, symmetric vector field estimation and joint design on multiple surfaces.

6.3. Data Structures and Robust Geometric Computation

6.3.1. The Stability of Delaunay triangulations

Participants: Jean-Daniel Boissonnat, Ramsay Dyer.

In collaboration with Arijit Ghosh (Indian Statistical Institute)

We introduce a parametrized notion of genericity for Delaunay triangulations which, in particular, implies that the Delaunay simplices of δ-generic point sets are thick [45]. Equipped with this notion, we study the stability of Delaunay triangulations under perturbations of the metric and of the vertex positions. We quantify the magnitude of the perturbations under which the Delaunay triangulation remains unchanged.

6.3.2. Delaunay Stability via Perturbations

Participants: Jean-Daniel Boissonnat, Ramsay Dyer.

In collaboration with Arijit Ghosh (Indian Statistical Institute)

We present an algorithm that takes as input a finite point set in Euclidean space, and performs a perturbation that guarantees that the Delaunay triangulation of the resulting perturbed point set has quantifiable stability with respect to the metric and the point positions [43]. There is also a guarantee on the quality of the simplices: they cannot be too flat. The algorithm provides an alternative tool to the weighting or refinement methods to remove poorly shaped simplices in Delaunay triangulations of arbitrary dimension, but in addition it provides a guarantee of stability for the resulting triangulation.

6.3.3. Deletions in 3D Delaunay Triangulation

Participant: Olivier Devillers.

In collaboration with Kevin Buchin (Technical University Eindhoven, The Netherlands), Wolfgang Mulzer (Freie Universität Berlin, Germany), Okke Schrijvers, (Stanford University, USA) and Jonathan Shewchuk (University of California at Berkeley, USA)

Deleting a vertex in a Delaunay triangulation is much more difficult than inserting a new vertex because the information present in the triangulation before the deletion is difficult to exploit to speed up the computation of the new triangulation.

The removal of the tetrahedra incident to the deleted vertex creates a hole in the triangulation that need to be retriangulated. First we propose a technically sound framework to compute incrementally a triangulation of the hole vertices: *the conflict Delaunay triangulation*. The conflict Delaunay triangulation matches the hole boundary and avoid to compute extra tetrahedra outside the hole. Second, we propose a method that uses *guided randomized reinsertion* to speed up the point location during the computation of the conflict triangulation. The hole boundary is a polyhedron, this polyhedron is simplified by deleting its vertices one by one in a random order maintaining a polyhedron called *link Delaunay triangulation*, then the points are inserted in reverse order into the conflict Delaunay triangulation using the information from the link Delaunay triangulation to avoid point location [30].

6.3.4. A Convex Body with a Chaotic Random Polytope

Participants: Olivier Devillers, Marc Glisse, Rémy Thomasse.

Consider a sequence of points in a convex body in dimension d whose convex hull is dynamically maintained when the points are inserted one by one, the convex hull size may increase, decrease, or being constant when a new point is added. Studying the expected size of the convex hull when the points are evenly distributed in the convex is a classical problem of probabilistic geometry that yields to some surprising facts. For example, although it seems quite natural to think that the expected size of the convex hull is increasing with n the number of points, this fact is only formally proven for n big enough [16]. The asymptotic behavior of the expected size is known to be logarithmic for a polyhedral body and polynomial for a smooth one. If for a polyhedral or a smooth body, the asymptotic behavior is *somehow* "nice" it is possible to construct strange convex objects that have no such nice behaviors and we exhibit a convex body, such that the behavior of the expected size of a random polytope oscillates between the polyhedral and smooth behaviors when n increases [51].

6.3.5. Delaunay Triangulations and Cycles on Closed Hyperbolic surfaces

Participants: Mikhail Bogdanov, Monique Teillaud.

This work [40] is motivated by applications of *periodic* Delaunay triangulations in the Poincaré disk conformal model of the hyperbolic plane $\mathbb{H}^2$. A periodic triangulation is defined by an infinite point set that is the image of a finite point set by a (non commutative) discrete group G generated by hyperbolic translations, such that the hyperbolic area of a Dirichlet region is finite (i.e., a cocompact Fuchsian group acting on $\mathbb{H}^2$ without fixed points).

We consider the projection of such a Delaunay triangulation onto the closed orientable hyperbolic surface $M = \mathbb{H}^2/G$. The graph of its edges may have cycles of length one or two. We prove that there always exists a finite-sheeted covering space of M in which there is no cycle of length ≤ 2. We then focus on the group defining the Bolza surface (homeomorphic to a torus having two handles), and we explicitly construct a sequence of subgroups of finite index allowing us to exhibit a covering space of the Bolza surface in which, for any input point set, there is no cycle of length one, and another covering space in which there is no cycle of length two. We also exhibit a small point set such that the projection of the Delaunay triangulation on the Bolza surface for any superset has no cycle of length ≤ 2.

The work uses mathematical proofs, algorithmic constructions, and implementation.

6.3.6. Universal Point Sets for Planar Graph Drawings with Circular Arcs

Participant: Monique Teillaud.

In collaboration with Patrizio Angelini (Roma Tre University), David Eppstein (University of California, Irvine), Fabrizio Frati (The University of Sydney), Michael Kaufmann (MPI, Tübingen), Sylvain Lazard (EPI VEGAS*), Tamara Mchedlidze (Karlsruhe Institute of Technology), and Alexander Wolff (Universität Würzburg).*

We prove that there exists a set S of n points in the plane such that every n-vertex planar graph G admits a plane drawing in which every vertex of G is placed on a distinct point of S and every edge of G is drawn as a circular arc. [25]

6.3.7. A Generic Implementation of dD Combinatorial Maps in CGAL

Participant: Monique Teillaud.

In collaboration with Guillaume Damiand (Université de Lyon, LIRIS, UMR 5205 CNRS)

We present a generic implementation of dD combinatorial maps and linear cell complexes in CGAL. A combinatorial map describes an object subdivided into cells; a linear cell complex describes the linear geometry embedding of such a subdivision. In this paper [49], we show how generic programming and new techniques recently introduced in the C++11 standard allow a fully generic and customizable implementation of these two data structures, while maintaining optimal memory footprint and direct access to all information. To the best of our knowledge, the CGAL software packages presented here [59], [60] offer the only available generic implementation of combinatorial maps in any dimension.

6.3.8. Silhouette of a Random Polytope

Participant: Marc Glisse.

In collaboration with Sylvain Lazard and Marc Pouget (EPI VEGAS*) and Julien Michel (LMA-Poitiers).*

We consider random polytopes defined as the convex hull of a Poisson point process on a sphere in $\mathbb{R}^3$ such that its average number of points is n. We show [52] that the expectation over all such random polytopes of the maximum size of their silhouettes viewed from infinity is $\Theta(\sqrt{n})$.

6.3.9. A New Approach to Output-Sensitive Voronoi Diagrams and Delaunay Triangulations

Participant: Donald Sheehy.

In collaboration with Gary Miller (Carnegie Mellon University)

We describe [35] a new algorithm for computing the Voronoi diagram of a set of n points in constant-dimensional Euclidean space. The running time of our algorithm is $O(f \log n \log \Delta)$ where f is the output complexity of the Voronoi diagram and Δ is the spread of the input, the ratio of largest to smallest pairwise distances. Despite the simplicity of the algorithm and its analysis, it improves on the state of the art for all inputs with polynomial spread and near-linear output size. The key idea is to first build the Voronoi diagram of a superset of the input points using ideas from Voronoi refinement mesh generation. Then, the extra points are removed in a straightforward way that allows the total work to be bounded in terms of the output complexity, yielding the output sensitive bound. The removal only involves local flips and is inspired by kinetic data structures.

6.3.10. A Fast Algorithm for Well-Spaced Points and Approximate Delaunay Graphs

Participant: Donald Sheehy.

In collaboration with Gary Miller and Ameya Velingker (Carnegie Mellon University)

We present [32] a new algorithm that produces a well-spaced superset of points conforming to a given input set in any dimension with guaranteed optimal output size. We also provide an approximate Delaunay graph on the output points. Our algorithm runs in expected time $O(2^{O(d)}(n \log n + m))$, where n is the input size, m is the output point set size, and d is the ambient dimension. The constants only depend on the desired element quality bounds.

To gain this new efficiency, the algorithm approximately maintains the Voronoi diagram of the current set of points by storing a superset of the Delaunay neighbors of each point. By retaining quality of the Voronoi diagram and avoiding the storage of the full Voronoi diagram, a simple exponential dependence on d is obtained in the running time. Thus, if one only wants the approximate neighbors structure of a refined Delaunay mesh conforming to a set of input points, the algorithm will return a size $2^{O(d)}m$ graph in $2^{O(d)}(n \log n + m)$ expected time. If m is superlinear in n, then we can produce a hierarchically well-spaced superset of size $2^{O(d)}n$ in $2^{O(d)}n \log n$ expected time.

6.3.11. Geometric Separators and the Parabolic Lift

Participant: Donald Sheehy.

A geometric separator for a set U of n geometric objects (usually balls) is a small (sublinear in n) subset whose removal disconnects the intersection graph of U into roughly equal sized parts. These separators provide a natural way to do divide and conquer in geometric settings. A particularly nice geometric separator algorithm originally introduced by Miller and Thurston has three steps: compute a centerpoint in a space of one dimension higher than the input, compute a conformal transformation that "centers" the centerpoint, and finally, use the computed transformation to sample a sphere in the original space. The output separator is the subset of S intersecting this sphere. It is both simple and elegant. We show [36] that a change of perspective (literally) can make this algorithm even simpler by eliminating the entire middle step. By computing the centerpoint of the points lifted onto a paraboloid rather than using the stereographic map as in the original method, one can sample the desired sphere directly, without computing the conformal transformation.

GRACE Project-Team

6. New Results

6.1. Diffusion layers for block ciphers

MDS matrices allow the construction of optimal linear diffusion layers in block ciphers. However, MDS matrices usually have a large description (for example, they can never be sparse), and this results in costly software/hardware implementations. We can solve this problem using *recursive MDS matrices*, which can be computed as a power of a simple companion matrix—and thus have a compact description suitable for constrained environments. Until now, finding recursive MDS matrices required an exhaustive search on families of companion matrices; this clearly limited the size of MDS matrices that one could look for. We have found a new direct construction, based on shortened BCH codes, which allows us to efficiently construct these matrices for arbitrary parameter sizes.

6.2. Rank metric codes over the rationals

Rank metric and Gabidulin codes over the rationals promise interesting applications to space-time coding. We have constructed optimal codes similar to Gabidulin codes but with complex coefficients, using number fields and Galois automorphisms. Using these codes we can completely bypass intermediate constructions using finite fields, which were the stumbling-block in classic constructions.

6.3. Cryptanalysis of McEliece cryptosystems based on Generalised Reed–Solomon codes

The McEliece encryption scheme based on binary Goppa codes was one of the first public-key encryption schemes [24]. Niederreiter [25] dramatically reduced the (huge) key size—a major problem with McEliece's original proposal—using Generalised Reed–Solomon (GRS) codes, but his modified scheme was broken by Sidelnikov and Shestakov [26] in 1992. There have been several attempts at repairing these smaller-key McEliece schemes. In collaboration with P. Gaborit, V. Gautier, A. Otmani and J.-P. Tillich, Alain Couvreur found polynomial time attacks on these schemes using the distinguishability of GRS codes from random codes.

6.4. New Identities relating Goppa codes

Goppa codes are strongly related to AG codes based on curves of genus 0. Among other applications, these codes are very famous for their cryptographic potential: they are one of the very few families of algebraic codes proposed for the McEliece encryption scheme which have not been broken up to now. At least for this reason, getting further knowledge on the structure of such codes is of interest. In [19], Alain Couvreur, A. Otmani and J.-P. Tillich proved a new identity yielding many improvements in the designed parameters of Goppa codes.

6.5. Root finding algorithms over local rings

Guillaume Quintin, in collaboration with J. Berthomieu and G. Lecerf, has developed new algorithms computing the roots of polynomials over complete local unramified rings [7]; this is important in the second stage of Guruswami–Sudan list decoding algorithms for codes over finite rings. Quintin has implemented these algorithms in MATHEMAGIX, using his FINITEFIELDZ and QUINTIX librairies.

6.6. Codes over rings

M. Barbier, C. Chabot and Guillaume Quintin proposed a new description for quasi–cyclic codes using the ring of matrices with polynomial entries, thus defining the new class of *quasi-BCH* codes. Guillaume Quintin proved that these codes can be regarded as interleaved subcodes of Reed–Solomon codes; this allowed them to define a polynomial-time decoding algorithm for quasi-BCH codes. Guillaume Quintin also generalized list decoding algorithms to codes over non commutative rings [8].

6.7. Quantum LDPC codes

For some time it was feared that quantum computers could not be built because of distortions of quantum states due to interaction with the environment. This issue could be addressed by the use of quantum codes. *Quantum LDPC codes* are very interesting candidates here, because their very fast decoding algorithm allows high error correction rates. But the design of good quantum LDPC codes is far more complicated than for their classical counterparts, and cannot be done by random generation. The best-known constructions come from algebraic topology and simplicial homology, but their limits were unknown. Nicolas Delfosse used Riemannian geometry theorems of Gromov to prove that an $[[n, k, d]]$-quantum code constructed from the homology of a simplicial surface satisfies $kd^2 \leq C(\log k)^2 n$ for some constant C [21].

Color codes are quantum LDPC codes constructed from 3–regular surface tilings whose set of faces is 3–colorable. Delfosse used morphisms of chain complexes to prove that the decoding of a color code can be reduced to the decoding of three associated surface codes; hence, every decoding algorithm for surface codes yields a decoding algorithm for color codes. From this result, Delfosse obtained theoretical lower bounds on the error threshold of a family of color codes [20].

6.8. New families of fast elliptic curves

Benjamin Smith has pioneered the use of mod-p reductions of Q-curves to produce elliptic curves with efficient scalar multiplication algorithms—which translates into faster encryption, decryption, signing, and signature verification operations on these curves. A theoretical article was presented at ASIACRYPT 2013 [9], and the Journal of Cryptology has invited the submission of a longer version. The theory was put into practice in collaboration with Craig Costello (Microsoft Research) and Huseyin Hisil (Yasar University). Their resulting publicly available implementation, which represents the state of the art in constant-time (side-channel conscious) elliptic curve scalar multiplication on 64-bit Intel platforms at the 128-bit security level, can carry out a constant-time scalar multiplication in 145k cycles on Ivy Bridge architectures. This work will appear in EUROCRYPT 2014 [17].

6.9. Tensor rank of multiplication over finite fields

Determining the tensor rank of multiplication over finite fields is a problem of great interest in algebraic complexity theory, but it also has practical importance: it allows us to obtain multiplication algorithms with a low bilinear complexity, which are of crucial significance in cryptography. In collaboration with S. Ballet and J. Chaumine [12], Julia Pieltant obtained new asymptotic bounds for the symmetric tensor rank of multiplication in finite extensions of finite fields $\mathbb{F}_q$. In the more general (not-necessarily-symmetric) case, Pieltant and H. Randriam obtained new uniform upper bounds for multiplication in extensions of $\mathbb{F}_q$. They also gave purely asymptotic bounds substantially improving those coming from uniform bounds, by using a family of Shimura curves defined over $\mathbb{F}_q$. This work will appear in Mathematics of Computation [22].

LFANT Project-Team

6. New Results

6.1. Class groups and other invariants of number fields

Participants: Karim Belabas, Jean-Paul Cerri, Pierre Lezowski.

In collaboration with E. Friedman, K. Belabas presented in [22] a new algorithm to compute the residue at $s = 1$ of the Dedekind zeta function of a number field, conditional on GRH. This improves on previous results of Eric Bach [31] by a useful constant factor. Such an estimate is one of the two key analytic ingredients to Buchmann's class group algorithm, the other being the existence (under GRH) of an explicit set of small generators [33].

In collaboration with F. Thorne, H. Cohen worked on Dirichlet series associated to cubic and quartic fields with given resolvent. In [23] they give an explicit formula for the Dirichlet series $\sum_K |\Delta(K)|^{-s}$, where the sum is over isomorphism classes of all cubic fields whose quadratic resolvent field is isomorphic to a fixed quadratic field k. This is a sequel to previous work of Cohen and Morra, where such formulæ are proved in a more general setting, in terms of sums over characters of certain groups related to ray class groups. Here, the analysis is carried further and they prove explicit formulæ for these Dirichlet series over $\mathbb{Q}$. As an application, they compute tables of the number of S_3-sextic fields K with discriminant ranging up to 10^{23}. An accompanying PARI/GP implementation is available.

In [24], they give an explicit formula for the Dirichlet series $\sum_K |\Delta(K)|^{-s}$, where this time the sum is over isomorphism classes of all quartic fields whose cubic resolvent field is isomorphic to a fixed cubic field k. This work is a sequel to an unpublished preprint of Cohen, Diaz y Diaz, and Olivier.

The papers by H. Cohen on Haberland's formula and numerical computation of Petersson scalar products and by A. Angelakis and P. Stevenhagen on imaginary quadratic fields with isomorphic abelian Galois groups, which were presented at the ANTS-X conference, were published in [17], [16].

6.2. Number and function fields

Participants: Athanasios Angelakis, Jean-Marc Couveignes, Karim Belabas.

In collaboration with Reynald Lercier, Jean-Marc Couveignes presents in [12] a randomised algorithm that on input a finite field K with q elements and a positive integer d outputs a degree d irreducible polynomial in $K[x]$. The running time is $d^{1+o(1)} \times (\log q)^{5+o(1)}$ elementary operations. The $o(1)$ in $d^{1+o(1)}$ is a function of d that tends to zero when d tends to infinity. And the $o(1)$ in $(\log q)^{5+o(1)}$ is a function of q that tends to zero when q tends to infinity. In particular, the complexity is quasi-linear in the degree d.

The book of surveys "Explicit methods in number theory. Rational points and Diophantine equations" [19] edited by K. Belabas with contributions from K. Belabas, F. Beukers, P. Gaudry, W. McCallum, B. Poonen, S. Siksek, M. Stoll and M. Watkins presents the state of the art of the use of explicit methods in arithmetic geometry to solve diophantine problems.

6.3. Quaternion algebras

Participants: Jean-Paul Cerri, Pierre Lezowski, Aurel Page.

In a joint work with J. Chaubert ([11]), J.-P. Cerri and P. Lezowski have studied totally definite quaternion fields over number fields which are Euclidean, that is to say that they admit a left or right Euclidean order. In particular, they have established the complete list of totally definite and Euclidean quaternion fields over real quadratic number fields. In this list, all fields are in fact norm-Euclidean. The proofs are both theoretic and algorithmic.

A. Page uploaded a new version of his article [30] on the computation of arithmetic Kleinian groups, incorporating comments from the referee.

6.4. Complex multiplication and modularity

Participants: Jean-Marc Couveignes, Andreas Enge, Nicolas Mascot, Enea Milio, Aurel Page, Damien Robert.

H. Ivey-Law has been implementing efficient algorithms to compute Hilbert class polynomials and modular polynomials for various modular functions, as well as various supplementary algorithms required by, or based on, these two primary components. These algorithms form an important and time-critical part of algorithms used to select elliptic curves for use in cryptographic applications.

The implementation is based on algorithms for these tasks published by A. Sutherland and his collaborators. It includes, more specifically, algorithms to compute Hilbert class polynomials for various different modular functions over $\mathbb{Z}$ or $\mathbb{Z}/M\mathbb{Z}$, modular polynomials for various different modular functions over $\mathbb{Z}$, $\mathbb{Z}/M\mathbb{Z}$, and/or pre-instantiated at a particular point. The supplementary algorithms include functionality for computing equations for isogenies between elliptic curves and equations for their codomains, for manipulating, interrogating and traversing isogeny volcanoes, for computing minimal polycyclic presentations of abstract groups, for testing supersingularity of j-invariants, for accessing optimised equations of the modular curve $X_1(N)$ for $N \leq 50$, for finding elliptic curves with a given trace or a given endomorphism ring, for calculating the endomorphism ring of a given elliptic curve, for computing the action of the torsor $\mathrm{Cl}(\mathcal{O})$ on the set of elliptic curves with endomorphism ring $\mathcal{O}$ and for enumerating the kernel of the map $\mathrm{Cl}(\mathbb{Z} + N\mathcal{O}) \to \mathrm{Cl}(\mathcal{O})$.

These algorithms are implemented in an experimental branch of PARI/GP, and will be integrated in the public version soon.

A. Enge and R. Schertz determine in [13] under which conditions singular values of multiple η-quotients of square-free level, not necessarily prime to 6, yield class invariants, that is, algebraic numbers in ring class fields of imaginary-quadratic number fields. It turns out that the singular values lie in subfields of the ring class fields of index $2^{k'-1}$ when $k' \geq 2$ primes dividing the level are ramified in the imaginary-quadratic field, which leads to faster computations of elliptic curves with prescribed complex multiplication. The result is generalised to singular values of modular functions on $X_0^+(p)$ for p prime and ramified.

The paper of R. Cosset and D. Robert [25] presenting an algorithm for computing isogenies between principally polarised abelian surface has been accepted for publication in Mathematics of Computation. This paper explains, given the theta coordinates of the points of a maximal isotropic kernel of the ℓ-torsion, how to compute the corresponding isogeny. It also gives formulæ for the conversion between theta coordinates and Mumford coordinates.

The paper by K. Lauter and D. Robert on Improved CRT Algorithm for Class Polynomials in Genus 2, which was presented at the ANTS-X conference, was published in [18].

A. Enge and E. Thomé describe in [14] a quasi-linear algorithm for computing Igusa class polynomials of Jacobians of genus 2 curves via complex floating-point approximations of their roots. After providing an explicit treatment of the computations in quartic CM fields and their Galois closures, they pursue an approach due to Dupont for evaluating ϑ-constants in quasi-linear time using Newton iterations on the Borchardt mean. They report on experiments with the implementation CMH and present an example with class number 20016.

N. Mascot's article on computing modular Galois representations [15] has been published in Rendiconti del Circolo Matematico di Palermo. This article describes an algorithm to compute Galois representations attached to a newform, and to deduce the Fourier coefficients of this newform modulo a small prime.

E. Milio has implemented R. Dupont's algorithms [38] in PARI/GP. With them, he has calculated the three modular polynomials in genus 2 and level 2 defined by Streng's version of Igusa modular forms and a modular polynomial of genus 2 and level 3 coming from theta modular forms.

6.5. Elliptic curve cryptology

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

Couveignes and Lercier study in [26] the problem of parameterisations by radicals of low genus algebraic curves. They prove that for q a prime power that is large enough and prime to 6, a fixed positive proportion of all genus 2 curves over the field with q elements can be parameterised by 3-radicals. This results in the existence of a deterministic encoding into these curves when q is congruent to 2 modulo 3. Deterministic encodings into curves are useful in numerous situations, for instance in discrete logarithm cryptography. The parameterisation found by Couveignes and Lercier is in some sense the first generic one for genus 2 curves.

A software for this method is in preparation.

The survey [21], published in the *Handbook of Finite Fields*, presents the state of the art of the use of elliptic curves in cryptography.

6.6. Pairings

Participants: Andreas Enge, Damien Robert.

In [27], A. Enge gives an elementary and self-contained introduction to pairings on elliptic curves over finite fields. For the first time in the literature, the three different definitions of the Weil pairing are stated correctly and proved to be equivalent using Weil reciprocity. Pairings with shorter loops, such as the ate, ate$_i$, R-ate and optimal pairings, together with their twisted variants, are presented with proofs of their bilinearity and non-degeneracy. Finally, different types of pairings are reviewed in a cryptographic context. The article can be seen as an update chapter to [40].

With D. Lubicz, D. Robert has worked on extending the algorithm to compute Weil and Tate pairings using theta functions from [42] to the ate and optimal ate pairings in [29]. The result includes how to compute the Miller functions with theta functions, but also how to generalise ate and optimal ate pairings to Kummer varieties. In contrast to preceding algorithms using Miller functions which needed a geometric interpretation of the addition law and worked with Jacobians, this new algorithm uses only the algebraic Riemann relations and works on any abelian variety (provided with a theta structure). This algorithm has been implemented using AVIsogenies.

POLSYS Project-Team

6. New Results

6.1. Fundamental Algorithms and Structured Systems

6.1.1. Structured polynomial systems: the quasi-homogeneous case

Let $\mathbb{K}$ be a field and $(f_1, ..., f_n) \subset \mathbb{K}[X_1, ..., X_n]$ be a sequence of quasi-homogeneous polynomials of respective weighted degrees $(d_1, ..., d_n)$ w.r.t a system of weights $(w_1, \cdots, w_n)$. Such systems are likely to arise from a lot of applications, including physics or cryptography. In [29], we design strategies for computing Gröbner bases for quasi-homogeneous systems by adapting existing algorithms for homogeneous systems to the quasi-homogeneous case. Overall, under genericity assumptions, we show that for a generic zero-dimensional quasi-homogeneous system, the complexity of the full strategy is polynomial in the weighted Bézout bound $\prod_{i=1}^{n} d_i / \prod_{i=1}^{n} w_i$. We provide some experimental results based on generic systems as well as systems arising from a cryptography problem. They show that taking advantage of the quasi-homogeneous structure of the systems allow us to solve systems that were out of reach otherwise.

6.1.2. Structured polynomial systems: the determinantal case

In [13], We study the complexity of solving the *generalized MinRank problem*, i.e. computing the set of points where the evaluation of a polynomial matrix has rank at most r. A natural algebraic representation of this problem gives rise to a *determinantal ideal*: the ideal generated by all minors of size $r + 1$ of the matrix. We give new complexity bounds for solving this problem using Gröbner bases algorithms under genericity assumptions on the input matrix. In particular, these complexity bounds allow us to identify families of generalized MinRank problems for which the arithmetic complexity of the solving process is polynomial in the number of solutions. We also provide an algorithm to compute a rational parametrization of the variety of a 0-dimensional and radical system of bi-degree $(D, 1)$. We show that its complexity can be bounded by using the complexity bounds for the generalized MinRank problem.

6.1.3. On the Complexity of the Generalized MinRank Problem

In [13] we study the complexity of solving the *generalized MinRank problem*, i.e. computing the set of points where the evaluation of a polynomial matrix has rank at most r. A natural algebraic representation of this problem gives rise to a *determinantal ideal*: the ideal generated by all minors of size $r + 1$ of the matrix. We give new complexity bounds for solving this problem using Gröbner bases algorithms under genericity assumptions on the input matrix. In particular, these complexity bounds allow us to identify families of generalized MinRank problems for which the arithmetic complexity of the solving process is polynomial in the number of solutions. We also provide an algorithm to compute a rational parametrization of the variety of a 0-dimensional and radical system of bi-degree $(D, 1)$. We show that its complexity can be bounded by using the complexity bounds for the generalized MinRank problem.

6.1.4. On the Complexity of Computing Gröbner Bases for Quasi-homogeneous Systems

Let $\mathbb{K}$ be a field and $(f_1, ..., f_n) \subset \mathbb{K}[X_1, ..., X_n]$ be a sequence of quasi-homogeneous polynomials of respective weighted degrees $(d_1, ..., d_n)$ w.r.t a system of weights $(w_1, \cdots, w_n)$. Such systems are likely to arise from a lot of applications, including physics or cryptography.

In [29], we design strategies for computing Gröbner bases for quasi-homogeneous systems by adapting existing algorithms for homogeneous systems to the quasi-homogeneous case. Overall, under genericity assumptions, we show that for a generic zero-dimensional quasi-homogeneous system, the complexity of the full strategy is polynomial in the weighted Bézout bound $\prod_{i=1}^{n} d_i / \prod_{i=1}^{n} w_i$.

We provide some experimental results based on generic systems as well as systems arising from a cryptography problem. They show that taking advantage of the quasi-homogeneous structure of the systems allow us to solve systems that were out of reach otherwise.

6.1.5. Gröbner bases of ideals invariant under a commutative group : the non-modular case

In [30], we propose efficient algorithms to compute the Gröbner basis of an ideal $I \subset k[x_1, \cdots, x_n]$ globally invariant under the action of a commutative matrix group G, in the non-modular case (where $char(k)$ doesn't divide $|G|$). The idea is to simultaneously diagonalize the matrices in G, and apply a linear change of variables on I corresponding to the base-change matrix of this diagonalization. We can now suppose that the matrices acting on I are diagonal. This action induces a grading on the ring $R = k[x_1, \cdots, x_n]$, compatible with the degree, indexed by a group related to G, that we call G-degree. The next step is the observation that this grading is maintained during a Gröbner basis computation or even a change of ordering, which allows us to split the Macaulay matrices into $|G|$ submatrices of roughly the same size. In the same way, we are able to split the canonical basis of R/I (the staircase) if I is a zero-dimensional ideal. Therefore, we derive *abelian* versions of the classical algorithms F_4, F_5 or FGLM. Moreover, this new variant of F_4/F_5 allows complete parallelization of the linear algebra steps, which has been successfully implemented. On instances coming from applications (NTRU crypto-system or the Cyclic-n problem), a speed-up of more than 400 can be obtained. For example, a Gröbner basis of the Cyclic-11 problem can be solved in less than 8 hours with this variant of F_4. Moreover, using this method, we can identify new classes of polynomial systems that can be solved in polynomial time.

6.1.6. Signature Rewriting in Gröbner Basis Computation

In [27] we introduce the RB algorithm for Gröbner basis computation, a simpler yet equivalent algorithm to F5GEN. RB contains the original unmodified F5 algorithm as a special case, so it is possible to study and understand F5 by considering the simpler RB. We present simple yet complete proofs of this fact and of F5's termination and correctness. RB is parametrized by a rewrite order and it contains many published algorithms as special cases, including SB. We prove that SB is the best possible instantiation of RB in the following sense. Let X be any instantiation of RB (such as F5). Then the S-pairs reduced by SB are always a subset of the S-pairs reduced by X and the basis computed by SB is always a subset of the basis computed by X.

6.1.7. An analysis of inhomogeneous signature-based Gröbner basis computations

In [8] we give an insight into the behaviour of signature-based Gröbner basis algorithms, like F5, G2V or SB, for inhomogeneous input. On the one hand, it seems that the restriction to sig-safe reductions puts a penalty on the performance. The lost connection between polynomial degree and signature degree can disallow lots of reductions and can lead to an overhead in the computations. On the other hand, the way critical pairs are sorted and corresponding s-polynomials are handled in signature- based algorithms is a very efficient one, strongly connected to sorting w.r.t. the well-known sugar degree of polynomials.

6.1.8. Improving incremental signature-based Gröbner basis algorithms

In [9] we describe a combination of ideas to improve incremental signature-based Gröbner basis algorithms having a big impact on their performance. Besides explaining how to combine already known optimizations to achieve more efficient algorithms, we show how to improve them even more. Although our idea has a positive affect on all kinds of incremental signature-based algorithms, the way this impact is achieved can be quite different. Based on the two best-known algorithms in this area, F5 and G2V, we explain our idea, both from a theoretical and a practical point of view.

6.1.9. A new algorithmic scheme for computing characteristic sets

Ritt-Wu's algorithm of characteristic sets is the most representative for triangularizing sets of multivariate polynomials. Pseudo-division is the main operation used in this algorithm. In [18] we present a new algorithmic scheme for computing generalized characteristic sets by introducing other admissible reductions than pseudo-division. A concrete subalgorithm is designed to triangularize polynomial sets using selected admissible reductions and several effective elimination strategies and to replace the algorithm of basic sets (used in Ritt-Wu's algorithm). The proposed algorithm has been implemented and experimental results show that it

performs better than Ritt-Wu's algorithm in terms of computing time and simplicity of output for a number of non-trivial test examples

6.2. Solving Systems over the Reals and Applications

6.2.1. On the Boolean complexity of real root refinement

In [32] 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 $r \leq d$.

6.2.2. On the minimum of a polynomial function on a basic closed semialgebraic set and applications

In [17] we give an explicit upper bound for the algebraic degree and an explicit lower bound for the absolute value of the minimum of a polynomial function on a compact connected component of a basic closed semialgebraic set when this minimum is not zero. We also present extensions of these results to non-compact situations. As an application, we obtain a lower bound for the separation of two disjoint connected components of basic closed semialgebraic sets, when at least one of them is compact.

6.2.3. Rational solutions to Linear Matrix Inequalities and Sums of Squares

Consider a $(D \times D)$ symmetric matrix A whose entries are linear forms in $\mathbb{Q}[X_1, ..., X_k]$ with coefficients of bit size $\leq \tau$. In [31], we provide an algorithm which decides the existence of rational solutions to the linear matrix inequality $\mathsf{A} \succeq 0$ and outputs such a rational solution if it exists. This problem is of first importance: it can be used to compute algebraic certificates of positivity for multivariate polynomials. Our algorithm runs within $(k\tau)^{O(1)}2^{O(\min(k,D)D^2)}D^{O(D^2)}$ bit operations; the bit size of the output solution is dominated by $\tau^{O(1)}2^{O(\min(k,D)D^2)}$. These results are obtained by designing algorithmic variants of constructions introduced by Klep and Schweighofer. This leads to the best complexity bounds for deciding the existence of sums of squares with rational coefficients of a given polynomial. We have implemented the algorithm; it has been able to tackle Scheiderer's example of a multivariate polynomial that is a sum of squares over the reals but not over the rationals; providing the first computer validation of this counter-example to Sturmfels' conjecture.

6.2.4. Exact Voronoi diagram of smooth convex pseudo-circles: General predicates, and implementation for ellipses

In [10] we examine the problem of computing exactly the Voronoi diagram (via the dual Delaunay graph) of a set of, possibly intersecting, smooth convex pseudo-cirlces in the Euclidean plane, given in parametric form. Pseudo-circles are (convex) sites, every pair of which has at most two intersecting points. The Voronoi diagram is constructed incrementally. Our first contribution is to propose robust and efficient algorithms, under the exact computation paradigm, for all required predicates, thus generalizing earlier algorithms for non-intersecting ellipses. Second, we focus on INCIRCLE, which is the hardest predicate, and express it by a simple sparse 5×5 polynomial system, which allows for an efficient implementation by means of successive Sylvester resultants and a new factorization lemma. The third contribution is our CGAL-based C++ software for the case of possibly intersecting ellipses, which is the first exact implementation for the problem. Our code spends about a minute to construct the Voronoi diagram of 200 ellipses, when few degeneracies occur. It is faster than the CGAL segment Voronoi diagram, when ellipses are approximated by k-gons for $k > 15$, and a state-of-the-art implementation of the Voronoi diagram of points, when each ellipse is approximated by more than 1250 points.

6.2.5. Patience of Matrix Games

In [15], for matrix games we study how small nonzero probability must be used in optimal strategies. We show that for $n \times n$ win-lose-draw games (i.e. $(-1, 0, 1)$ matrix games) nonzero probabilities smaller than $n^{-O(n)}$ are never needed. We also construct an explicit $n \times n$ win-lose game such that the unique optimal strategy uses a nonzero probability as small as $n^{-\Omega(n)}$. This is done by constructing an explicit $(-1, 1)$ nonsingular $n \times n$ matrix, for which the inverse has only nonnegative entries and where some of the entries are of value $n^{\Omega(n)}$.

6.2.6. A polynomial approach for extracting the extrema of a spherical function and its application in diffusion MRI

Antipodally symmetric spherical functions play a pivotal role in diffusion MRI in representing sub-voxel-resolution microstructural information of the underlying tissue. This information is described by the geometry of the spherical function. In [14] we propose a method to automatically compute all the extrema of a spherical function. We then classify the extrema as maxima, minima and saddle-points to identify the maxima. We take advantage of the fact that a spherical function can be described equivalently in the spherical harmonic (SH) basis, in the symmetric tensor (ST) basis constrained to the sphere, and in the homogeneous polynomial (HP) basis constrained to the sphere. We extract the extrema of the spherical function by computing the stationary points of its constrained HP representation. Instead of using traditional optimization approaches, which are inherently local and require exhaustive search or re-initializations to locate multiple extrema, we use a novel polynomial system solver which analytically brackets all the extrema and refines them numerically, thus missing none and achieving high precision. To illustrate our approach we consider the Orientation Distribution Function (ODF). In diffusion MRI the ODF is a spherical function which represents a state-of-the-art reconstruction algorithm whose maxima are aligned with the dominant fiber bundles. It is, therefore, vital to correctly compute these maxima to detect the fiber bundle directions. To demonstrate the potential of the proposed polynomial approach we compute the extrema of the ODF to extract all its maxima. This polynomial approach is, however, not dependent on the ODF and the framework presented in this line of work can be applied to any spherical function described in either the SH basis, ST basis or the HP basis.

6.2.7. Improving Angular Speed Uniformity by Reparameterization

In [20] we introduce the notion of angular speed uniformity as a quality measure for parameter-izations of plane curves and propose an algorithm to compute uniform reparameterizations for quadratic and cubic curves. We prove that only straight lines have uniform rational parameterizations. For any plane curve other than lines, we show how to find a rational reparameterization that has the maximum uniformity among all the rational parameterizations of the same degree. We also establish specific results for quadratic and certain cubic Bézier curves.

6.2.8. Formalization and Specification of Geometric Knowledge Objects

[7] presents our work on the identification, formalization, structuring, and specification of geometric knowledge objects for the purpose of semantic representation and knowledge management. We classify geometric knowledge according to how it has been accumulated and represented in the geometric literature, formalize geometric knowledge statements by adapting the language of first-order logic, specify knowledge objects with embedded knowledge in a retrievable and extensible data structure, and organize them by modeling the hierarchic structure of relations among them. Some examples of formal specification for geometric knowledge objects are given to illustrate our approach. The underlying idea of the approach has been used successfully for automated geometric reasoning, knowledge base creation, and electronic document generation.

6.2.9. A Framework for Improving Uniformity of Parameterizations of Curves

In [16] we define quasi-speed as a generalization of linear speed and angular speed for parameterizations of curves and use the uniformity of quasi-speed to measure the quality of the parameterizations. With such conceptual setting, a general framework is developed for studying uniformity behaviors under reparameterization via proper parameter transformation and for computing reparameterizations with improved uniformity

of quasispeed by means of optimal single-piece, C^0 piecewise, and C^1 piecewise Möbius transformations. Algorithms are described for uniformity-improved reparameterization using different Möbius transformations with different optimization techniques. Examples are presented to illustrate the concepts, the framework, and the algorithms. Experimental results are provided to validate the framework and to show the efficiency of the algorithms.

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

6.3.1. On the Complexity of Solving Quadratic Boolean Systems

A fundamental problem in computer science is to find all the common zeroes of m quadratic polynomials in n unknowns over $\mathbb{F}_2$. The cryptanalysis of several modern ciphers reduces to this problem. Up to now, the best complexity bound was reached by an exhaustive search in $4\log_2 n\, 2^n$ operations. We give an algorithm that reduces the problem to a combination of exhaustive search and sparse linear algebra. This algorithm has several variants depending on the method used for the linear algebra step. Under precise algebraic assumptions on the input system, we show in [4], that the deterministic variant of our algorithm has complexity bounded by $O(2^{0.841n})$ when $m = n$, while a probabilistic variant of the Las Vegas type has expected complexity $O(2^{0.792n})$. Experiments on random systems show that the algebraic assumptions are satisfied with probability very close to 1. We also give a rough estimate for the actual threshold between our method and exhaustive search, which is as low as 200, and thus very relevant for cryptographic applications.

6.3.2. Decomposing polynomial sets into simple sets over finite fields: The positive-dimensional case

Our work in [19] presents an algorithm for decomposing any positive-dimensional polynomial set into simple sets over an arbitrary finite field. The algorithm is based on some relationship established between simple sets and radical ideals, reducing the decomposition problem to the problem of computing the radicals of certain ideals. In addition to direct application of the algorithms of Matsumoto and Kemper, the algorithm of Fortuna and others is optimized and improved for the computation of radicals of special ideals. Preliminary experiments with an implementation of the algorithm in Maple and Singular are carried out to show the effectiveness and efficiency of the algorithm.

6.3.3. Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm

In 2004, an algorithm is introduced to solve the DLP for elliptic curves defined over a non prime finite field $\mathbb{F}_{q^n}$. One of the main steps of this algorithm requires decomposing points of the curve $E(\mathbb{F}_{q^n})$ with respect to a factor base, this problem is denoted PDP. In [11], we apply this algorithm to the case of Edwards curves, the well-known family of elliptic curves that allow faster arithmetic as shown by Bernstein and Lange. More precisely, we show how to take advantage of some symmetries of twisted Edwards and twisted Jacobi intersections curves to gain an exponential factor $2^{\omega(n-1)}$ to solve the corresponding PDP where ω is the exponent in the complexity of multiplying two dense matrices. Practical experiments supporting the theoretical result are also given. For instance, the complexity of solving the ECDLP for twisted Edwards curves defined over $\mathbb{F}_{q^5}$, with $q \approx 2^{64}$, is supposed to be $\sim 2^{160}$ operations in $E(\mathbb{F}_{q^5})$ using generic algorithms compared to 2^{130} operations (multiplication of two 32-bits words) with our method. For these parameters the PDP is intractable with the original algorithm. The main tool to achieve these results relies on the use of the symmetries and the quasi-homogeneous structure induced by these symmetries during the polynomial system solving step. Also, we use a recent work on a new algorithm for the change of ordering of Gröbner basis which provides a better heuristic complexity of the total solving process.

6.3.4. A Distinguisher for High Rate McEliece Cryptosystems

The Goppa Code Distinguishing (GD) problem consists in distinguishing the matrix of a Goppa code from a random matrix. The hardness of this problem is an assumption to prove the security of code-based cryptographic primitives such as McEliece's cryptosystem. Up to now, it is widely believed that the GD

problem is a hard decision problem. We present in [12] the first method allowing to distinguish alternant and Goppa codes over any field. Our technique can solve the GD problem in polynomial-time provided that the codes have sufficiently large rates. The key ingredient is an algebraic characterization of the key-recovery problem. The idea is to consider the rank of a linear system which is obtained by linearizing a particular polynomial system describing a key-recovery attack. Experimentally it appears that this dimension depends on the type of code. Explicit formulas derived from extensive experimentations for the rank are provided for "generic" random, alternant, and Goppa codes over any alphabet. Finally, we give theoretical explanations of these formulas in the case of random codes, alternant codes over any field of characteristic two and binary Goppa codes.

6.3.5. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic

We investigate in this paper the security of HFE and Multi-HFE schemes as well as their minus and embedding variants. Multi-HFE is a generalization of the well-known HFE schemes. The idea is to use a multivariate quadratic system instead of a univariate polynomial in HFE over an extension field as a private key. According to the authors, this should make the classical direct algebraic (message-recovery) attack proposed by Faugère and Joux on HFE no longer efficient against Multi-HFE. We consider here the hardness of the key-recovery in Multi-HFE and its variants, but also in HFE (both for odd and even characteristic). We first improve and generalize the basic key recovery proposed by Kipnis and Shamir on HFE. To do so, we express this attack as matrix/vector operations. In one hand, this permits to improve the basic Kipnis-Shamir (KS) attack on HFE. On the other hand, this allows to generalize the attack on Multi-HFE. Due to its structure, we prove that a Multi-HFE scheme has much more equivalent keys than a basic HFE. This induces a structural weakness which can be exploited to adapt the KS attack against classical modifiers of multivariate schemes such as minus and embedding. Along the way, we discovered that the KS attack as initially described cannot be applied against HFE in characteristic 2. We have then strongly revised KS in characteristic 2 to make it work. In all cases, the cost of our attacks is related to the complexity of solving MinRank. Thanks to recent complexity results on this problem, we prove that our attack is polynomial in the degree of the extension field for all possible practical settings used in HFE and Multi-HFE. This makes then Multi-HFE less secure than basic HFE for equally-sized keys. As a proof of concept, we have been able to practically break the most conservative proposed parameters of multi-HFE in few days (256 bits security broken in 9 days).

6.3.6. Cryptanalysis of a Public-Key Encryption Scheme Based on New Multivariate Quadratic Assumptions

In [24], we investigate the security of a public-key encryption scheme introduced by Huang, Liu and Yang (HLY) at PKC'12. This new scheme can be provably reduced to the hardness of solving a set of quadratic equations whose coefficients of highest degree are chosen according to a discrete Gaussian distributions. The other terms being chosen uniformly at random. Such a problem is a variant of the classical problem of solving a system of non-linear equations (PoSSo), which is known to be hard for random systems. The main hypothesis of Huang, Liu and Yang is that their variant is not easier than solving PoSSo for random instances. In this paper, we disprove this hypothesis. To this end, we exploit the fact that the new problem proposed by Huang, Liu and Yang reduces to an easy instance of the Learning With Errors (LWE) problem. The main contribution of this paper is to show that security and efficiency are essentially incompatible for the HLY proposal. That is, one cannot find parameters which yield a secure and a practical scheme. For instance, we estimate that a public-key of at least 1.03 GB is required to achieve 80-bit security against known attacks. As a proof of concept, we present practical attacks against all the parameters proposed Huang, Liu and Yang. We have been able to recover the private-key in roughly one day for the first challenge (i.e. Case 1) proposed by HLY and in roughly three days for the second challenge (i.e. Case 2).

6.3.7. On the Complexity of the BKW Algorithm on LWE

In [3], we present 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.

6.3.8. Combined Attack on CRT-RSA. Why Public Verification Must Not Be Public?

In [25] we introduce a new Combined Attack on a CRT-RSA implementation resistant against Side-Channel Analysis and Fault Injection attacks. Such implementations prevent the attacker from obtaining the signature when a fault has been induced during the computation. Indeed, such a value would allow the attacker to recover the RSA private key by computing the gcd of the public modulus and the faulty signature. The principle of our attack is to inject a fault during the signature computation and to perform a Side-Channel Analysis targeting a sensitive value processed during the Fault Injection countermeasure execution. The resulting information is then used to factorize the public modulus, leading to the disclosure of the whole RSA private key. After presenting a detailed account of our attack, we explain how its complexity can be significantly reduced by using Coppersmith's techniques based on lattice reduction. We also provide simulations that confirm the efficiency of our attack as well as two different countermeasures having a very small impact on the performance of the algorithm. As it performs a Side-Channel Analysis during a Fault Injection countermeasure to retrieve the secret value, this article recalls the need for Fault Injection and Side-Channel Analysis countermeasures as monolithic implementations.

6.3.9. Polynomial root finding over local rings and application to error correcting codes

GURUSWAMI and SUDAN designed a polynomial-time list-decoding algorithm. Their method divides into two steps. First it computes a polynomial Q in $\mathbb{F}_q[x][y]$ such that the possible transmitted messages are roots of Q in $\mathbb{F}_q[x]$. In the second step one needs to determine all such roots of Q. Several techniques have been investigated to solve both steps of the problem.

The Guruswami and Sudan algorithm has been adapted to other families of codes such as algebraic-geometric codes and alternant codes over fields. Extensions over certain types of finite rings have further been studied for Reed-Solomon codes, for alternant codes, and for algebraic-geometric codes. In all these cases, the two main steps of the Guruswami and Sudan algorithm are roughly preserved, but to the best of our knowledge, the second step has never been studied into deep details from the complexity point of view. In [5], we investigate root-finding for polynomials over *Galois rings*, which are often used within these error correcting codes, and that are defined as non-ramified extension of $\mathbb{Z}/p^n\mathbb{Z}$. We study the cost of our algorithms, discuss their practical performances, and apply our results to the Guruswami and Sudan list decoding algorithm over Galois rings.

SECRET Project-Team

5. New Results

5.1. Symmetric cryptosystems

Participants: Anne Canteaut, Pascale Charpin, Virginie Lallemand, Gaëtan Leurent, María Naya-Plasencia, Joëlle Roué, Valentin Suder.

From outside, it might appear that symmetric techniques become obsolete after the invention of public-key cryptography in the mid 1970's. However, they are still widely used because they are the only ones that can achieve some major features like high-speed or low-cost encryption, fast authentication, and efficient hashing. Today, we find symmetric algorithms in GSM mobile phones, in credit cards, in WLAN connections. Symmetric cryptology is a very active research area which is stimulated by a pressing industrial demand for low-cost implementations (in terms of power consumption, gate complexity...). These extremely restricted implementation requirements are crucial when designing secure symmetric primitives and they might be at the origin of some weaknesses. Actually, these constraints seem quite incompatible with the rather complex mathematical tools needed for constructing a provably secure system.

The specificity of our research work is that it considers all aspects of the field, from the practical ones (new attacks, concrete specifications of new systems) to the most theoretical ones (study of the algebraic structure of underlying mathematical objects, definition of optimal objects). But, our purpose is to study these aspects not separately but as several sides of the same domain. Our approach mainly relies on the idea that, in order to guarantee a provable resistance to the known attacks and to achieve extremely good performance, a symmetric cipher must use very particular building blocks, whose algebraic structures may introduce unintended weaknesses. Our research work captures this conflict for all families of symmetric ciphers. It includes new attacks and the search for new building blocks which ensure both a high resistance to the known attacks and a low implementation cost. This work, which combines cryptanalysis and the theoretical study of discrete mathematical objects, is essential to progress in the formal analysis of the security of symmetric systems.

In this context, the very important challenges are the designs of low-cost ciphers and of secure hash functions. Most teams in the research community are actually working on the design and on the analysis (cryptanalysis and optimisation of the performance) of such primitives.

5.1.1. Hash functions

Following the recent attacks against almost all existing hash functions (MD5, SHA-0, SHA-1...), we have initiated a research work in this area, especially within the Saphir-2 ANR project and with several PhD theses. Our work on hash functions is two-fold: we have designed two new hash functions, named FSB and Shabal, which have been submitted to the SHA-3 competition, and we have investigated the security of several hash functions, including the new SHA-3 standard.

Recent results:

- Upper bounds on the degree of an iterated permutation from the degree of the inverse of the inner transformation; this result has been applied both to hash functions and to block ciphers. Most notably, this work leads to the best (theoretical) analysis of the hash function Keccak, which has been selected for the new SHA-3 standard [11].
- Study of a new technique for attacking symmetric primitives based on the existence of linear relations between some input and output bits of the Sbox. This method has been used for improving the best known attack against the SHA-3 candidate Hamsi [36], [58].

5.1.2. Block ciphers

Even if the security of the current block cipher standard, AES, is not threatened when it is used in a classical context, there is still a need for the design of improved attacks, and for the determination of design criteria which guarantee that the existing attacks do not apply. This notably requires a deep understanding of all previously proposed attacks. Moreover, there is a high demand from the industry of lightweight block ciphers for some constrained environments. Several such algorithms have been proposed in the last few years and their security should be carefully analysed. Most of our work in this area is related to an ANR Project named BLOC.

Recent results:

- Cryptanalysis of several recently proposed lightweight block ciphers. This includes an attack against the full cipher KLEIN-64 [66], [49], and an attack against 8 rounds (out of 12) of PRINCE [37].
- Analysis of the resistance of AES-like permutations to improved rebound attacks. Most notably, this improved technique leads to a distinguisher on 10 rounds of the internal permutation of the SHA-3 candidate Grøstl [14].
- Proposal of a new family of distinguishers against AES-based permutations, named *limited-birthday distinguishers*; these distinguishers exploit some some improved rebound techniques. They have been successfully applied to various AES-based primitives including AES, ECHO, Grøstl, LED, PHOTON and Whirlpool [42].
- Design of an improved variant of Meet-in-the-Middle attacks, named *Sieve-in-the-Middle*: instead of selecting the key candidates by searching for a collision in an intermediate state which can be computed forwards and backwards, we here look for the existence of valid transitions through some middle Sbox. In the same paper, an improved technique is also proposed to build bicliques without needing any additional data (on the contrary to classical biclique attacks). These new methods have been exploited to break 8 rounds (out of 12) of the lightweight block cipher PRINCE [37], [59], [30].
- Analysis of the differential properties of the AES Superbox [48].
- Design of a new block cipher, named ZORRO, for which physical security is considered as an optimisation criterion [41].
- Design and study of a new construction for low-latency block ciphers, named *reflection ciphers*, which generalises the so-called α-reflection property exploited in PRINCE. This construction aims at reducing the implementation overhead of decryption on top of encryption [24].

5.1.3. 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, characterising 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. For instance, bent functions, which are the Boolean functions which achieve the highest possible nonlinearity, have been extensively studied in order to provide some elements for a classification, or to adapt these functions to practical cryptographic constructions. We have also been interested in functions with a low differential uniformity (*e.g.*, APN functions), which are the S-boxes ensuring an (almost) optimal resistance to differential cryptanalysis.

Recent results:

- Study of the algebraic properties (e.g. the algebraic degree) of the inverses of APN power permutations [16], [51].
- Definition of a new criterion for Sboxes and link with some recent algebraic attacks on the hash function Hamsi [36], [58].
- Definition of some extended criterion for estimating the resistance of a block cipher to differential attacks. Most notably, this new criterion points out the fact that affinely equivalent Sboxes may not provide the same security level regarding differential cryptanalysis. 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 [21], [48].
- A new sufficient (and simpler) condition for checking that a mapping is APN has been established [62].
- Surveys of PN and APN mappings [55], [54].

5.2. Code-based cryptography

Participants: Grégory Landais, Rafael Misoczki, Nicolas Sendrier, Dimitrios Simos, Jean-Pierre Tillich.

Most popular public-key cryptographic schemes rely either on the factorisation problem (RSA, Rabin), or on the discrete logarithm problem (Diffie-Hellman, El Gamal, DSA). These systems have evolved and today instead of the classical groups ($\mathbf{Z}/n\mathbf{Z}$) we may use groups on elliptic curves. They allow a shorter block and key size for the same level of security. An intensive effort of the research community has been and is still being conducted to investigate the main aspects of these systems: implementation, theoretical and practical security. It must be noted that these systems all rely on algorithmic number theory. As they are used in most, if not all, applications of public-key cryptography today (and it will probably remain so in the near future), cryptographic applications are thus vulnerable to a single breakthrough in algorithmics or in hardware (a quantum computer can break all those schemes).

Diversity is a way to dilute that risk, and it is the duty of the cryptographic research community to prepare and propose alternatives to the number theoretic based systems. The most serious tracks today are lattice-based cryptography (NTRU,...), multivariate cryptography (HFE,...) and code-based cryptography (McEliece encryption scheme,...). All these alternatives are referred to as *post-quantum cryptosystems*, since they rely on difficult algorithmic problems which would not be solved by the coming-up of the quantum computer.

The code-based primitives have been investigated in details within the project-team. 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:

- Design of a new variant of McEliece using Moderate Density Parity Check (MDPC) codes [45];
- Cryptanalysis of McEliece system based on Wild Goppa codes from a quadratic finite field extension. This polynomial-time structural attack relies on some filtration of nested subcodes which will reveal the secret algebraic description of the underlying secret code [39], [63].
- Cryptanalysis of a variant of the McEliece cryptosystem based on Reed-Solomon codes [38].

- Cryptanalysis of a variant of the McEliece cryptosystem based on convolutional codes proposed by Löndahl and Johansson in 2012 [43].
- Design of the first algorithm for distinguishing between Goppa codes (or alternant codes) over any field and random codes. Provided that the codes have sufficiently large rates, this technique can solve in polynomial-time the Goppa-Code-Distinguishing problem, which is an assumption in the security proof of McEliece cryptosystem [12].
- Study of the hardness of the code equivalence problem over $\mathbf{F}_q$. This problem has been extensively studied for permutation-equivalence (which covers all cases for $q = 2$). For $q \in \{3, 4\}$, we have generalised the support-splitting algorithm, and we have shown that the problem seems intractable for most instances when $q \geq 5$ [46]. This property has been exploited in an improvement version of an identification protocol due to Girault [47].

5.3. Reverse engineering of communication systems

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

To assess the quality of a cryptographic algorithm, it is usually assumed that its specifications are public, as, in accordance with Kerckhoffs principle [1], it would be dangerous to rely, even partially, on the fact that the adversary does not know those specifications. However, this fundamental rule does not mean that the specifications are known to the attacker. In practice, before mounting a cryptanalysis, it is necessary to strip off the data. This reverse engineering process is often subtle, even when the data formatting is not concealed on purpose. A typical case is interception; some raw data, not necessarily encrypted, are observed out of a noisy channel. To access the information, the whole communication system has first to be disassembled and every constituent reconstructed. 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 DGA.

Recent results:

- Reconstruction of the constellation labelling (i.e. used in the modulator of a communication system) in the presence of errors and when the underlying code is convolutional (Marion Bellard's PhD).

5.4. Quantum information theory

Participants: André Chailloux, Anthony Leverrier, Denise Maurice, Jean-Pierre Tillich.

The field of Quantum Information and Computation aims at exploiting the laws of quantum physics to manipulate information in radically novel ways. Two main applications come to mind: quantum computers, that offer the promise of solving some problems intractable with classical computers (for instance, factorization); and quantum cryptography, which provides new ways to exchange data in a provably secure fashion.

The main obstacle towards the development of quantum computing is decoherence, a consequence of the interaction of the computer with a noisy environment. We investigate approaches to quantum error-correction as a way to fight against this effect, and we study more particularly some families of quantum error-correcting codes which generalise the best classical codes available today.

Our research also covers quantum cryptography where we study the security of efficient protocols for key distribution, in collaboration with experimental groups. More generally, we investigate how quantum theory severely constrains the action of honest and malicious parties in cryptographic scenarios.

5.4.1. 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. Our approach for overcoming this problem has been to study whether or not the family of turbo-codes and LDPC codes (and the associated iterative decoding algorithms) have a quantum counterpart.

[1] Kerckhoffs stated that principle in a paper entitled *La Cryptographie militaire*, published in 1883.

Recent results:

- Construction of quantum codes combining an improved version of a family of spatially coupled quantum LDPC codes with a family of error reducing turbo-codes [44];
- construction of quantum LDPC codes with fixed non-zero rate and a minimum distance which grows proportionally to the square root of the block-length. This greatly improves the previously best known construction whose minimum distance was logarithmic in the block-length [19].
- Mamdouh Abbara's PhD thesis [9]

5.4.2. 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 with security properties based on quantum theory.

Recent results:

- Experimental demonstration of quantum key distribution with continuous variables over 80 km [15], greatly improving over previous records around 25 km.
- Security proof of continuous-variable quantum key distribution protocols against general attacks [17], [29].
- Security proof of device-independent quantum key distribution in the bounded storage model [18].
- Study of BosonSampling, a recently introduced problem where quantum computers offer a provable speedup over classical computers [67], [28].
- Introduction and study of "Local Orthogonality", an information-theoretical principle for quantum correlations [13], [68].
- Introduction of a general formalism for the study of contextuality and non locality in quantum theory, based on the combinatorics of hypergraphs [65], [27].

Specfun Team

6. New Results

6.1. Creative telescoping for bivariate hyperexponential functions

In [8], we gave a new algorithm for the symbolic integration of bivariate hyperexponential functions, which outperforms state-of-the-art implementations like Maple's function *DEtools[Zeilberger]*. The approach was to extend Hermite's reduction for rational functions and the Hermite-like reduction for hyperexponential functions in a suitable way. A key feature of the algorithm is that it can avoid the costly computation of certificates.

6.2. Creative telescoping for rational functions

In [10] we described a precise and elementary algorithmic version of the Griffiths–Dwork method for the creative telescoping of rational functions. This leads to bounds on the order and degree of the coefficients of the differential equation, and to the first complexity result which is single exponential in the number of variables. One of the important features of the algorithm is that it does not need to compute certificates. The approach is vindicated by a prototype implementation.

6.3. Complexity of the uncoupling of linear functional systems

Uncoupling algorithms transform a linear differential system of first order into one or several scalar differential equations. We examined in [9] two approaches to uncoupling: the cyclic-vector method (*CVM*) and the Danilevski-Barkatou-Zürcher algorithm (*DBZ*). We gave tight size bounds on the scalar equations produced by *CVM*, and designed a fast variant of *CVM* whose complexity is quasi-optimal with respect to the output size. We exhibited a strong structural link between *CVM* and *DBZ* enabling to show that, in the generic case, *DBZ* has polynomial complexity and that it produces a single equation, strongly related to the output of *CVM*. We proved that algorithm *CVM* is faster than *DBZ* by almost two orders of magnitude, and provided experimental results that validate the theoretical complexity analyses.

6.4. Computation of integrals related to the Ising model

We showed in [2] that the n-fold integrals of the magnetic susceptibility of the Ising model, as well as various other n-fold integrals of the "Ising class", or n-fold integrals from enumerative combinatorics, like lattice Green functions, correspond to a distinguished class of functions generalising algebraic functions: they are actually diagonals of rational functions. This algebraic structure explains many remarkable properties of the integrals of the Ising class.

6.5. Non-D-finite excursions in the quarter plane

The number of excursions (finite paths starting and ending at the origin) having a given number of steps and obeying various geometric constraints is a classical topic of combinatorics and probability theory. We proved in [3] that the sequence of numbers of excursions in the quarter plane corresponding to a nonsingular step set $S \subseteq \{0, \pm 1\}^2$ with infinite group does not satisfy any nontrivial linear recurrence with polynomial coefficients. Accordingly, in those cases, the trivariate generating function of the numbers of walks with given length and prescribed ending point is not D-finite. This solves an open problem in the field of lattice path combinatorics.

6.6. A human proof of Gessel's lattice path conjecture

Gessel walks 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 Gessel 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 proposed in [15] the first "human proofs" of these results. They are derived from a new expression for the generating function of Gessel walks.

6.7. Efficient algorithms for rational first integrals

We presented in [14] fast algorithms for computing rational first integrals with bounded degree of a planar polynomial vector field. Our approach is inspired by an idea of Ferragut and Giacomini. We improve upon their work by proving that rational first integrals can be computed 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 solving the problem in $\tilde{\mathrm{O}}(d^2N^{2\omega+1})$ arithmetic operations, where N denotes the given bound for the degree of the rational first integral, and where $d \leq N$ is the degree of the vector field, and ω the exponent of linear algebra. By comparison, the best previous algorithm uses at least $d^{\omega+1}N^{4\omega+4}$ arithmetic operations. The new algorithms are very efficient in practice.

6.8. Reactive document checking in Coq

In an effort to improve the reactivity of Coq, the way it processes and checks a single document has been completely redesigned [7]. The current development version is able to reschedule the tasks to be performed in order to minimize the time required to give interactive feedback to the user. On typical documents taken from the formal proof of the Odd Order Theorem, the worst reaction time of the tool dropped from 5 minutes to 9 seconds. This improvement will be part of the next stable release of the Coq system.

6.9. Efficient normalization of ring/field expressions in Coq

The implementation of Coq's proof commands for manipulation of ring/field expressions has been improved in response to the demand for better efficiency that emerged in the formalization of Apéry's irrationality proof of $\zeta(3)$. The data structure used for the abstract syntax tree of ring/field expressions has been refined to enable a more efficient and more precise interpretation into concrete ring/field expressions. Moreover the collection of non-nullity conditions for denominators in a field expressions has been speeded up, making the type-checking time of a field normalization proof not be dominated by this collecting phase.

6.10. Documentation of Coq's canonical structures

The device employed to model a hierarchy of algebraic structures with overloaded notations in Coq has been documented in [6] and in the user manual of the tool.

6.11. Maintenance and development of the SSReflect extension for Coq and its user manual

The Small Scale Reflection extension of Coq has been maintained together with its user manual. Some new linguistic constructs to model non-structural reasoning and to enable the user to better factor out repeated arguments have been developed and documented. Some language constructs have been made compatible with the type-classes mechanism offered by Coq. The release of version 1.5 has been prepared.

6.12. Efficient proof-search techniques in sequent calculus

We have proposed in [11] a sequent calculus which is focussed, polarized, and parameterized by an abstract notion of theory. This new combination of features aims at proposing a framework which is adapted to the simulation in sequent calculus of efficient, general-purpose decision procedures (tableaux methods, satisfiability, ...) that can interact with theory-specific decision procedures (for linear arithmetics, arrays, ...). In particular we propose a tight simulation of the Davis–Putnam–Logemann–Loveland algorithm modulo theory, and show how to simulate some advanced optimizations that are crucial to realistic implementations of SMT solvers.

6.13. A formal proof of the irrationality of $\zeta(3)$

We have obtained a formal proof, machine-checked by the Coq proof assistant, of the irrationality of the constant $\zeta(3)$, under the single assumption of the asymptotic behavior of the least common multiple of the first n natural numbers. The core of this formal proof is based on (untrusted) computer-algebra calculations performed outside the proof assistant with the Algolib Maple library. Then, we verify formally and a posteriori the desired properties of the objects computed by Maple and complete the proof of irrationality.

6.14. Documentation of the Mathematical Components libraries

The approach to finite-group theory adopted in the libraries formalizing in Coq the proof of the Odd Order Theorem has been documented in [5].

VEGAS Project-Team

5. New Results

5.1. Classical and probabilistic computational geometry

Participants: Xavier Goaoc, Guillaume Moroz, Sylvain Lazard, Marc Pouget.

5.1.1. Probabilistic complexity analysis of random geometric structures

Average-case analysis of data-structures or algorithms is commonly used in computational geometry when the more classical worst-case analysis is deemed overly pessimistic. Since these analyses are often intricate, the models of random geometric data that can be handled are often simplistic and far from "realistic inputs".

Complexity analysis of random geometric structures made simpler. In a joint work with Olivier Devillers and Marc Glisse (Inria Geometrica), we presented a new simple scheme for the analysis of geometric structures. While this scheme only produces results up to a polylog factor, it is much simpler to apply than the classical techniques and therefore succeeds in analyzing new input distributions related to smoothed complexity analysis. We illustrated our method on two classical structures: convex hulls and Delaunay triangulations. Specifically, we gave short and elementary proofs of the classical results that n points uniformly distributed in a ball in R^d have a convex hull and a Delaunay triangulation of respective expected complexities $\widetilde{\Theta}(n^{((d+1)/(d-1))})$ and $\widetilde{\Theta}(n)$. We then prove that if we start with n points well-spread on a sphere, e.g. an (ϵ, κ)-sample of that sphere, and perturb that sample by moving each point randomly and uniformly within distance at most δ of its initial position, then the expected complexity of the convex hull of the resulting point set is $\widetilde{\Theta}(\sqrt{(}n)^{(1-1/d)}\delta^{-(d-1)/(4d)})$. We presented these results in the *Symposium on Computational Geometry* 2013 [20].

Monotonicity of the number of facets of random polytopes. We also proved a result on the size of the convex hull K_n of n points sampled uniformly in a convex set K. More precisely, let $u_n^{K,i}$ be the expected number of facets of dimension i of the convex hull. We proved that, in the plane, $u_n^{K,0}$ is an increasing sequence. In higher dimension, if K is a convex, smooth, compact body, then we showed that the sequence $u_n^{K,d-1}$ is asymptotically increasing. This result, published in the *Electronic Communications in Probability* [13], was obtained in collaboration with Olivier Devillers and Marc Glisse (Inria Geometrica) and Matthias Reitzner (Osnabruck Univ.).

Worst-case silhouette size of random polytopes. Finally, we studied from a probabilistic point of view the size of the silhouette of a polyhedron. While the silhouette size of a polyhedron with n vertices may be linear for some view points, several experimental and theoretical studies show a sublinear behavior for a wide range of constraints. The latest result on the subject proves a bound in $\Theta(\sqrt{n})$ on the size of the silhouette from a random view point of polyhedra of size n approximating non-convex surfaces in a reasonable way [9]. This result considers the polyhedron given and average the sizes of the silhouettes over all view points. This year, we addressed the problem of bounding the worst-case size of the silhouette where the average is taken over a set of polyhedra. Namely, we consider random polytopes defined as the convex hull of a Poisson point process on a sphere in $\mathbb{R}^3$ such that its average number of points is n. We show that the expectation over all such random polytopes of the maximum size of their silhouettes viewed from infinity is $\Theta(\sqrt{n})$. This work was done in collaboration with Marc Glisse (Inria Geometrica) and Julien Michel (Université de Poitiers) [24].

5.1.2. Embedding geometric structures

We continued working this year on the problem of embedding geometric objects on a grid of $\mathbb{R}^3$. Essentially all industrial applications take, as input, models defined with a fixed-precision floating-point arithmetic, typically doubles. As a consequence, geometric objects constructed using exact arithmetic must be embedded on a fixed-precision grid before they can be used as input in other software. More precisely, the problem is, given a geometric object, to find a similar object representable with fixed-precision floating-point arithmetic, where similar means topologically equivalent, close according to some distance function, etc. We are working on the problem of rounding polyhedral subdivisions on a grid of $\mathbb{R}^3$, where the only known method, due to Fortune in 1999, considers a grid whose refinement depends on the combinatorial complexity of the input, which does not solve the problem at hand. This project is joint work with Olivier Devillers (Inria Geometrica) and William Lenhart (Williams College, USA).

5.1.3. Bounded-Curvature Shortest Paths

We considered the problem of computing shortest paths having curvature at most one almost everywhere and visiting a sequence of n points in the plane in a given order. This problem is a sub-problem of the Dubins Traveling Salesman Problem and also arises naturally in path planning for point car-like robots in the presence of polygonal obstacles. We showed that when consecutive waypoints are distance at least four apart, this question reduces to a family of convex optimization problems over polyhedra in $\mathbb{R}^n$. This result, done in collaboration with Hyo-Sil Kim (KAIST) was published in the *SIAM Journal on Computing* [15].

5.1.4. Approximating Geodesics in Meshes

A standard way to approximate the distance between any two vertices p and q on a mesh is to compute, in the associated graph, a shortest path from p to q that goes through one of k sources, which are well-chosen vertices. Precomputing the distance between each of the k sources to all vertices of the graph yields an efficient computation of approximate distances between any two vertices. One standard method for choosing k sources, which has been used extensively and successfully for isometry-invariant surface processing, is the so-called *Farthest Point Sampling* (FPS), which starts with a random vertex as the first source, and iteratively selects the farthest vertex from the already selected sources.

We analyzed the stretch factor $\mathcal{F}_{FPS}$ of approximate geodesics computed using FPS, which is the maximum, over all pairs of distinct vertices, of their approximated distance over their geodesic distance in the graph. We show that $\mathcal{F}_{FPS}$ can be bounded in terms of the minimal value $\mathcal{F}^*$ of the stretch factor obtained using an optimal placement of k sources as $\mathcal{F}_{FPS} \leq 2r_e^2\mathcal{F}^* + 2r_e^2 + 8r_e + 1$, where r_e is the ratio of the lengths of the longest and the shortest edges of the graph. This provides some evidence explaining why farthest point sampling has been used successfully for isometry-invariant shape processing. Furthermore, we showed that it is NP-complete to find k sources that minimize the stretch factor [25].

5.1.5. On Point-sets that Support Planar Graphs

A set of points is said universal if it supports a crossing-free drawing of any planar graph. For a planar graph with n vertices, if bends on edges of the drawing are permitted, universal point-sets of size n are known, but only if the bend-points are in arbitrary positions. If the locations of the bend-points must also be specified as part of the point-set, no result was known, and we prove that any planar graph with n vertices can be drawn on a universal set $\mathcal{S}$ of $O(n^2/\log n)$ points with at most one bend per edge and with the vertices and the bend points in $\mathcal{S}$. If two bends per edge are allowed, we show that $O(n \log n)$ points are sufficient, and if three bends per edge are allowed, $\Theta(n)$ points are sufficient. When no bends on edges are permitted, no universal point-set of size $o(n^2)$ is known for the class of planar graphs. We show that a set of n points in balanced biconvex position supports the class of maximum-degree-3 series-parallel lattices. These results were published this year in the journal *Computational Geometry: Theory and Application* [14].

We also considered the setting in which graphs are drawn with curved edges. We proved that, surprisingly, there exists a universal set of n points in the plane for which every n-vertex planar graph admits a planar drawing in which the edges are drawn as a circular arc. This result was presented in the *Canadian Conference on Computational Geometry* [17].

5.2. Non-linear computational geometry

Participants: Guillaume Moroz, Sylvain Lazard, Marc Pouget, Yacine Bouzidi, Laurent Dupont.

5.2.1. Solving bivariate systems and topology of algebraic curves

In the context of our algorithm Isotop for computing the topology of algebraic curves [4], we work on the problem of solving a system of two bivariate polynomials. We focus on the problem of computing a Rational Univariate Representation (RUR) of the solutions, that is, roughly speaking, a univariate polynomial and two rational functions which map the roots of the polynomial to the two coordinates of the solutions of the system.

Separating linear forms. We first presented an algorithm for computing a separating linear form of a system of bivariate polynomials with integer coefficients, that is a linear combination of the variables that takes different values when evaluated at distinct (complex) solutions of the system. In other words, a separating linear form defines a shear of the coordinate system that sends the algebraic system in generic position, in the sense that no two distinct solutions are vertically aligned. The computation of such linear forms is at the core of most algorithms that solve algebraic systems by computing rational parameterizations of the solutions and, moreover, the computation of a separating linear form is the bottleneck of these algorithms, in terms of worst-case bit complexity. Given two bivariate polynomials of total degree at most d with integer coefficients of bitsize at most τ, our algorithm computes a separating linear form in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations in the worst case, which decreases by a factor d^2 the best known complexity for this problem ($\widetilde{O}_B$ refers to the complexity where polylogarithmic factors are omitted and O_B refers to the bit complexity). This result was presented at the *International Symposium on Symbolic and Algebraic Computation* in 2013 [19] and submitted to a journal [23].

Solving bivariate systems & RURs. Given such a separating linear form, we also presented an algorithm for computing a RUR with worst-case bit complexity in $\widetilde{O}_B(d^7 + d^6\tau)$ and a bound on the bitsize of its coefficients in $\widetilde{O}(d^2 + d\tau)$. We showed in addition that isolating boxes of the solutions of the system can be computed from the RUR with $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations. Finally, we showed how a RUR can be used to evaluate the sign of a bivariate polynomial (of degree at most d and bitsize at most τ) at one real solution of the system in $\widetilde{O}_B(d^8 + d^7\tau)$ bit operations and at all the $\Theta(d^2)$ real solutions in only $O(d)$ times that for one solution. These results were also presented at the *International Symposium on Symbolic and Algebraic Computation* in 2013 [18] and submitted to a journal [22].

This work is done in collaboration with Fabrice Rouillier (project-team Ouragan at Inria Paris-Rocquencourt).

5.2.2. Reflection through quadric mirror surfaces

We addressed the problem of finding the reflection point on a quadric mirror surfaces 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. This is a classical problem known as Alhazen's problem dating from around 1000 A.D. and based on the work of Ptolomy around 150 A.D. [31], [33]. We proposed a new algorithm for this problem based on our algorithm for the computation of the intersection of quadrics [7], [30] and using a characterization the reflection point as the tangential intersection point between the mirror and an ellipsoid with foci P_1 and P_2. The implementation is in progress. This work is done in collaboration with Nuno Gonçalves, University of Coimbra (Portugal).

5.2.3. Fast polynomial evaluation and composition

Evaluating a polynomial can be done with different evaluation schemes. The Hörner scheme for example allows to evaluate a polynomial of degree n in $O(n)$ arithmetic operations. When the cost of the arithmetic operations is constant, such as in floating point arithmetic, this leads to $O(n)$ binary operations. However, with integers, the size of the elements grows linearly after each multiplication and this may lead to $O(n^2)$ binary operations. This problem arises also with polynomial composition.

The best way to handle these cases is to use divide-and-conquer algorithms to keep a linear complexity in the degree up to logarithmic factors. State-of-the-art algorithms split at the highest pure power of 2 lower or equal to $\frac{n}{2}$. However when n is not a pure power of 2, this strategy might not be optimal.

We developed the library *fast_polynomial* to explore different divide-and-conquer schemes and observed notably that splitting at $\lfloor \frac{n}{2} \rfloor$ is more efficient in some cases. In particular, this evaluation scheme does not suffer the staircase effect observed in state-of-the-art evaluations. Experimentally, it is always faster than our own implementation of the classical divide-and-conquer scheme, and faster than the state of the art library *Flint 2* when the degree of the input polynomial is between 2^k and $2^k + 2^{k-1}$. These results are presented in the technical report [26].

5.3. Combinatorics and combinatorial geometry

Participant: Xavier Goaoc.

5.3.1. Simplifying inclusion-exclusion formulas

In a joint work with Jiří Matoušek, Pavel Paták, Zuzana Safernová, Martin Tancer (Charles University, Prague, Czech republic), we worked on computing simplified inclusion-exclusion formulas. Let $\mathcal{F} = \{F_1, F_2, ..., F_n\}$ be a family of n sets on a ground set S, such as a family of balls in $\mathbb{R}^d$. For every finite measure μ on S, such that the sets of $\mathcal{F}$ are measurable, the classical *inclusion-exclusion formula* asserts that $\mu(F_1 \cup F_2 \cup \cdots \cup F_n) = \sum_{I:\varnothing \neq I \subseteq [n]} (-1)^{|I|+1} \mu\left(\bigcap_{i \in I} F_i\right)$; that is, the measure of the union is expressed using measures of various intersections. The number of terms in this formula is exponential in n, and a significant amount of research, originating in applied areas, has been devoted to constructing simpler formulas for particular families $\mathcal{F}$. We provide an upper bound valid for an arbitrary $\mathcal{F}$: we show that every system $\mathcal{F}$ of n sets with m nonempty fields in the Venn diagram admits an inclusion-exclusion formula with $m^{O(\log^2 n)}$ terms and with ± 1 coefficients, and that such a formula can be computed in $m^{O(\log^2 n)}$ expected time. We also construct systems of n sets on n points for which every valid inclusion-exclusion formula has the sum of absolute values of the coefficients at least $\Omega(n^{3/2})$. This work was presented at the EUROCOMB conference [21] in September 2013.

5.3.2. Helly numbers of acyclic families

In a joint work with Éric Colin de Verdière (CNRS-ENS) and Grégory Ginot (IMJ-UPMC), we worked on applications of algebraic topology to combinatorial geometry, and more precisely on extending classical results on nerve complexes. The nerve complex of a family is an abstract simplicial complex that encode its intersection patterns. Nerves are widely used in computational geometry and topology, in particular in reconstruction problems where one aims at inferring the geometry of an object from a point sample while guaranteeing that the topology is correct. Indeed, the *nerve theorem* ensures that the nerve of a family of geometric objects has the same "topology" (formally: homotopy type) as the union of the objects whenever they form a "good cover", that is, when any subset of the objects has an empty or contractible intersection. We relaxed this "good cover" condition to allow for families of non-connected sets. We defined an analogue of the nerve, called the *multinerve*, that is suitable for general acyclic families, and we proved that this combinatorial structure enjoys an analogue of the nerve theorem. Using multinerve, we could derive a new *topological Helly-type theorem* for acyclic families that generalizes previous results of Amenta, Kalai and Meshulam, and Matoušek. We finally used this new Helly-type theorem to (re)prove, in a unified way, bounds on transversal Helly numbers in *geometric transversal theory*. This article was submitted to the journal *Advances in mathematics* in 2012; it was accepted in 2013 and will appear in 2014 [16].

5.3.3. Set systems and families of permutations with small traces

In a joint work with Otfried Cheong (KAIST, South Korea) and Cyril Nicaud (Univ. Marne-La-Vallée), we studied two problems of the following flavor: how large can a family of combinatorial objects defined on a finite set be if its number of distinct "projections" on any small subset is bounded? We first consider set systems, where the "projections" is the standard notion of trace, and for which we generalized Sauer's Lemma on the size of set systems with bounded VC-dimension. We then studied families of permutations, where the "projections" corresponds to the notion of containment used in the study of permutations with excluded patterns, and for which we delineated the main growth rates ensured by projection conditions. One of our motivations for considering these questions is the "geometric permutation problem" in geometric transversal

theory, a question that has been open for two decades. This work was submitted to the European Journal of Combinatorics in 2012 and published in 2013 [12].

ALF Project-Team

6. New Results

6.1. Processor Architecture within the ERC DAL project

Participants: Pierre Michaud, Nathanaël Prémillieu, Luis Germán Garcia Morales, Bharath Narasimha Swamy, Sylvain Collange, André Seznec, Arthur Perais, Surya Natarajan, Sajith Kalathingal, Tao Sun, Andrea Mondelli, Aswinkumar Sridharan, Alain Ketterlin, Kamil Kedzierski.

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 (may be 1000's) simpler, more silicon and power effective cores.

In the DAL research project, http://www.irisa.fr/alf/index.php?option=com_content&view=article&id=55&Itemid=3&lang=en, 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.

6.1.1. Microarchitecture exploration of control flow reconvergence

Participants: Nathanaël Prémillieu, André Seznec.

After continuous progress over the past 15 years [8], [10], the accuracy of branch predictors seems to be reaching a plateau. Other techniques to limit control dependency impact are needed. Control flow reconvergence is an interesting property of programs. After a multi-option control-flow instruction (i.e. either a conditional branch or an indirect jump including returns), all the possible paths merge at a given program point: the reconvergence point.

Superscalar processors rely on aggressive branch prediction, out-of-order execution and instruction level parallelism for achieving high performance. Therefore, on a superscalar core , the overall speculative execution after the mispredicted branch is cancelled, leading to a substantial waste of potential performance. However, deep pipelines and out-of-order execution induce that, when a branch misprediction is resolved, instructions following the reconvergence point have already been fetched, decoded and sometimes executed. While some of this executed work has to be cancelled since data dependencies exist, canceling the control independent work is a waste of resources and performance. We have proposed a new hardware mechanism called SYRANT, SYmmetric Resource Allocation on Not-taken and Taken paths, addressing control flow reconvergence at a reasonable cost. Moreover, as a side contribution of this research we have shown that, for a modest hardware cost, the outcomes of the branches executed on the wrong paths can be used to guide branch prediction on the correct path [13].

6.1.2. Efficient Execution on Guarded Instruction Sets

Participants: Nathanaël Prémillieu, 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 use and high confidence only guard prediction use. 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 [35], [34], [13].

6.1.3. 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.

First, we introduce a new value predictor VTAGE harnessing the global branch history [32]. VTAGE directly inherits the structure of the indirect jump predictor ITTAGE [8]. VTAGE is able to predict with a very high accuracy many values that were not correctly predicted by previously proposed predictors, such as the FCM predictor and the stride predictor. Three sources of information can be harnessed by these predictors: the global branch history, the differences of successive values and the local history of values. Moreover, VTAGE does not suffer from short critical prediction loops and can seamlessly handle back-to-back predictions, contrarily to previously proposed, hard to implement FCM predictors.

Second, we show that all predictors are amenable to very high accuracy at the cost of some loss on prediction coverage [32]. 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 is crucial as adding *selective replay* in the OoO core would tremendously increase complexity.

Third, we leverage the possibility of validating predictions at commit to introduce a new microarchitecture, EOLE [31]. 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.

6.1.4. Helper threads

Participants: Bharath Narasimha Swamy, Alain Ketterlin, André Seznec.

As the number of cores on die increases with the improvements in silicon process technology, the strategy of replicating identical cores does not scale to meet the performance needs of mixed workloads. Heterogeneous Many Cores (HMC) that mix many simple cores with a few complex cores are emerging as a design alternative that can provide both high performance and power-efficient execution. The availability of many simple cores in a HMC presents an opportunity to utilize low power cores to accelerate sequential execution on the complex core. For example simple cores can execute pre-computational (or helper) code and generate prefetch requests for the main thread.

We explore the design of a lightweight architectural framework that provides instruction set support and a low-latency interface to simple-cores for efficient helper code execution. We utilize static analyses and profile data to generate helper codelets that target delinquent loads in the main thread. The main thread is instrumented to initiate helper execution ahead of time, and utilizes instruction set support to signal helper execution on the simple core, and to pass live-in values for the helper codelet. Pre-computational code executes on the simple core and generates prefetch requests that install data into a shared last-level cache. Initial experiments with a trace based simulation framework show that helper execution has the potential to cover cache-missing loads on the main thread.

The restriction of prefetching to a lower level shared cache in a loosely coupled system limits the benefits of helper execution. The main thread should have a low latency access mechanism to data prefetched by helper execution. We plan to explore direct, yet light weight, mechanisms for data communication between the helper core and the main core.

6.1.5. Adaptive Intelligent Memory Systems

Participants: André Seznec, Aswinkumar Sridharan.

On multicores, the processors are sharing the memory hierarchy, buses, caches, and memory. The performance of any single application is impacted by its environment and the behavior of the other applications co-running on the multicore. Different strategies have been proposed to isolate the behavior of the different co-running applications, for example performance isolation cache partitioning, while several studies have addressed the global issue of optimizing throughput through the cache management.

However these studies are limited to a few cores (2-4-8) and generally features mechanisms that cannot scale to 50-100 cores. Moreover so far the academic propositions have generally taken into account a single parameter, the cache replacement policy or the cache partitioning. Other parameters such as cache prefetching and its aggressiveness already impact the behavior of a single thread application on a uniprocessor. Cache prefetching policy of each thread will also impact the behavior of all the co-running threads.

Our objective is to define an Adaptive and Intelligent Memory System management hardware, AIMS. The goal of AIMS will be to dynamically adapt the different parameters of the memory hierarchy access for each individual co-running process in order to achieve a global objective such as optimized throughput, thread fairness or respecting quality of services for some privileged threads.

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

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

Multi-core have become ubiquitous and industry is already moving towards the many-core era. Many open-ended questions remain unanswered for the upcoming many-core era. From the software perspective, it is unclear which applications will be able to benefit from many cores. From the hardware perspective, the tradeoff between implementing many simple cores, fewer medium aggressive cores or even only a moderate number of aggressive cores is still in debate. Estimating the potential performance of future parallel applications on the yet-to-be-designed future many cores is very speculative. The simple models proposed by Amdahl's law or Gustafson's law are not sufficient and may lead to erroneous conclusions. In this paper, we propose a still simple execution time model for parallel applications, the SNAS model. As previous models, the SNAS model evaluates the execution time of both the serial part and the parallel part of the application, but takes into account the scaling of both these execution times with the input problem size and the number of processors. For a given application, a few parameters are collected on the effective execution of the application with a few threads and small input sets. The SNAS model allows to extrapolate the behavior of a future application exhibiting similar scaling characteristics on a many core and/or a large input set. Our study shows that the execution time of the serial part of many parallel applications tends to increase along with the problem size, and in some cases with the number of processors. It also shows that the efficiency of the execution of the parallel part decreases dramatically with the number of processors for some applications. Our model also indicates that since several different applications scaling will be encountered, hybrid architectures featuring a few aggressive cores and many simple cores should be privileged.

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

Participants: Sylvain Collange, André Seznec, Sajith Kalathingal.

We aim at exploring the design of a unique core that efficiently run both sequential and massively parallel sections. We explore how the architecture of a complex superscalar core has to be modified or enhanced to be able to support the parallel execution of many threads from the same application (10's or even 100's a la GPGPU on a single core).

SIMD execution is the preferred way to increase energy efficiency on data-parallel workloads. However, explicit SIMD instructions demand challenging auto-vectorization or manual coding, and any change in SIMD width requires at least a recompile, and typically manual code changes. Rather than vectorize at compile-time, our approach is to dynamically vectorize SPMD programs at the micro-architectural level. The SMT-SIMD hybrid core we propose extracts data parallelism from thread parallelism by scheduling groups of threads in lockstep, in a way inspired by the execution model of GPUs. As in GPUs, conditional branches whose outcome differ between threads are handled with conditionally masked execution. However, while GPUs rely on explicit re-convergence instructions to restore lockstep execution, we target existing general-purpose instruction sets, in order to run legacy binary programs. Thus, the main challenge consists in detecting re-convergence points dynamically.

We proposed instruction fetch policies that apply heuristics to maximize the cycles spent in lockstep execution. We evaluated their efficiency and performance impact on an out-of-order superscalar core simulator. Results validate the viability of our approach, by showing that existing compiled SPMD programs are amenable to lockstep execution without modification nor recompilation.

6.2. Other Architecture Studies

Participants: Damien Hardy, Pierre Michaud, Ricardo Andrés Velásquez, Sylvain Collange, André Seznec, Sajith Kalathingal, Junjie Lai.

GPU, performance, simulation, vulnerability

6.2.1. Performance Upperbound Analysis of GPU applications

Participants: Junjie Lai, André Seznec.

In the framework of the ANR Cosinus PetaQCD project (ended Oct 2012), we have been modeling the demands of high performance scientific applications on hardware. GPUs have become popular and cost-effective hardware platforms. In this context, we have been addressing the gap between theoretical peak performance on GPU and the effective performance. There have been many studies on optimizing specific applications on GPU and also a lot of studies on automatic tuning tools. However, the gap between the effective performance and the maximum theoretical performance is often huge. A tighter performance upperbound of an application is needed in order to evaluate whether further optimization is worth the effort. We designed a new approach to compute the CUDA application's performance upperbound through intrinsic algorithm information coupled with low-level hardware benchmarking. Our analysis [11], [22] allows us to understand which parameters are critical to the performance and have more insights of the performance result. As an example, we analyzed the performance upperbound of SGEMM (Single-precision General Matrix Multiply) on Fermi and Kepler GPUs. Through this study, we uncover some undocumented features on Kepler GPU architecture. Based on our analysis, our implementations of SGEMM achieve the best performance on Fermi and Kepler GPUs so far (5 % improvement on average).

6.3. Microarchitecture Performance Analysis

Participants: Ricardo Andrés Velásquez, Pierre Michaud, André Seznec.

6.3.1. Selecting benchmark combinations for the evaluation of multicore throughput

Participants: Ricardo Andrés Velásquez, Pierre Michaud, André Seznec.

In [26], we have shown that fast approximate microarchitecture models such as BADCO [16] can be useful for selecting multiprogrammed workloads for evaluating the throughput of multicore processors. Computer architects usually study multiprogrammed workloads by considering a set of benchmarks and some combinations of these benchmarks. However, there is no standard method for selecting such sample, and different authors have used different methods. The choice of a particular sample impacts the conclusions of a study. Using BADCO, we propose and compare different sampling methods for defining multiprogrammed workloads for computer architecture. We evaluate their effectiveness on a case study, the comparison of several multicore last-level cache replacement policies. We show that random sampling, the simplest method, is robust to define a representative sample of workloads, provided the sample is big enough. We propose a method for estimating the required sample size based on fast approximate simulation. We propose a new method, workload stratification, which is very effective at reducing the sample size in situations where random sampling would require large samples.

6.3.2. A systematic approach for defining multicore throughput metrics

Participant: Pierre Michaud.

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

Measuring throughput is not as straightforward as measuring execution time. This has led to an ongoing debate on what forms a meaningful throughput metric for multi-program workloads. In [29], we present a method to construct throughput metrics in a systematic way: we start by expressing assumptions on job size, job distribution, scheduling, etc., that together define a theoretical throughput experiment. The throughput metric is then the average throughput of this experiment. Different assumptions lead to different metrics, so one should select the metric whose assumptions are close to the real usage he/she has in mind. We elaborate multiple metrics based on different assumptions. In particular, we identify the assumptions that lead to the commonly used weighted speedup and harmonic mean of speedups. Our study clarifies that they are actual throughput metrics, which was recently questioned. We also propose some new throughput metrics, whose calculation sometimes requires approximation. We use synthetic and real experimental data to characterize metrics and show how they relate to each other. Our study can also serve as a starting point if one needs to define a new metric based on specific assumptions, other than the ones we consider in this study. Throughput metrics should always be defined from explicit assumptions, because this leads to a better understanding of the implications and limits of the results obtained with that metric.

6.4. Compiler, vectorization, interpretation

Participants: Erven Rohou, Emmanuel Riou, Arjun Suresh, André Seznec, Nabil Hallou, Alain Ketterlin, Sylvain Collange.

6.4.1. Vectorization Technology To Improve Interpreter Performance

Participant: Erven Rohou.

Recent trends in consumer electronics have created a new category of portable, lightweight software applications. Typically, these applications have fast development cycles and short life spans. They run on a wide range of systems and are deployed in a target independent bytecode format over Internet and cellular networks. Their authors are untrusted third-party vendors, and they are executed in secure managed runtimes or virtual machines. Furthermore, due to security policies, these virtual machines are often lacking just-in-time compilers and are reliant on interpreter execution.

The main performance penalty in interpreters arises from instruction dispatch. Each bytecode requires a minimum number of machine instructions to be executed. In this work we introduce a powerful and portable representation that reduces instruction dispatch thanks to vectorization technology. It takes advantage of the vast research in vectorization and its presence in modern compilers. Thanks to a split compilation strategy, our approach exhibits almost no overhead. Complex compiler analyses are performed ahead of time. Their results are encoded on top of the bytecode language, becoming new SIMD IR (i.e., intermediate representation) instructions. The bytecode language remains unmodified, thus this representation is compatible with legacy interpreters.

This approach drastically reduces the number of instructions to interpret and improves execution time. [15]. SIMD IR instructions are mapped to hardware SIMD instructions when available, with a substantial improvement.

6.4.2. Improving sequential performance: the case of floating point computations

Participants: Erven Rohou, André Seznec, Arjun Suresh.

One way to enhance sequential performance is to consider floating point computations. Languages and instruction sets provide support for only a few representations, namely float and double, and programmers are likely to use the most accurate (unless they handle large data structures). Still, in most cases, programmers do not formally specify the precision they require from their applications, and have no guarantee on the precision they actually get. This is an opportunity for a tradeoff between performance and precision: programs could run faster at the expense of a less accurate result (note that existing compilers already embed some unsafe transformations, for example when flags such as -fast or -ffastmath are used).

The first step consisted in applying memoization to the math library libm. In this case, results are still correct. The performance improvement comes from caching results of pure functions, and retrieving them instead of recomputing a result. This shows good results on floating point intensive benchmarks. In a next step, a helper thread will monitor the patterns of parameters and precompute likely values to "prefetch" results ahead of time.

Reduced precision comes into play when no pattern can be identified, but the new value is close enough to already computed values. We plan to apply interpolation to compute the result faster than the standard code. We will also investigate how we can leverage known properties of mathematical functions, as well as programmer hints about useful properties of user-defined functions, and where reduced precision is acceptable.

6.4.3. Identifying divergence in GPU architectures

Participant: Sylvain Collange.

This research is done in collaboration with Fernando M. Q. Pereira, Diogo Sampaio and Rafael Martins de Souza, UFMG, Brazil.

GPU architectures rely on SIMD execution by vectorizing across SPMD threads. They achieve the best performance when consecutive threads take the same paths through conditional branches and access contiguous memory locations. Thus, many GPU code optimizations that target the control flow or memory access patterns necessitate accurate information about which branches and memory accesses are divergent across threads.

To enable such optimizations, we proposed divergence analysis, a compiler pass that identifies similarities in the control flow and data flow of concurrent threads [37]. This static analysis identifies program variables that are affine functions of the thread identifier and propagate this knowledge to conditional branches and memory accesses. Our analysis consistently outperforms other comparable analyses, thanks to the combination of taking into account affine relations between variables and accurately modeling control dependencies.

6.4.4. 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 proposed 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 [38].

On-Stack-Replacement has been previously proposed to recompile functions while they run. However, it relies on compiler-generated switch points. We proposed a new technique to recompile functions at arbitraty points, thus reinforcing the Obfuscating JIT approach. A prototype is being developed [27].

A new obfuscation technique based of decomposition of CFGs into threads has been proposed. We exploit the mainstream multi-core processing in these systems to substantially increase the complexity of programs, making reverse engineering more complicated. The novel method automatically partitions any serial thread into an arbitrary number of parallel threads, at the basic-block level. The method generates new control-flow graphs, preserving the blocks' serial successor relations and guaranteeing that one basic-block is active at a time through using guards. The method generates m^n different combinations for m threads and n basic-blocks, significantly complicating the execution state. We also provide proof of correctness for the method.

6.4.5. Padrone

Participants: Erven Rohou, Alain Ketterlin, Emmanuel Riou.

The objective of the ADT PADRONE is to design and develop a platform for re-optimization of binary executables at run-time. Development is ongoing, and an early prototype is functional. In [24], we described the infrastructure of Padrone, and showed that its profiling overhead is minimum. We illustrated its use through two examples. The first example shows how a user can easily write a tool to identify hotspots in their application, and how well they perform (for example, by computing the number of executed instructions per cycle). In the second example, we illustrate the replacement of a given function (typically a hotspot) by an optimized version, while the program runs.

We believe PADRONE fills an empty design point in the ecosystem of dynamic binary tools.

6.4.6. Dynamic Analysis and Re-Optimization

Participants: Erven Rohou, Emmanuel Riou, Nabil Hallou, Alain Ketterlin.

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

Dynamic binary analysis and re-optimization is specially interesting for legacy or commercial applications, but also in the context of cloud deployment, where actual hardware is unknown, and other applications competing for hardware resources can vary.

Initial results show that we are able to identify function hotspots that contain vectorized code for the Intel SSE extension, analyze them, and reoptimize the loops to target the latest and more powerful AVX ISA extension.

6.4.7. 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. Conventional wisdom attributes a significant penalty to this branch, due to its high misprediction rate. We revisit this assumption [36], considering current interpreters, and modern predictors. Using both hardware counters and simulation, we show that the accuracy of indirect branch prediction is no longer critical for interpreters. We also compare the characteristics of these interpreters and analyze why the indirect branch is less important than before.

6.5. WCET estimation

Participants: Damien Hardy, Benjamin Lesage, Hanbing Li, Isabelle Puaut, Erven Rohou, André Seznec.

Predicting the amount of resources required by embedded software is of prime importance for verifying that the system will fulfill its real-time and resource constraints. A particularly important point in hard real-time embedded systems is to predict the Worst-Case Execution Times (WCETs) of tasks, so that it can be proven that tasks temporal constraints (typically, deadlines) will be met. Our research concerns methods for obtaining automatically upper bounds of the execution times of applications on a given hardware. Our new results this year are on (i) multi-core architectures (ii) WCET estimation for faulty architectures (iii) traceability of flow information in compilers for WCET estimation.

6.5.1. WCET estimation and multi-core systems

6.5.1.1. Predictable shared caches for mixed-criticality real-time systems

Participants: Benjamin Lesage, Isabelle Puaut, André Seznec.

The general adoption of multi-core architectures has raised new opportunities as well as new issues in all application domains. In the context of real-time applications, it has created one major opportunity and one major difficulty. On the one hand, the availability of multiple high performance cores has created the opportunity to mix on the same hardware platform the execution of a complex critical real-time workload and the execution of non-critical applications. On the other hand, for real-time tasks timing deadlines must be met and enforced. Hardware resource sharing inherent to multicores hinders the timing analysis of concurrent tasks. Two different objectives are then pursued: enforcing timing deadlines for real-time tasks and achieving highest possible performance for the non-critical workload.

In this work, we suggest a hybrid hardware-based cache partitioning scheme that aims at achieving these two objectives at the same time. Plainly considering inter-task conflicts on shared cache for real-time tasks yields very pessimistic timing estimates. We remove this pessimism by reserving private cache space for real-time tasks. Upon the creation of a real-time task, our scheme reserves a fixed number of cache lines per set for the task. Therefore uniprocessor worst case execution time (WCET) estimation techniques can be used, resulting in tight WCET estimates. Upon the termination of the real-time task, this private cache space is released and made available for all the executed threads including non-critical ones. That is, apart the private spaces reserved for the real-time tasks currently running, the cache space is shared by all tasks running on the processor, i.e. non-critical tasks but also the real-time tasks for their least recently used blocks. Experiments show that the proposed cache scheme allows to both guarantee the schedulability of a set of real-time tasks with tight timing constraints and enable high performance on the non-critical tasks.

This work is the main contribution of the PhD thesis of Benjamin Lesage [12].

6.5.1.2. WCET estimation for massively parallel processor arrays

Participant: Isabelle Puaut.

This is joint work with Dumitru Potop-Butucaru, Inria, EPI AOSTE.

Classical timing analysis techniques for parallel code isolates micro-architecture analysis from the analysis of synchronizations between cores by performing them in two separate analysis phases (WCET – worst-case execution time – and WCRT – worst-case response time analyses). This isolation has its advantages, such as a reduction of the complexity of each analysis phase, and a separation of concerns that facilitates the development of analysis tools. But isolation also has a major drawback: a loss in precision which can be significant. To consider only one aspect, to be safe the WCET analysis of each synchronization-free sequential code region has to consider an undetermined micro-architecture state. This may result in overestimated WCETs, and consequently on pessimistic execution time bounds for the whole parallel application. The contribution of this work [33], [23] is an *integrated* WCET analysis approach that considers at the same time micro-architectural information and the synchronizations between cores. This is achieved by extending a state-of-the-art WCET estimation technique and tool to manage synchronizations and communications between the sequential threads running on the different cores. The benefits of the proposed method are twofold. On the one hand, the micro-architectural state is not lost between synchronization-free code regions running on the same core, which results in tighter execution time estimates. On the other hand, only one tool is required for the temporal validation of the parallel application, which reduces the complexity of the timing validation toolchain.

Such a holistic approach is made possible by the use of deterministic and composable software and hardware architectures (homogeneous multi-cores without cache sharing, static assignment of the code regions on the cores). We demonstrate the interest of the approach using an adaptive differential pulse-code modulation (*adpcm*) encoder where the integrated WCET approach provides significantly tighter response time estimations than the more classical WCRT approaches, with a gain of 21% on average.

6.5.2. WCET estimation for architectures with faulty caches

Participants: Damien Hardy, Isabelle Puaut.

Semiconductor technology evolution suggests that permanent failure rates will increase dramatically with scaling, in particular for SRAM cells. While well known approaches such as error correcting codes exist to recover from failures and provide fault-free chips, they will not be affordable anymore in the future due to their non-scalable cost. Consequently, other approaches like fine grain disabling and reconfiguration of hardware elements (e.g. individual functional units or cache blocks) will become economically necessary. This fine-grain disabling will lead to degraded performance compared to a fault-free execution.

A common implicit assumption in all static worst-case execution time (WCET) estimation methods is that the hardware is not subject to faults. Their result is not safe anymore when using fine grain disabling of hardware components, which degrades performance.

In [21] a method that statically calculates a probabilistic WCET bound in the presence of permanent faults in instruction caches is provided. The method, from a given program, cache configuration and probability of cell failure, derives a probabilistic WCET bound. The proposed method, because it relies on static analysis, is guaranteed to identify the longest program path, its probabilistic nature only stemming from the presence of faults. The method is computationally tractable because it does not require an exhaustive enumeration of the possible locations of faulty cache blocks. Experimental results show that it provides WCET estimates very close to, but never below, the method that derives probabilistic WCETs by enumerating all possible locations of faulty cache blocks. The proposed method not only allows to quantify the impact of permanent faults on WCET estimates, but also can be used in architectural exploration frameworks to select the most appropriate fault management mechanisms.

6.5.3. 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 though 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 a first step, we have considered the issue of conveying information through the compilation flow, without any optimization. We have created our own WCET information type and used the annotation files FFX (Flow Fact in XML, provided by IRIT, partner of the W-SEPT project), and applied them to the LLVM compiler framework. We are currently studying the impact of optimizations on the traceability of annotations. We are currently designing a framework for flow fact transformation for a large panel of compiler optimizations.

6.6. HPC and mobile computing

Participant: François Bodin.

We have initiated a research action on the interaction between mobile computing and HPC. We aim at studying data representation linked to parallel programming in heterogeneous systems. In particular, we want to explore energy tradeoffs when changing hardware resources from a light mobile platform to remote execution in a datacenter.

As a test case, we are developing an application for inventorying art pieces in the public domain. This is done in collaboration with University of Rennes 2. This test case is a pluridisplinary collaboration whose goal for University of Rennes 2 is to study how mobile computing can contribute to art studies and dissemination.

6.7. Application-specific number systems

Participant: Sylvain Collange.

This research is done in collaboration with Mark G. Arnold, XLNS Research, USA.

Reconfigurable FPGA platforms let designers build efficient application-specific circuits, when the performance or energy efficiency of general-purpose CPUs is insufficient, and the production volume is not enough to offset the very high cost of building a dedicated integrated circuit (ASIC). One way to take advantage of the flexibility offered by FPGAs is to tailor arithmetic operators for the application. In particular, the Logarithmic Number System (LNS) is suitable for embedded applications dealing with low-precision, high-dynamic range numbers.

Like floating-point, LNS can represent numbers from a wide dynamic range with constant relative accuracy. However, while standard floating-point offer so-called subnormal numbers to represent numbers close to zero with constant absolute accuracy, LNS numbers abruptly overflow to zero, resulting in a gap in representable numbers close to zero that can impact the accuracy of numerical algorithms.

We proposed a generalization of LNS that incorporate features analogous to subnormal floating-point [18], [28]. The Denormal LNS (DLNS) system we introduce defines a class of hybrid number systems that offer quasi-constant absolute accuracy close to zero and quasi-constant relative accuracy on larger numbers. These systems can be configured to range from pure LNS (constant relative accuracy) to fixed-point (constant absolute accuracy across the whole range).

ATEAMS Project-Team

5. New Results

5.1. Empirical analyses of source code

Rascal was used to perform empirical investigations of existing source code bases. First of all, Davy Landman performed an analysis of project management source code to investigate if domain knowledge is present in source code and, if so, how easy it is to extract that knowledge [26]. An earlier experiment in static analysis of PHP code was finalized by Mark Hills. The result is a deep study of feature usage in a large number of well-known PHP projects [25]. Vadim Zaytsev conducted an experiment to recognize micro-patterns in grammars and meta-models [32]. Finally, Jeroen van den Bos performed a deep empirical study to find out as to how far a domain-specific language facilitates evolution [34]. The results showed that the Derric DSL did indeed cover most evolution scenarios, but there is still room for improving the language. In all cases Rascal proved to be instrumental in performing the experiments.

5.2. Better parsing and disambiguation

Ali Afroozeh worked on a new implementation of GLL parsing, called Iguana. Unlike traditional parser generators, Iguana adopted the interpretive approach that is also used in the Ensō parser. This experiment is still ongoing, but the new parser is expected to be integrated into Rascal beginning of 2014. Additionally, a longstanding problem of disambiguation using operator precedence was solved [23]. Traditional approaches are either not safe (i.e. they make the language smaller), or they do not support complex precedence rules as found in, for instance, OCaml.

5.3. Extensible Programming

Modular and extensible implementation of languages could have major impact on how DSLs will be implemented. Anastasia Izmaylova continued here work on improving the extensibility of Rascal's module system, by providing open recursive function combinators.

Extensible programming is traditionally plagued by what has become known as "the expression problem", which captures the fact that most programming languages either support extension of data variants, or extension of operations, but not both. Object Algebras are simple solution to this problem. In [30] we have extended this model to support feature-oriented programming. These results are currently being integrated into the Ensō system.

5.4. DSLs for Games

In collaboration with the Hogeschool van Amsterdam, Riemer van Rozen developed a workbench for MicroMachinations, a DSL for game economies [28]. Completely built using Rascal, this DSL environment features syntax highlighting, static analysis, interactive simulation, and SPIN-based model-checking of process models describing the economy of a game. The project shows the versatility of Rascal as a language workbench for the development of DSLs.

5.5. DSLs for Questionnaires

In the context of computational auditing we have intensified our research on DSLs for questionnaires. It was proposed by Tijs van der Storm as the benchmark task for the Language Workbench Challenge 2013 (LWC'13), which has resulted in a thorough overview and qualitative comparison of language workbenches [24]. As a side-effect, there are now two publicly available Rascal implementations of the questionnaire DSL (QL-R-Kemi and Demoqles). A first step has been made to collect all implementations to create a "chrestomathy" for further study and dissemination of language workbench concepts and DSL implementation patterns. Other results include a formal semantics of the dynamics of questionnaires [21], and an initial prototype of a questionnaire model for modeling the Dutch Tax Income filing application by Pablo Inostroza Valdera.

5.6. Live Programming

Live programming aims to bring the dynamic execution of programs closer to the programmer, ideally almost obliterating the gap between editing and executing the program. We are working on applying such principles in the context of DSLs. This has lead to two results: a live programming environment for a DSL for questionnaires [36], and Trinity, a data-driven IDE for Derric [35]. Riemer van Rozen has worked on applying similar techniques to MicroMachinations, so that game economies can be adapted at runtime.

5.7. Visualization and interaction

Atze van der Ploeg worked on designing new algorithms and abstractions in the domain of visualization and abstraction. His first result is a fast algorithm for drawing non-layered, tidy trees [20]. DeForm is a library for the declarative specification of resolution-independent 2D graphics [27]. In [31] he proposed a reformulation of the traditional functional reactive programming (FRP) framework, which is both simple and efficient to implement.

5.8. Guarded Coroutines

Anastasia Izmaylova and Paul Klint have built an initial version of a compiler for Rascal. The performance improvements with respect to the interpreter are impressive. Moreover, the design of compiler is based on a new construct for implementing languages with complex backtracking and pattern matching semantics: guarded coroutines. This construct will be instrumental in extending the Rascal language with new kinds of control-flow and concurrency.

5.9. Data structures for meta programming

The efficiency of many meta programs is dependent on the internal data structures used to represent collections, trees, relations etc. Michael Steindorfer has worked on comparing the performance of various persistent collection libraries (e.g., those used in Rascal, Clojure, and Scala). This has lead to a redesign of the PDB collection library that underlies the data structures of Rascal. Furthermore, he developed the Orpheus tool, an object redunancy profiler to assess the effects of maximal sharing.

CAIRN Project-Team

6. New Results

6.1. Reconfigurable Architecture Design

6.1.1. Arithmetic Operators for Cryptography and Fault-Tolerance

Participants: Arnaud Tisserand, Emmanuel Casseau, Thomas Chabrier, Karim Bigou, Franck Bucheron, Jérémie Métairie, Nicolas Veyrat-Charvillon, Nicolas Estibals.

Arithmetic Operators for Fast and Secure Cryptography. 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). But fast recoding methods require pre-computations: multiples of base point or off-line conversion. In paper [42] presented at ARITH, we presented a multi-base (e.g. (2,3,5,7)) recoding method for ECC scalar multiplication based on i) a greedy algorithm starting least significant terms first, ii) cheap divisibility tests by multi-base elements and iii) fast exact divisions by multi-base elements. Multi-base terms are obtained on-the-fly using a special recoding unit which operates in parallel to curve-level operations and at very high speed. This ensures that all recoding steps are performed fast enough to schedule the next curve-level operations without interruptions. The proposed method can be fully implemented in hardware without pre-computations. We report FPGA implementation details and very good performance compared to state-of-art results. A specific version of our method allows random recodings of the scalar which can be used as a partial counter-measure against side-channel attacks. The PhD thesis defended by Thomas Chabrier [18] deals with MBNS and other types of arithmetic recodings for ECC scalar multiplication (title: "Arithmetic recodings for ECC cryptoprocessors with protections against side-channel attacks").

In the paper [67], presented at ComPAS, we presented efficient arithmetic operators for divisibility tests and modulo operations for large operands (e.g. 160-600 bits like in cryptographic applications) and by a set of small constants such as $(2^a, 3, 5, 7, 9)$ where $1 \leq a \leq 12$. These operators have been validated and implemented on FPGAs.

In the paper [39] presented at CHES, we described a new RNS modular inversion algorithm based on the extended Euclidean algorithm and the plus-minus trick. In our algorithm, comparisons over large RNS values are replaced by cheap computations modulo 4. Comparisons to an RNS version based on Fermat's little theorem were carried out. Comparisons to a version based on Fermat's little theorem were carried out. The number of elementary modular operations is significantly reduced: a factor 12 to 26 for multiplications and 6 to 21 for additions. Virtex 5 FPGAs implementations show that for a similar area, our plus-minus RNS modular inversion is 6 to 10 times faster. Other implementation results of RNS for ECC cryptosystems have been presented in [75] and [74].

ECC 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.

Arithmetic Operators for Fault Tolerance. In the ARDyT project, 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.

6.1.2. Reconfigurable Processor Extensions Generation

Participants: Christophe Wolinski, François Charot.

Most proposed techniques for automatic instruction sets extension usually dissociate pattern selection and instruction scheduling steps. The effects of the selection on the scheduling subsequently produced by the compiler must be predicted. This approach is suitable for specialized instructions having a one-cycle duration because the prediction will be correct in this case. However, for multi-cycle instructions, a selection that does not take scheduling into account is likely to privilege instructions which will be, *a posteriori*, less interesting than others in particular in the case where they can be executed in parallel with the processor core. The originality of our research work is to carry out specialized instructions selection and scheduling in a single optimization step. This complex problem is modeled and solved using constraint programming techniques. This approach allows the features of the extensible processor to be taken into account with a high degree of flexibility. Different architectures models can be envisioned. This can be an extensible processor tightly coupled to a hardware extension having a minimal number of internal registers used to store intermediate results, or a VLIW-oriented extension made up of several processing units working in parallel and controlled by a specialized instruction. These techniques have been implemented in the Gecos source-to-source framework.

Novel techniques addressing the interactions between code transformation (especially loops) and instruction set extension are under study. The idea is to automatically transform the original loop nests of a program (using the polyhedral model) to select specialized and vector instructions. These new instructions may use local memories located in the hardware extension and used to store intermediates data produced at a given loop iteration. Such transformations lead to patterns whose effect is to significantly reduce the pressure on the memory of the processor. An experiment realized on the matrix multiplication (extracted from PolyBench/C, the polyhedral benchmark suite) using an Xtensa extensible and configurable processor from Tensilica shows interesting speedups. Speedup of 4.3 for the transformed code compared to the initial code for matrices of size 512x512 and speedup of 8.75 (respectively 20.15) in case of an extension allowing SIMD vector operations on vector of 4 32-bit words (respectively 16 32-bit words) are observed.

6.1.3. Runtime Mapping of Hardware Accelerators on the FlexTiles 3D Self-Adaptive Heterogeneous Manycore

Participants: Olivier Sentieys, Antoine Courtay, Christophe Huriaux.

FlexTiles is a 3D stacked chip with a manycore layer and a reconfigurable layer. This heterogeneity brings a high level of flexibility in adapting the architecture to the targeted application domain for performance and energy efficiency. A virtualisation layer on top of a kernel hides the heterogeneity and the complexity of the manycore and fine-tunes the mapping of an application at runtime. The virtualisation layer provides self-adaptation capabilities by dynamically relocation of application tasks to software on the manycore or to hardware on the reconfigurable area. This self-adaptation is used to optimize load balancing, power consumption, hot spots and resilience to faulty modules. The reconfigurable technology is based on a Virtual Bit-Stream (VBS) that allows dynamic relocation of accelerators just as software based on virtual binary code allows task relocation.

We have proposed a novel approach to hardware task relocation in an FPGA-based reconfigurable fabric, allowing offline design, routing, and unfinalized placement of hardware IPs and dynamic placement of the corresponding bit-streams at run-time. Our proposal relies on a custom dual-context FPGA configuration memory organization in a shift-register manner and on a dedicated bit-stream insertion controller leading to a break-through in terms of adaptive capabilities of the reconfigurable hardware. We show that using our custom shift-register organization across the configuration memory, and under some weak constraints, can greatly reduce the overhead implied by the 1-D to 2-D mapping of the shift-register onto the logic fabric. The use of partial dynamic reconfiguration in FPGA-based systems has grown in recent years as the spectrum of applications which use this feature has increased. For these systems, it is desirable to create a series of partial bitstreams which represent tasks that can be located in multiple regions in the FPGA substrate. While the transferal of homogeneous collections of lookup-table based logic blocks from region to region has been

shown to be relatively straightforward, it is more difficult to transfer partial bitstreams which contain fixed function resources, such as block RAMs and DSP blocks. To do so, we explore adding enhancements to the FPGA architecture which allow for the migration of partial bitstreams including fixed resources from region to region even if these fixed function resources are not located in the same position in the region. Our approach does not require significant, time-consuming place-and-route during the migration process. We quantify the cost of inserting additional routing resources into the FPGA architecture to allow for easy migration of heterogeneous, fixed function resources. Our experiments show that this flexibility can be added for a relatively low overhead and performance penalty. As mentioned above, the Virtual Bit-Stream (VBS) is a concept of an unfinalized, pre-routed bit-stream which could be loaded almost anywhere on a custom FPGA logic fabric. Unlike classical bit-streams, the VBS is not tied to a specific location on the circuit, hence its "virtual" qualifier. The goal is to generate a single VBS only once for each and every possible location of the logic fabric in the FPGA in a unfinished manner: the time-consuming packing, place and route steps are done offline and only local routing is done at runtime in order to ensure fast decoding time as well as low memory overhead. The VBS concept is pending for a European patent application.

6.1.4. Power Models of Reconfigurable Architectures

Participants: Robin Bonamy, Daniel Chillet, Olivier Sentieys.

Including a reconfigurable area in complex systems-on-chip is considered as an interesting solution to reduce the area of the global system and to support high performance. But the key challenge in the context of embedded systems is currently the power budget and the designer needs some early estimations of the power consumption of its system. Power estimation for reconfigurable systems is a difficult issue since several parameters need to be taken into account to define an accurate model.

One first parameter concerns the choice of tasks to execute and their allocation in the computing resources. Indeed, several hardware implementations of an algorithm can be obtained and exploited by the operating system for a flexible allocation of tasks to optimize energy consumption. These different hardware implementations can be obtained by varying the parallelism level, which has a direct impact on area and execution time and therefore on power and energy consumption. To highlight this point, we made several evaluations of delay, area, power, and energy impacts of loop transformations using High Level Synthesis tools. Real power measurements have been made on an FPGA platform and for different task implementations to build a model of energy consumption versus execution time.

Furthermore, we also considered the opportunity of the dynamic reconfiguration, which makes possible to partially reconfigure a specific part of the circuit while the rest of the system is running. This opportunity has two main effects on power consumption. First, thanks to the area sharing ability, the global size of the device can be reduced and the static (leakage) power consumption can thus be reduced. Secondly, it is possible to delete the configuration of a part of the device which reduces the dynamic power consumption when a task is no longer used.

We analyzed the power consumption during the dynamic reconfiguration on a Virtex 5 board. Three models of the partial and dynamic reconfiguration power consumption with different complexity/accuracy tradeoffs are extracted. These models are used in design space exploration to include impact of reconfiguration on energy consumption of a complete system. We proposed a methodology for power/energy consumption modeling and estimation in the context of heterogeneous (multi)processor(s) and dynamically reconfigurable hardware systems. We developed an algorithm to explore all task mapping possibilities for a complete application (e.g. for H264 video coding) with the aim to extract one of the best solutions with respect to the designer's constraints. This algorithm is a step ahead for defining on-line power management strategies to decide which task instances must be executed to efficiently manage the available power using dynamic partial reconfiguration. All these results are presented in the Robin Bonamy's thesis [17]

6.1.5. Real-time Spatio-Temporal Task Scheduling on 3D Architecture

Participants: Quang-Hai Khuat, Quang-Hoa Le, Emmanuel Casseau, Antoine Courtay, Daniel Chillet.

One of the main advantages offered by a three-dimensional system-on-chip (3D SoC) is the reduction of wire length between different blocks of a system, thus improving circuit performance and alleviating power overheads of on-chip wiring. To fully exploit this advantage, an efficient management referring to allocate temporarily the tasks at different levels of the architecture is greatly important.

In the context of 3D SoC, we have developed several spatio-temporal scheduling algorithms for 3D MultiProcessor Reconfigurable System-on-Chip (3DMPRSoC) architectures composed of a multiprocessor layer and an embedded Field Programmable Gate Array (eFPGA) layer with dynamic reconfiguration. These two layers are interconnected vertically by through-silicon vias (TSVs) ensuring tight coupling between software tasks on processors and associated hardware accelerators on the eFPGA. Our algorithms cope with task dependencies and try to allocate communicating tasks close to each other in order to reduce direct communication cost, thus reducing global communication cost.

In the 3DMPRSoC context, our algorithms favor direct communications including: i) point-to-point communication between hardware accelerators on the eFPGA, ii) communication between software tasks through the Network-on-Chip of the multiprocessor layer, and iii) communication between software task and accelerator through TSV. When a direct communication between two tasks occurs, the data are stored in a shared memory placed onto the multiprocessor layer.

Our work in [68] takes all types of communication into consideration and proposes a scheduling and placement strategy of tasks reducing the global communication cost to 17% compared with our previous algorithm based on Pfair. In this work, the eFPGA layer of the 3DMPRSoC is supposed to contain homogeneous partial reconfiguration regions (PRR) and the size of a hardware accelerator is limited by the size of a PRR. To exceed this limitation, we analyzed the Vertex-List Structure (VLS) method for relocating hardware accelerators of various sizes anywhere onto the eFPGA if resources are available. Then, we proposed VLS-BCF algorithm [49] based on VLS that allows for reducing the overall communication cost significantly – up to 24% – compared to classical methods.

6.1.6. Ultra-Low-Power Reconfigurable Controllers

Participants: Vivek D. Tovinakere, Olivier Sentieys, Steven Derrien.

A key concern in the design of controllers in wireless sensor network (WSN) nodes is the flexibility to execute different control tasks for managing resources, sensing and communications tasks of the node. In this paper, low-power flexible controllers for WSN nodes based on reconfigurable microtasks are presented. A microtask is a digital control unit made up of an FSM and datapath. Scalable architectures for reconfigurable FSMs along with variable precision adders in datapath are proposed for flexible controllers. Power gating as a low power technique is considered for low power operation in reconfigurable microtasks by exploiting coarse grain power gating opportunities in FSMs and adders. Gate-level models are applied to analyze energy savings in logic clusters due to power gating. Power estimation results on typical benchmark microtasks show a $2\times$ to $5\times$ improvement in energy efficiency w.r.t a microcontroller at a cost of $5\times$ when compared with a microtask implemented as an ASIC with higher NRE costs [21].

6.2. Compilation and Synthesis for Reconfigurable Platform

6.2.1. Polyhedral-Based Loop Transformations for High-Level Synthesis

Participants: Steven Derrien, Antoine Morvan, Patrice Quinton, Tomofumi Yuki, Mythri Alle.

After almost two decades of research effort, there now exists a large choice of robust and mature C to hardware tools that are used as production tools by world-class chip vendor companies. Although these tools dramatically slash design time, their ability to generate efficient accelerators is still limited, and they rely on the designer to expose parallelism and to use appropriate data layout in the source program. We believe this can be overcome by tackling the problem directly at the source level, using source-to-source optimizing compilers. More precisely, our aim is to study how polyhedral-based program analysis and transformation can be used to address this problem. In the context of the PhD of Antoine Morvan, we have studied how it was possible to improve the efficiency and applicability of nested loop pipelining (also known as nested software

pipelining) in C to hardware tools. Loop pipelining is a key transformation in high-level synthesis tools as it helps maximizing both computational throughput and hardware utilization.

We have first studied how polyhedral based loop transformations (such as coalescing) could be used to improve the efficiency of pipelining small trip-count inner loops [27] and implemented the transformation in the Gecos source to source toolbox. We also have proposed a technique to widen the applicability of loop pipelining to kernels exposing complex dynamic memory access patterns for which compile time dependency analysis techniques cannot be used efficiently. Our approach borrows from the notion of runtime memory disambiguation used in super scalar processors to add a data dependency hazards detection mechanism to the synthesized circuits. The approach has shown promising results and led to a presentation presented at the 50th ACM/IEEE Design Automation Conference [37]. In addition to our work on nested loop pipelining, we also investigated how to extend existing polyhedral code generation techniques to enable the synthesis of fast and area-efficient control-logic. Our approach was implemented in the Gecos framework and presented at the Field Programmable Technology international conference in late 2013 [63].

6.2.2. Compiling for Embedded Reconfigurable Multi-Core Architectures

Participants: Steven Derrien, Olivier Sentieys, Maxime Naullet, Antoine Morvan, Tomofumi Yuki, Ali Hassan El-Moussawi.

Current and future wireless communication and video standards have huge processing power requirements, which cannot be satisfied with current embedded single processor platforms. Most platforms now therefore integrate several processing core within a single chip, leading to what is known as embedded multi-core platforms. This trend will continue, and embedded system design will soon have to implement their systems on platforms comprising tens if not hundred of high performance processing cores. Examples of such architectures are the Xentium processor from by Recore or the Kahrisma processor, a radically new concept of morphable processor from Karlsruhe Institute of Technology (KIT). This evolution will pose significant design challenges, as parallel programming is notoriously difficult, even for domain experts. In the context of the FP7 European Project Alma (Architecture-oriented parallelization for high performance embedded Multi-core systems using scilAb), we are studying how to help designers programming these platforms by allowing them to start from a specification in Matlab and/or Scilab, which are widely used for prototyping image/video and wireless communication applications. Our research work in this field revolves around two topics. The first one aims at exploring how floating-point to fixed-point conversion can be performed jointly with the SIMD instruction selection stage to explore performance/accuracy trade-off in the software final implementation. The second one aims at exploring how program transformation techniques (leveraging the polyhedral model and/or based on the domain specific semantics of scilab built-in functions) can be used to enable an efficient coarse grain parallelization of the target application on such multi-core machines [30].

6.2.3. Numerical Accuracy Analysis and Optimization

Participants: Olivier Sentieys, Steven Derrien, Romuald Rocher, Pascal Scalart, Tomofumi Yuki, Aymen Chakhari, Gaël Deest.

Most of analytical methods for numerical accuracy evaluation use perturbation theory to provide the expression of the quantization noise at the output of a system. Existing analytical methods do not consider correlation between noise sources. This assumption is no longer valid when a unique datum is quantized several times. In [35], an analytical model of the correlation between quantization noises is provided. The different quantization modes are supported and the number of eliminated bits is taken into account. The expression of the power of the output quantization noise is provided when the correlation between the noise sources is considered. The proposed approach allows improving significantly the estimation of the output quantization noise power compared to the classical approach, with a slight increase of the computation time.

Trading off accuracy to the system costs is popularly addressed as the word-length optimization (WLO) problem. Owing to its NP-hard nature, this problem is solved using combinatorial heuristics. In [56], a novel approach is taken by relaxing the integer constraints on the optimization variables and obtain an alternate noise-budgeting problem. This approach uses the quantization noise power introduced into the system due to

fixed-point word-lengths as optimization variables instead of using the actual integer valued fixed-point word-lengths. The noise-budgeting problem is proved to be convex in the rounding mode quantization case and can therefore be solved using analytical convex optimization solvers. An algorithm with linear time complexity is provided in order to realize the actual fixed-point word-lengths from the noise budgets obtained by solving the convex noise-budgeting problem.

An analytical approach is studied to determine accuracy of systems including unsmooth operators. An unsmooth operator represents a function which is not derivable in all its definition interval (for example the sign operator). The classical model is no longer valid since these operators introduce errors that do not respect the Widrow assumption (their values are often higher than signal power). So an approach based on the distribution of the signal and the noise was proposed. We focused on recursive structures where an error influences future decision (such as Decision Feedback Equalizer). In that case, numerical analysis method (e.g. Newton Raphson algorithm) can be used. Moreover, an upper bound of the error probability can be analytically determined [43]. We also studied the case of Turbo Coder and Decoder to determine data word-length ensuring sufficient system quality.

One of the limitation of analytical accuracy technique is that they are based on a Signal Flow Graph Representation of the system to be analyzed. This SFG model is currently built-out of a source program by flattening its whole control-flow (including full loop unrolling) which raises significant accuracy analysis issues. In 2013 we have started studying how we could bridge numerical analysis techniques with more compact polyhedral program representations to provide a more general and scalable framework.

6.2.4. Design Tools for Reconfigurable Video Coding

Participants: Emmanuel Casseau, Hervé Yviquel.

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 designer provides the application description in the RVC-CAL language, after which the co-design flow automatically generates a network of processors that can be synthesized on FPGA platforms. The processors are based on a low complexity and configurable TTA processor (Very Long Instruction Word -style processor). The architecture model of the platform is composed of processors with their local memories, an interconnection network and shared memories. Both shared and local memories are used to limit the traditional memory bottleneck. Processors are connected together through the shared memories. The design flow is implemented around two open-source toolsets: Orcc (Open RVC-CAL Compiler: http://orcc.sourceforge.net) and TCE (TTA-based Co-design Environment: http://tce.cs.tut.fi). The inputs of the design flow are the RVC application, the platform configuration (i.e. the configuration of the TTA processors and their number), and the mapping specification (i.e. the mapping of the actors onto the processors). Orcc generates a high-level description of the processors, an intermediate representation of the software code associated to each actor, and the processor interconnection requirements. Then TCE uses these informations to generate a complete multi-processor platform design: the VHDL descriptions of the processors using a pre-existing database of hardware components and the executable binary code that will execute the actors on the processors.

This work is done in collaboration with Mickael Raulet from IETR INSA Rennes and has been implemented in the Orcc open-source compiler and with Jarmo Takala team from Tampere University of Technology (Finland) who is involved in the TCE toolset.

6.3. Interaction between Algorithms and Architectures

6.3.1. Design Methodologies for Software Defined Radios

Participants: Matthieu Gautier, Olivier Sentieys, Emmanuel Casseau, Arnaud Carer, Ganda-Stéphane Ouedraogo, Mai-Thanh Tran, Vaibhav Bhatnagar.

Software Defined Radio (SDR) is a flexible signal processing architecture with reconfiguration capabilities that can adapt itself to various air interfaces. It was first introduced by Joseph Mitola as an underlying structure for Cognitive Radio (CR). The FPGA (Field Programmable Gate Array) technology is expected to play a key role in the development of SDR platforms. FPGA-based SDR is a quite old paradigm and we are fronting this challenge while leveraging the nascent High Level Synthesis tools and languages.
Actually, our goal is to propose methods and tools for rapid implementation of new waveforms in the stringent flexibility paradigm. We proposed a novel design flow for FPGA-based SDR applications [38] [70]. This flow relies upon HLS principles and its entry point is a Domain-Specific Language (DSL) which partly handles the complexity of programming an FPGA and integrates SDR features.

6.3.2. Adaptive Precision under Performance Constraints in OFDM Wireless Receivers

Participants: Olivier Sentieys, Matthieu Gautier, Fernando Cladera [Master's Student].

To cope with rapid variations of channel parameters, wireless receivers are designed with a significant performance margin to reach a given Bit Error Rate (BER), even for worst-case channel conditions. 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 an FPGA and an ARM processor using soft SIMD. Results show that up to 66% of the dynamic energy consumption can be saved using this adaptive technique.

6.3.3. MIMO Systems and Cooperative Strategies for Low-Energy Wireless Networks

Participants: Olivier Berder, Olivier Sentieys, Pascal Scalart, Matthieu Gautier, Le-Quang-Vinh Tran, Duc-Long Nguyen [Master's Student], Ruifeng Zhang, Viet-Hoa Nguyen.

Since a couple of years, the CAIRN team has reached a significant expertise in multi-antenna systems, especially in linear precoding. In order to obtain an efficient, simple and general form of precoders, we considered an SNR-like matrix to approximate the minimum distance. The precoding matrix is first parameterized as the product of a diagonal power allocation matrix and an input-shaping matrix and demonstrated that the minimum diagonal entry of the latter is obtained when the input-shaping matrix is a DFT-matrix. The major advantage of this design is that the solution can be available for all rectangular QAM-modulations and for any number of datastreams [28]. On the other hand the sphere decoder was applied at the receiver side instead of maximum likelihood and the performance complexity trade-off was investigated. Some adjustments of traditional sphere decoding algorithm were mandatory to adapt to the precoded MIMO systems [55].

Another way to exploit the MIMO diversity, especially in WSN where only one antenna can be supported by limited size devices, is to use space-time codes in a distributed manner. In this context, a new protocol, called fully distributed space-time coded (FDSTC) protocol having information exchange between relays, was proposed and compared with the conventional distributed space-time coded (DSTC) protocol using non-regenerative relays (NR-relays) and regenerative relays (R-relays). At the same spectral efficiency, FDSTC has better performance in terms of outage probability in high SNR regions. In terms of energy efficiency, the FDSTC protocol is shown to outperform DSTC for long-range transmissions [32]. As very few dedicated MAC protocols exist, we investigated a novel low-latency MAC protocol (ARQ-CRI) for low-power cooperative wireless sensor networks WSNs, while preserving (in high traffic mode) or even increasing (in low traffic mode) energy-efficiency [54]. An energy efficient opportunistic MAC protocol with the mechanisms of reservation and a relay candidate coordination were also proposed, and the multi-relay transmission probability was analyzed. Simulation and experiment results on a real wireless sensor network platform in different channels demonstrated the proposed scheme greatly reduces the multi-relay transmission probability and achieves about 84% improvement of energy efficiency compared with the traditional opportunistic MAC schemes [66].

6.3.4. Energy Harvesting and Adaptive Wireless Sensor Networks

Participants: Olivier Berder, Olivier Sentieys, Arnaud Carer, Mahtab Alam, Ruifeng Zhang, Trong-Nhan Le.

As tiny sensor nodes are equipped with limited battery, the optimization of the power consumption of these devices is extremely vital. In typical WSN platforms, the radio transceiver consumes major proportion of the energy. Major concerns are therefore to decrease both the transmit power and radio activity. We designed an adaptive transmit power optimization technique that is applied under varying channel to reduce the energy per successful transmitted bit. Each node locally adapts its output power according to the signal-to-noise ratio (SNR) variations (for all the neighbor nodes). It is found that by dynamically adapting the transmit power on average can help to reduce the energy consumption by a factor of two [36].

To further extend the system lifetime of WSN, energy harvesting techniques have been considered as potential solutions for long-term operations. Instead of minimizing the consumed energy as for the case of battery-powered systems, the harvesting node is adapted to Energy Neutral Operation (ENO) to achieve a theoretically infinite lifetime. Several types of energy sources can be used, as light, motion or heat [51]. We even investigated the possibility for a single sediment-microbial fuel cell (MFC) to power a wireless sensor network [31]. Through experiments conducted on the PowWow platform, it was shown that the energy harvesting device adapts to the intermittent power supplied by the MFC, and the radio-transmitter is able to switch from a continuous to degraded mode. Given the harvesting capability, we then tried to design power managers (PM) able to optimize the quality of service of WSN while maintaining ENO. Our PM adapts the duty cycle of the node according to the estimation of harvested energy and the consumed energy provided by a simple energy monitor for a super capacitor based WSN to achieve the ENO [52]. When possible, as is sometimes the case for solar or wind energy, it is also of prime interest to benefit from an accurate energy predictor to estimate the energy that can be harvested in the near future, therefore we proposed a low complexity energy predictor using adaptive filter [53]. Finally, with colleagues from University College of Cork, we recently investigated the possibility to combine energy harvesting platforms with low power wake-up radios. A nano-watt wake-up radio receiver (WUR) was used cooperatively with the main transceiver in order to reduce the wasted energy of idle listening in asynchronous MAC protocols, while still maintaining the same reactivity [50].

6.3.5. Impact of RF Front-End Nonlinearity on WSN Communications.

Participants: Amine Didioui, Olivier Sentieys, Carolynn Bernier [CEA Leti].

In the context of a collaboration with CEA Leti, we studied the impact of RF front-end non-linearity on the performance of wireless sensor networks (WSN). More specifically, we investigated the problem of interference caused by intermodulation between in-band interferers. We analyzed this problem using an enhanced model of signal-to-interference-and-noise ratio (SINR) that includes an interference term due to intermodulation. Using a WSN simulator and the selectivity and the third-order input intercept point (IIP3) specifications of a radio transceiver, we have shown that the new SINR model provides helpful information for the analysis of intermodulation problems caused by in-band signals in IEEE 802.15.4 WSNs. In [45], we presented a reconfigurable receiver model whose purpose is to enable the study of reconfiguration strategies for future energy-aware and adaptive transceivers. This model is based on Figure of Merits of measured circuits. To account for real-life RF interference mechanisms, a link quality estimator is also provided. We show that adapting the receiver performance to the channel conditions can lead to considerable power saving. The models proposed can easily be implemented in a wireless network simulation in order to validate the value of a reconfigurable architecture in real-world deployment scenarios.

6.3.6. HarvWSNet: A Co-Simulation Framework for Energy Harvesting Wireless Sensor Networks.

Participants: Amine Didioui, Olivier Sentieys, Carolynn Bernier [CEA Leti].

Recent advances in energy harvesting (EH) technologies now allow wireless sensor networks (WSNs) to extend their lifetime by scavenging the energy available in their environment. While simulation is the most widely used method to design and evaluate network protocols for WSNs is simulation, existing network simulators are not adapted to the simulation of EH-WSNs and most of them provide only a simple linear battery model. To overcome these issues, we have proposed HarvWSNet, a co-simulation framework based on WSNet and Matlab that provides adequate tools for evaluating EH-WSN lifetime [44]. Indeed, the framework allows for the simulation of multi-node network scenarios while including a detailed description of each node's energy harvesting and management subsystem and its time-varying environmental parameters. A case study based on a temperature monitoring application has demonstrated HarvWSNet's ability to predict network lifetime while minimally penalizing simulation time [40].

6.3.7. Synchronisation Algorithms and Parallel Architecture for Wireless and High-Rate Optical OFDM Systems

Participants: Pramod Udupa, Olivier Sentieys, Arnaud Carer, Pascal Scalart.

Multi-band Coherent Optical OFDM (MB CO-OFDM) is widely predicted to be one of the technologies which will empower 100 Gigabit Ethernet (100GbE) networks. CO-OFDM uses coherent technology and advanced digital signal processing (DSP) to achieve net data rate of 10 Gbps in a single band. This strict throughput requirement puts a constraint on the kind of signal processing algorithms and architectures used for building the system. In [72], a scalable parallel architecture using radix-2^2 for IFFT was proposed. The second proposal consists of a scalable parallel timing synchronization algorithm which can support very high input rates at the receiver. MOPS count as well as area versus throughput for the synchronization algorithm are provided for the OFDM transceiver to show the improvements due to proposed architecture. Architecture exploration was performed using a leading-edge high-level synthesis (HLS) tool.

A novel low complexity parallel algorithm and its associated architecture were proposed for initial synchronization in orthogonal frequency division multiplexing (OFDM) systems. The method is hierarchical and uses auto-correlation for the first step and cross-correlation for the second step [60]. The main advantage of the proposed approach is that it reduces the computational complexity by a factor of five (80%), while achieving similar mean square error (MSE) as cross-correlation based methods. The method uses block-level parallelism for auto-correlation step, which speeds up the computation significantly. After fixed-point analysis, a parallel architecture is proposed to accelerate both coarse and fine synchronization steps. This parallel architecture is scalable and provides speed-up proportional to number of parallel blocks [59].

CAMUS Team

6. New Results

6.1. VMAD and APOLLO

The goal of the APOLLO project is to provide a set of annotations (pragmas) that the user can insert in the source code to perform advanced analyses and optimizations, for example dynamic speculative parallelization. It is based on the prototype VMAD developed previously by the team between 2009 and 2012.

APOLLO includes a modified LLVM compiler and a runtime system. The program binary files are first generated by our compiler to include necessary data, instrumentation instructions, parallel code skeletons, and callbacks to the runtime system which is implemented as a dynamic library. External modules associated to specific analyses and transformations are dynamically loaded when required at runtime.

APOLLO uses sampling and multi-versioning to limit the runtime overhead (profiling, analysis, and code generation). At runtime, targeted codes are launched by successive chunks that can be either original, instrumented or optimized/parallelized versions. After each chunk execution, decisions can be taken relatively to the current optimization strategy. APOLLO is handling advanced memory access profiling [26], [17] through linear interpolation of the addresses, dynamic dependence analysis [18], version selection [26] and speculative polyhedral parallelization [22], [17].

Alexandra Jimborean defended her PhD thesis on this topic in 2012 [25].

In 2012, Aravind Sukumaran-Rajam started his PhD in our team to extend this work in order to handle more general programs which do not exhibit a pure polyhedral memory behavior. The investigated approach will explore approximative modelling of dependences still allowing advanced optimizing transformations of loop nests. A main issue concerns speculation verification when using approximative modelling.

Juan Manuel Martinez started his PhD in our team in 2013, with the goal of improving the flexibility of the parallel code generation phase inside Apollo. Indeed, although code skeletons are a good solution to fast dynamic parallel code generation, their shapes limit the kind of optimizing transformations that may be applied at runtime. Juan Manuel's work consists in defining elementary code skeletons that may be assembled at runtime to form a large panel of possible codes. These elementary skeletons will be defined as the objects forming the Apollo specific intermediate representation. Juan Manuel Martinez is a former master student of the University of Buenos Aires, Argentina (associate team EA-Ancome), and has already been working on VMAD to make the code generation support tiling. He defended his master thesis on this subject in October 2013 at the University of Buenos Aires.

Jean-Fran cois Dollinger will extend the framework to handle heterogeneous architectures (GPGPUs) in 2014. Willy Wolff, a master student from the University of Strasbourg, joined the APOLLO group in September 2013. His work is to implement just-in-time compilation in the APOLLO framework.

6.2. The Multifor programming construct

We have proposed a new programming control structure called "multifor", allowing to take advantage of optimization and parallelization opportunities that are not easily attainable using the standard programming structures.

In a multifor-loop, several loops whose bodies are run in interleaved fashion can be defined. Respective iteration domains are mapped onto each other according to a run frequency – the grain – and a relative position – the offset. Imen Fassi developped a source-to-source compiler called IBB (Iterate-But-Better) which is automatically translating any C source code containing multifor-loops into an equivalent source code where multifor-loops have been transformed into equivalent for-loops. Traditional polyhedral software tools, and particularly CLooG [21], are used to generate the corresponding code. Additionally, a promising perspective related to non-linear mapping of iteration spaces has also been developed, yielding to run a loop nest inside any other one by solving the problem of inverting "ranking Ehrhart polynomials".

This work is the PhD work of Imen Fassi, who started her work in 2013 and who is co-advised by Yosr Slama, Assistant Professor at the University El Manar in Tunis, Tunisia, and Philippe Clauss. A first paper [15] on this topic has been published at the IMPACT workshop that was held in conjunction with the HIPEAC conference in Berlin, Germany, in January 2013. Another paper describing the IBB compiler and showing the efficiency of multifor codes has been submitted to an international conference.

Obviously, reasoning on such a syntactic sugar suppose an associated precise and unambiguous meaning. Therefore a denotational semantics has been defined that resolves all such semantic issues and that is well-suited to prove code transformations. It has been presented to the French community of Compilation during the sixth meeting in Annecy [8].

6.3. CPU+GPU adaptive computation

In this work, we aim to automatically use CPU and GPU to jointly execute a parallel code. To ensure load balance between different PUs, thus to preserve performance, it is necessary to consider the underlying hardware and the program parameters. Compiler optimizations, execution context, hardware availability and specification make it difficult to determine execution times statically. To overcome this hurdle we rely on a portable and automatic method for predicting execution times of statically generated codes on multicore CPUs and on CUDA GPUs. This approach relies on three stages: automatic code generation, offline profiling and online prediction.

This is the latest result of PhD student Jean-François Dollinger, advised by Vincent Loechner since 2011. Preliminary results, a "fastest-wins" algorithm between a multicore CPU and the best predicted GPU code version, was published in 2013 in ICPP [14]. We are currently writing a conference paper presenting the latest advances, and preparing a journal paper to be submitted in 2014, before Jean-François Dollinger's PhD defense by the end of the year.

6.4. Minimizing the synchronization overhead of X10 programs

The CAMUS team has for long focused on compiling, optimizing, and parallelizing *sequential* programs. The project described in this section is somewhat unusual in this context, in that it targets programs written in an explicitly parallel language, and applies polyhedral modeling techniques to reschedule computations, effectively introducing parallel-to-parallel program transformations. This work has been done in collaboration with the Inria COMPSYS team at ENS Lyon, and first results will be presented at the *Compiler Construction* conference (CC'14) in April 2014.

The need to leverage the computing power of multi-core processors (and distributed computers) has lead to the design of explicitly parallel programming languages. Such languages often employ a fork/join model, and include syntax to launch and synchronize tasks (also called activities) with well-defined semantics. This brings parallel constructions under the control of the compiler, and introduces new optimization opportunities. Our work has focused on the various synchronization primitives available to the programmer, and more specifically on how one type of synchronization can be replaced with another for specific classes of programs, the goal being to minimize the synchronization overhead. We have demonstrated significant speedups on programs written using the X10 programming language, and have obtained similar results on equivalent Habanero-Java programs.

More specifically, our proposed optimization works by eliminating the use of clocks in X10 programs whose activities can be characterized with a polyhedral time-domain. The X10 language basically has two activity synchronization primitives: one is the explicit use of "clocks" (synchronization barriers) during activity execution, the other is the implicit use of activity containers that synchronize only on the end of activities. Under reasonable conditions on the patterns of activity creation and control, we have shown that long-running activities using clocks can be replaced by short-lived activities synchronized only on the end of their containers, and that this transformation provides a significant gain at run time. This work has two main contributions. First, it extends a known transformation framework to the case where the original program is already parallel.

[8] http://compilation.gforge.inria.fr/2013_04_Annecy

Second, it shows that the polyhedral model has applications far beyond its current use in data dependence and memory locality analyzes. This work also opens up new research directions. First, it turns out that our transformation is far more general than the use we currently make of it, and therefore that it provides a solid basis for other optimizations of parallel programs. Second, the polyhedral model we have developed provides an immediate cost model for synchronization primitives, which is not used in our current work, but may provide sound heuristics to adapt the optimization phase to the characteristics of specific run time components. We plan to explore these aspects in the near future.

This work has been done in collaboration with Paul Feautrier, member of the COMPSYS Inria team, in ENS Lyon. The CAMUS team has invited Paul Featrier for one week in June 2013 in Strasbourg. We are currently seeking funding to organize more frequent stays at either Lyon or Strasbourg.

This work has been invited for presentation at the LCPC workshop held in Lyon in July 2013 (http://labexcompilation.ens-lyon.fr/cpc2013). An extended version of this work has been accepted for publication at the *Compiler Construction* conference, to be held in April 2014.

6.5. Switcheable scheduling

Parallel applications used to be executed alone until their termination on partitions of supercomputers. The recent shift to multicore architectures for desktop and embedded systems is raising the problem of the coexistence of several parallel programs. Operating systems already take into account the *affinity* mechanism to ensure a thread will run only onto a subset of available processors (e.g., to reuse data remaining in the cache since its previous execution). But this is not enough, as demonstrated by the large performance gaps between executions of a given parallel program on desktop computers running several processes. To support many parallel applications, advances must be made on the system side (scheduling policies, runtimes, memory management...). However, automatic optimization and parallelization can play a significant role by generating programs with dynamic-auto-tuning capabilities to adapt themselves to the complete execution context, including the system load.

Our approach is to design at compile-time programs that can adapt at run-time to the execution context. The originality of our solution is to rely on *switcheable scheduling*, a selected set of program restructuring which allows to swap between program versions at some meeting points without backtracking. A first step selects pertinent versions according to their performance behavior on some execution contexts. The second step builds the auto-adaptive program with the various versions. Then at runtime the program selects the best version by a low overhead sampling and profiling of the versions, ensuring every computation is useful.

This work is an addition to the research directions of CAMUS related to dynamic optimization. It has been started at Paris-Sud University by Cédric Bastoul before he joined CAMUS during this year. This is an ongoing work with the PhD student Lénaïc Bagnères (GRAND-LARGE Team at Inria Saclay-Île-de-France, co-advised by Christine Eisenbeis and Cédric Bastoul). The first results have been presented in 2013 at the HiPEAC Computing System Week [9] and at the Rencontres Françaises de Compilation [10].

6.6. Interactive Code Restructuring

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.

[9] http://www.hipeac.net/thematic-session/let-us-push-thread-level-speculation
[10] http://compil13.cri.mines-paristech.fr

This is a rather new research direction which strengthen CAMUS's static parallelization and optimization issue. It has been initiated at Paris-Sud University as a collaboration between Compilation, represented by Cédric Bastoul before he joined CAMUS during this year, and Human-Machine Interaction, represented by Stéphane Huot from the IN-SITU Team at Inria Saclay-Île-de-France. This work is essentially the PhD topic of Alexander Zinenko (IN-SITU Team at Inria Saclay-Île-de-France, co-advised by Stéphane Huot and Cédric Bastoul, CORDI Grant) which started in 2013.

COMPSYS Project-Team

6. New Results

6.1. Parameterized Construction of Program Representations for Sparse Dataflow Analysiss

Participants: André Tavares [UFMG, Belo Horizonte, Brazil], Benoit Boissinot [Ex-Compsys, Google Zurich], Fernando Magno Quintão Pereira [UFMG, Belo Horizonte, Brazil], Fabrice Rastello.

Data-flow analysis usually associates information with control flow regions. Informally, if these regions are too small like a point between two consecutive statements, we call the analysis dense. On the other hand, if these regions include many such points, then we call it sparse. This work presents a systematic method to build program representations that support sparse analyses. To pave the way to this framework, we clarify the literature about well-known intermediate program representations. We show that our approach, subsumes, up to parameter choices, many of these representations, such as the SSA, SSI, and e-SSA forms. In particular, our algorithms are faster, simpler and more frugal than the previous techniques used to construct SSI (static single information) form programs. We produce intermediate representations isomorphic to Choi *et al.*'s sparse evaluation graphs (SEG) for the family of data-flow problems that can be partitioned by variables. However, contrary to SEGs, we can handle - sparsely - problems that are not in this family. We have tested our ideas in the LLVM compiler, comparing different program representations in terms of size and construction time.

This work is part of the collaboration with UFMG (see Section 8.4) and has been accepted for presentation and publication at CC'14 (Compiler Construction Conference) [9].

6.2. A Framework for Enhancing Data Reuse via Associative Reordering

Participants: Kevin Stock [OSU, Columbus, USA], Louis-Noël Pouchet [UCLA, Los Angeles, USA], Fabrice Rastello, J. Ramanujam [LSU, Houston, USA], P. Sadayappan [OSU, Columbus, USA].

The freedom to reorder computations involving associative operators has been widely recognized and exploited in designing parallel algorithms and to a more limited extent in optimizing compilers. However, the use of associative reordering for enhancing data locality has not been previously explored to our knowledge.

In this work, we develop a novel framework for utilizing associativity of operations in regular loop computations to enhance register reuse. Stencils represent a particular class of important computations where our optimization framework can be applied to enhance performance. We use a multi-dimensional retiming formalism to characterize the space of valid transformations and to generate the transformed code. Experimental results demonstrate the effectiveness of the framework.

This work has been submitted to PLDI'14 and is part of the collaboration with P. Sadayappan from the University of Columbus (OSU) (see Section 8.4).

6.3. Function Cloning Revisited

Participants: Matheus Vilela [UFMG, Belo Horizonte, Brazil], Guilherme Balena [UFMG, Belo Horizonte, Brazil], Guilherme Marques [UFMG, Belo Horizonte, Brazil], Fernando Magno Quintão Pereira [UFMG, Belo Horizonte, Brazil], Fabrice Rastello.

Compilers rely on two main techniques to implement optimizations that depend on the calling context of functions: inlining and cloning. Historically, function inlining has seen more widespread use, as it tends to be more effective in practice. Yet, function cloning provides benefits that inline leaves behind. In particular, cloning gives the program developer a way to fight performance bugs, because it generates reusable code. Furthermore, it deals with recursion more naturally. Finally, it might lead to less code expansion, the inlining's nemesis.

In this work, we revisited function cloning under the light of these benefits. We discuss four independent code specialization techniques based on function cloning, which, although simple, find wide applicability, even in highly optimized benchmarks, such as SPEC CPU 2006. We claim that our optimizations are easy to implement and to deploy. We use Wu and Larus's well-known static profiling heuristic to measure the profitability of a clone. This metric gives us a concrete way to point out to program developers potential performance bugs, and gives us a metric to decide if we should keep a clone or not. By implementing our ideas in LLVM, we have been able to speed up some of the SPEC benchmarks by up to 6% on top of the -O2 optimization level.

This work is part of the collaboration with UFMG (see Section 8.4) and was also done in the context of the collaboration with Kalray and the ManycoreLabs project (see Section 7.2).

6.4. Register Allocation and Promotion through Combined Instruction Scheduling, Loop Splitting and Unrolling

Participants: P. Sadayappan [OSU, Columbus, USA], Fabrice Rastello, Lukasz Domanaga.

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, loop fission, 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, loop splitting, 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 a constrained 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 part of the collaboration with OSU (see Section 8.4) and was also done in the context of the collaboration with Kalray and the ManycoreLabs project (see Section 7.2). It contributes to the developments of the Tirex toolbox (see 5.17). It has also been submitted to PLDI'14.

6.5. Beyond Reuse Distance Analysis: Dynamic Analysis for Characterization of Data Locality Potential

Participants: Naznin Fauzia [OSU, Columbus, USA], Venmugil Elango [OSU, Columbus, USA], Mahesh Ravishankar [OSU, Columbus, USA], J. (ram) Ramanujam [LSU, Houston, USA], Fabrice Rastello, Atanas Rountev [OSU, Columbus, USA], Louis-Noël Pouchet [UCLA, Los Angeles, USA], P. Sadayappan [OSU, Columbus, USA].

Emerging computer architectures will feature drastically decreased flops/byte (ratio of peak processing rate to memory bandwidth) as highlighted by recent studies on Exascale architectural trends. Further, flops are getting cheaper while the energy cost of data movement is increasingly dominant. The understanding and characterization of data locality properties of computations is critical in order to guide efforts to enhance data locality.

Reuse distance analysis of memory address traces is a valuable tool to perform data locality characterization of programs. A single reuse distance analysis can be used to estimate the number of cache misses in a fully associative LRU cache of any size, thereby providing estimates on the minimum bandwidth requirements at different levels of the memory hierarchy to avoid being bandwidth bound. However, such an analysis only holds for the particular execution order that produced the trace. It cannot estimate potential improvement in data locality through dependence preserving transformations that change the execution schedule of the operations in the computation.

In this work, we develop a novel dynamic analysis approach to characterize the inherent locality properties of a computation and thereby assess the potential for data locality enhancement via dependence preserving transformations. The execution trace of a code is analyzed to extract a computational directed acyclic graph (CDAG) of the data dependences. The CDAG is then partitioned into convex subsets, and the convex partitioning is used to reorder the operations in the execution trace to enhance data locality. The approach enables us to go beyond reuse distance analysis of a single specific order of execution of the operations of a computation in characterization of its data locality properties. It can serve a valuable role in identifying promising code regions for manual transformation, as well as assessing the effectiveness of compiler transformations for data locality enhancement. We demonstrate the effectiveness of the approach using a number of benchmarks, including case studies where the potential shown by the analysis is exploited to achieve lower data movement costs and better performance.

This work is part of the collaboration with OSU (see Section 8.4) and has been accepted for publication at ACM TACO [2].

6.6. Characterizing the Inherent Data Movement Complexity of Computations via Lower Bounds

Participants: P. Sadayappan [OSU, Columbus, USA], Venmugil Elango [OSU, Columbus, USA], J. (ram) Ramanujam [LSU, Houston, USA], Louis-Noël Pouchet [UCLA, Los Angeles, USA], Fabrice Rastello.

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. Although the problem of characterizing data access complexity has been modeled previously using the formalism of Hong & Kung's red/blue pebble game [27], applicability of previously-developed approaches has been extremely limited. We improve on prior work in several ways: 1) we develop an approach to composing lower bounds from arbitrary decompositions of computational directed acyclic graphs, thereby eliminating a significant limitation of previous approaches that required homogeneity of analyzed computations, 2) we develop a complementary graph min-cut based strategy to Hong & Kung's S-partitioning approach, and 3) we develop an automated approach to generate concrete I/O lower bounds of arbitrary, possibly irregular computational directed acyclic graphs. We provide experimental results demonstrating the utility of the developed approach.

This work has been submitted to PLDI'14 and is part of an informal collaboration with P. Sadayappan from the University of Columbus (OSU) (see Section 8.4).

6.7. Enhancing the Compilation of Synchronous Dataflow Programs

Participants: Paul Feautrier, Abdoulaye Gamatié [LIRMM, Montpellier], Laure Gonnord.

In this work [12], which is an extension of [26], we propose an enhancement of the compilation of synchronous programs with a combined numerical-Boolean abstraction. While our approach applies to synchronous dataflow languages in general, here, we consider the SIGNAL language for illustration. In the new abstraction, every signal in a program is associated with a pair of the form (`clock`, `value`), where `clock` is a Boolean function and `value` is a Boolean or numeric function. Given the performance level reached by recent progress in satisfiability modulo theory (SMT), we use an SMT solver to reason on this abstraction. Through sample examples, we show how our solution is used to determine absence of reaction captured by empty clocks; mutual exclusion captured by two or more clocks whose associated signals never occur at the same time; or hierarchical control of component activations via clock inclusion. We also show that the analysis improves the quality of the code generated automatically by a compiler, e.g., a code with smaller footprint, or a code executed more efficiently thanks to optimizations enabled by the new abstraction. The implementation of the whole approach includes a translator of synchronous programs towards the standard input format of SMT solvers, and an ad hoc SMT solver that integrates advanced functionalities to cope with the issues of interest in this work. These results have been published in 2013 (but considered as published in 2012) in the CSI Journal of Computing [24].

6.8. Synthesis of Ranking Functions using Extremal Counter-Examples

Participants: David Monniaux [Verimag, Grenoble], Lucas Séguinot [Student at ENS Cachan Bretagne], Laure Gonnord.

In [14], we presented a new algorithm adapted from scheduling techniques to synthesize (multi-dimensional) affine functions from general flowcharts programs. But, as for other methods, our algorithm tried to solve linear constraints on each control point and each transition, which can lead to quasi-untractable linear programming instances.

In contrast to these approaches, we proposed a new algorithm based on the following observations:

- Searching for ranking functions for loop headers is sufficient to prove termination.
- Furthermore, there exist loops such that there is a linear lexicographic ranking function that decreases along each path inside the loop, from one loop iteration to the next, but such that there is no lexicographic linear ranking function that decreases at each step along these paths. For these reasons, it is tempting to treat each path inside a loop as a single transition.

Unfortunately the number of paths may be exponential in the size of the program, thus the constraint system may become very large, even though it features fewer variables. To face this theoretical complexity, even though the number of paths may be large, we argue that, in practice, few of them actually matter in the constraint system (we formalize this concept by giving a characterization as geometric extremal points). Our algorithm therefore builds the constraint system lazily, taking paths into account *on demand*.

We are currently testing our preliminary implementation and submitting a paper on these new results.

6.9. Data-Aware Process Networks

Participants: Christophe Alias, Alexandru Plesco.

The following results concern the applied research activities directly linked to the Zettice start-up (see Section 7.3), which aims at applying polyhedral techniques to high-level circuit synthesis (HLS). Following the guidelines of Inria DTI, as this research aims to be transferred, these results are not published before being "protected" or exploited. An Inria patent deposit is currently processed.

- **Data-aware process networks (DPN).** This is the intermediate representation of the HLS flow. DPN is a parallel execution model fitting the hardware constraints of circuit synthesis, in which the data transfer and the synchronizations are made explicit. We formally described the DPN model and a translation scheme from C programs, and we showed the consistency in the meaning where any terminating sequential program is translated to an equivalent DPN, guaranteed to be deadlock free.
- **Front-end analysis.** We designed many program analyses to produce a quality DPN from a C program:
  - *Throughput optimization.* A I/O scheme has been designed, with the corresponding compiler analysis, to minimize the I/O traffic with the external memory. This allows us to balance efficiently the spilling of temporary value to the memory, and the local buffer size. This scheme impacts the DPN structure itself.
  - *Communication vectorization.* The matrix structure of the memory allows us to load data by chunks. A polyhedral analysis has been designed to solve this issue.
  - *Synchronization scheme.* As parallel units need to communicate intermediate results, synchronizations must be ensured.Unlike KPN, DPN do not use FIFO, but buffers, which required an efficient synchronization mechanism.
- **Back-end analysis.** Once generated, a DPN must be mapped to an FPGA. This raises many interesting issues:
  - *Pipeline completion.* Data paths make an extensive use of pipelined operators, which delays the signal. An algorithm has been designed to enforce the time coherence of signals.
  - *Polyhedral units.* DPNs make an extensive use of piece-wise affine functions, which must be mapped properly to ensure the efficiency of the whole system. A preliminary algorithm has been designed to reach a correct trade-off between critical path size and LUT usage.

All these analyses have been fully implemented. The tool Dcc (DPN C Compiler) implements all the front-end analyses. The tool IceGEN implements the back-end analysis.

6.10. Program Equivalence Modulo A/C (Associativity/Commutativity)

Participants: Guillaume Iooss [PhD student], Christophe Alias, Sanjay Rajopadhye [Colorado State University].

Program equivalence is a well-known problem with a wide range of applications, such as algorithm recognition, program verification, and program optimization. This problem is also known to be undecidable if the class of programs is rich enough, in which case semi-algorithms are commonly used. We focus on programs represented as a system of affine recurrence equations (SARE), defined over parametric polyhedral domains, a well-known formalism for the *polyhedral model*, which includes as a proper subset, the class of affine control loop programs. Several semi-algorithms for program equivalence have already been proposed for this class. A few of them take into account algebraic properties such as associativity and commutativity. However, to the best of our knowledge, none of them is able to manage reductions, i.e., accumulations of a parametric number of sub-expressions using an associative and commutative operator.

Our contributions are:

- An equivalence checking algorithm able to manage associativity and commutativity properties. Our method subsumes the previous approaches and is, to the best of our knowledge, the first one able to manage these properties over a parametric number of expressions.
- A semi-algorithm to construct a perfect matching problem on a parametric bipartite graph. We partially solve this problem through a heuristic based on the augmenting path algorithm. This heuristic is able to find a set of non-interfering augmenting paths to improve a proposed maximum matching, as long as these augmenting paths do not have a parametric length.

A preliminary implementation is under development. This work has been submitted to ESOP'14.

6.11. Constant Aspect-Ratio Parametric Tiling

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

Parametric tiling is a well-known transformation that is widely used to improve locality, parallelism, and granularity. However, parametric tiling is also a non-linear transformation and this prevents polyhedral analysis or further polyhedral transformation after parametric tiling. It is therefore generally applied during the code generation phase.

This result consists on a method to stay polyhedral in a special case of parametric tiling, where all the dimensions are tiled and all the tile sizes are constant multiples of a single tile size parameter. We call this *Constant Aspect Ratio Tiling*. We show how to mathematically transform a polyhedron and an affine function into their tiled counterpart and show how to obtain good generated code.

This work has been accepted for publication at IMPACT'14 [8].

6.12. Parametric Tiling with Inter-Tile Data Reuse

Participants: Alain Darte, Alexandre Isoard.

Loop tiling is a loop transformation widely used to improve spatial and temporal data locality, increase computation granularity, and enable blocking algorithms, which are particularly useful when offloading kernels on platforms with small memories. When hardware caches are not available, data transfers must be software-managed: they can be reduced by exploiting data reuse between tiles and, this way, avoid some useless external communications. An important parameter of loop tiling is the sizes of the tiles, which impact the size of the necessary local memory. However, for most analyzes that involve 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. In this work, we showed that, actually, parametric tiling with inter-tile data reuse is nevertheless possible.

Our solution is the first parametric solution for generating the memory transfers needed when a kernel is offloaded to a distant accelerator, tile by tile after loop tiling, and when all intermediate results are stored locally on the accelerator. For such computations, there is a complete decoupling between loads and stores, and when a value has been defined in a previous tile, it has to be loaded from the local memory and not from the distant memory as this memory is not yet up-to-date. In other words, inter-tile reuse is mandatory. This also saves external communications. Our solution is parametric in the sense that we derive the set of loads and stores from and to the distant memory with the tile sizes as parameters. Although the direct formulation is quadratic, we can still solve it in an affine way by developing techniques that consider, in the analysis, all (unaligned) possible tiles obtained by translation and not just those that belong to a tiling (partitioning) of the iteration space. We were able to use a similar technique to also parameterize the computations of local memory sizes, thanks to parametric lifetime analysis and folding with modulos, even for pipeline schedules similar to double buffering. Our method is currently implemented with the `iscc` calculator of `ISL`, a library for the manipulation of integer sets defined with Presburger arithmetic.

Also, the whole analysis can handle approximations thanks to the introduction of the concept of pointwise functions, well suited to deal with unaligned tiles. We believe that this 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.

This work has been accepted for publication at IMPACT'14 [5].

6.13. Data Races in the Parallel Language X10

Participants: Tomofumi Yuki [Colorado State University and Inria/IRISA], Paul Feautrier, Sanjay Rajopadhye [Colorado State University], Vijay Saraswat [IBM Research].

Parallel programmers are now required to efficiently utilize the massive amount of parallelism provided by multi-core and many-core systems. Parallel programming is difficult, and the existing tools are mostly low-level extensions to sequential languages or libraries. As an effort to improve this situation, several groups have initiated the design of parallel programming languages, mostly based on the partitioned global address space (PGAS) paradigm. One of these languages is X10, which is developed at IBM Research by a team led by Vijay Saraswat.

While such languages hide the low-level details of parallel programming, they cannot guarantee that the object code will be correct by construction. Parallelism introduces two new types of bugs: non-determinism and deadlocks, and experience shows that it is possible to guarantee the absence of one type but not both. X10 programs are guaranteed deadlock-free but may have non-determinism. Non-determinism can be detected at runtime, but this approach cannot give absolute guarantees. However, it is possible, at least for a restricted class of X10 programs, to check for non-determinism at compile time.

The first step in this direction is to define the *polyhedral fragment* of X10, in which the only control constructs are `for` loops with affine bounds, and the only data structures are arrays with affine subscripts. X10 has many parallel constructs: as a first effort, we focused on `async`, which creates an activity (lightweight thread) and `finish`, which waits for termination of all impending activities. The execution order (or *happens-before relation*) of such a program is an incomplete lexicographic order, in which terms relating operations in different activities are removed. The dataflow analysis method of [23] has to be adapted to a partial execution order, which may have many extrema instead of a unique maximum. Multiple extrema denote data races, thus non-determinism. A detector along these lines has been implemented and presented at PPoPP'13 (Symposium on Principles and Practice of Parallel Programming) [10].

X10 other parallel programming primitive directives are *clocks* and `atomic`. The `at` construct allows downloading a computation to another *place*. Clocks are a dynamic version of barriers. Their analysis involves counting their instances. For polyhedral programs, this can be done using the Ehrhart and Barvinok theories; the results are polynomials. Checking whether clocks remove non-determinism involves finding integer

roots and hence is undecidable. However, modern SMT solvers are able to solve most of these problems. The resulting paper [13] has been submitted to the ECOOP conference.

6.14. Clock Removal in X10

Participants: Paul Feautrier, Eric Violard [Inria/Camus], Alain Ketterlin [Inria/Camus].

In the light of the previous work on the determinism of X10, a natural question is: are the parallel programming directives of X10 redundant? The answer is yes, at least for static control programs, i.e., programs in which the set of operations and their execution order do not depend on the input data. The basic idea is that the synchronization which occurs when several activities execute an advance is similar to the synchronization at the end of a finish. If one is able to count advances, one may construct a front by gathering all operations with the same advance count. Each front is executed inside one finish, and fronts are executed sequentially in order of increasing counts. For polyhedral programs, advance counting can be done at compile time. If the counts are affine functions, the restructuring can be done by classical polyhedral code generators like CLooG, and no overhead is incurred. For polynomial counts, one overall enclosing loop must be added, but the resulting program can usually be optimized by simple loop transformations, e.g., pushing guards into enclosing loop bounds. For arbitrary programs, the counts have to be computed dynamically; this is possible only if the program has static control.

This result does not contradict the previous undecidability proof (Section 6.13), as the translation of a polyhedral program is usually not polyhedral. Application of the method to a set of simple kernels has shown significant speedups. The interpretation of this result is that, at least in the present state of the X10 runtime, the implementation of the `async` primitive is more mature than the implementation of clocks. A paper on this topic has been accepted at CC'14 (Compiler Construction Conference) [7].

6.15. Static Analysis of OpenStream Programs

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

The objective of the collaboration between the Compsys and Parkas teams in the ManycoreLabs project (Section 7.2) is to evaluate the possibility of applying polyhedral techniques to the parallel language OpenStream, which is developed by Inria Parkas. When applicable, these techniques are invaluable for compile-time debugging and for improving the target code for a better adaptation to the target architecture.

OpenStream is a two-level language, in which a sequential control code directs the initialization of parallel task instances that communicate through *streams*. OpenStream programs are deterministic by construction, but may have deadlocks. If the control code is polyhedral, one may statically compute, for each task instance, its read and write indices for each stream. These indices may be polynomials of arbitrary degree. When linear, the full power of the polyhedral model may be brought to bear for dependence and dataflow analysis, scheduling and deadlock detection, and program transformations.

In the general case, one can think of two approaches: the first one consists in over-approximating dependences until problems become linear. In the second approach, one first leverages modern developments in SMT solvers, which allow them to solve polynomial problems, albeit with no guarantee of success. Furthermore, the task index functions have special properties that may be used to construct original analysis algorithms. Three preliminary results in this direction:

- the proof that deadlock detection is undecidable in general, thanks to an adaptation of the proof designed for X10 (Section 6.13),
- a characterization of deadlocks in terms of dependence graphs, which implies that streams can be safely bounded as soon as a schedule exists with such sizes,
- a preliminary analysis of some solvable cases.

A document is available as Deliverable 2.5.3 for the ManycoreLabs project.

6.16. Array Contraction in Parallel Programs

Participants: Alain Darte, Alexandre Isoard.

Array contraction is a technique to reuse array elements when they are dead, in a form of array folding. A standard technique for array contraction is to use affine remappings with modulos. When the modulo is equal to 1, this corresponds to the removal of the corresponding array dimension. Array contraction is well-known for sequential programs, after element-wise array liveness analysis. It has also been customized for parallel codes obtained through affine schedules by Lefebvre and Feautrier, and Quilleré-Rajopadhye, both frameworks being generalized by the lattice-based memory allocation framework of Darte, Schreiber, and Villard [17] and the construction of the set of conflicting array indices. We showed how the same framework can be used for a larger range of parallel programs, including programs with outer parallel loops, programs exhibiting pipelining, a subset of X10, etc. The optimality of the construction can be shown, despite a related (but actually non-contradictory here) NP-completeness result for worst-case of register pressure in the context of register allocation. A research report on this topic is in preparation.

CONTRAINTES Project-Team

6. New Results

6.1. A Stronger Necessary Condition for the Multistationarity of Chemical Reaction Networks

Participant: Sylvain Soliman.

In the last thirty years, the conjecture of Thomas on the necessary presence of a positive circuit for the occurrence of multistationarity has opened a whole field of research, allowing better modeling and understanding of biochemical networks, especially in the emerging field of systems biology. However, if that aspect is striking in the field of discrete modeling of gene regulatory networks, it did not have the same impact in the Ordinary Differential Equations (ODE) based modeling community. This is mostly due to the fact that this necessary condition, the existence of a positive loop in the Jacobian of the ODE system, is almost always satisfied.

In [5] we improve on the ten years old proof by Soulé, using the structural information from the stoichiometric matrix of a biochemical reaction system. This allows us to state a more strict version of the famous Thomas' necessary condition for multistationarity. In particular, the obvious cases where Thomas' condition was trivially satisfied, mutual inhibition due to a multimolecular reaction and mutual activation due to a reversible reaction, can now easily be ruled out. The new condition makes it possible to use circuit analysis as an useful tool in the arsenal of the computational biologist, together with other structural methods.

6.2. Petri Net Analyses of Biochemical Reaction Networks using Constraint Logic Programming

Participants: François Fages, Thierry Martinez, Faten Nabli, Sylvain Soliman.

The Thesis of Faten Nabli [1] marks our achievements on the static analysis of biochemical reaction networks using Petri Net concepts and Constraint Logic Programming algorithms. This Thesis presents a Boolean model and two constraint-based methods for enumerating all minimal siphons and traps of a Petri net, by iterating the resolution of Boolean satisfiability problems executed with either a SAT solver or a CLP(B) program. The performances of these methods are compared with respect to a state-of-the-art algorithm from the Petri net community. On a benchmark with 80 Petri nets from the Petriweb database and 403 Petri nets from curated biological models of the Biomodels database, we show that miniSAT and CLP(B) solvers are overall both faster by two orders of magnitude with respect to the dedicated algorithm. Furthermore, we analyse why these programs perform so well on even very large biological models and show a polynomial time complexity result for Petri nets of fixed treewidth, using a similar theorem for constraint satisfaction problems with bounded treewidth constraint graphs. Faten Nabli has been hired with a Post-Doc position at Sanofi Paris.

6.3. Structural Model Reduction: CLP and SAT Solvers for Computing Subgraph Epimorphisms

Participants: François Fages, Steven Gay, Thierry Martinez, Francesco Santini, Sylvain Soliman.

This year, in [8], we have developed and compared CLP and SAT solvers on the NP-complete problem of deciding the existence of a subgraph epimorphism between two graphs. Our interest in this variant of graph matching problem stems from the study of model reductions in systems biology, where large systems of biochemical reactions can be naturally represented by bipartite digraphs of species and reactions. In this setting, model reduction can be formalized as the existence of a sequence of vertices, species or reaction, deletion and merge operations which transforms a first reaction graph into a second graph [2]. This problem is in turn equivalent to the existence of a subgraph (corresponding to delete operations) epimorphism (i.e. surjective homomorphism, corresponding to merge operations) from the first graph to the second. We show how subgraph epimorphism problems can be modeled as Boolean constraint satisfaction problems, and we compare CLP and SAT solvers on a large benchmark of reaction graphs from systems biology.

6.4. Quantitative Model Reduction: a CLP Solver for Computing Tropical Equilibrations

Participants: François Fages, Sylvain Soliman.

Model reduction is a central topic in computational systems biology and dynamical systems theory, for reducing the complexity of quantitative models, finding important parameters, and developing multi-scale models for instance. While perturbation theory is a standard mathematical tool to analyze the different time scales of a dynamical system, and decompose the system accordingly, tropical methods provide a simple algebraic framework to perform these analyses systematically in polynomial systems. The crux of these tropicalization methods is in the computation of tropical equilibrations. In [13], we show that constraint-based methods, using reified constraints for expressing the equilibration conditions, make it possible to numerically solve non-linear tropical equilibration problems, out of reach of standard computation methods. We illustrate this approach first with the reduction of simple biochemical mechanisms such as the Michaelis-Menten and Goldbeter-Koshland models, and second, with performance figures obtained on a large scale on the model repository `biomodels.net`.

This work is done in collaboration with Ovidiu Radulescu, Univ. Montpellier, in the context of a larger project about symbolic methods in systems biology with François Boullier, LIFL and Andras Weber, Univ. Bonn.

6.5. Species Minimization in Biochemical Reaction Computing

Participants: Hui-Ju Chiang, François Fages.

Engineering biochemical reactions for computational purposes is a common pursue in synthetic biology. In such design tasks, molecular species have to be carefully engineered to ensure modularity and orthogonality, and are scarce resources. Minimizing the number of involved molecular species is crucial to accomplish a complex computation within a confined biochemical environment.

In [10], we investigate an approach to species minimization by reusing modular and regular reactions in an asynchronous time-multiplexed fashion. Our method enhances not only species utility, but also re-programmability and robustness in realizing various logic circuits. A case study demonstrates the ease of design in realizing general logic computation, and simulation confirms the feasibility and robustness of the proposed method.

This work is done in collaboration with Jie-Hong Jiang and Katherine Chiang from NTU Taiwan in the context of a common project about biochemical programming.

6.6. Hybrid Composition and Simulation of Heterogeneous Biochemical Models

Participants: Hui-Ju Chiang, François Fages, Sylvain Soliman.

[2] Steven Gay, Sylvain Soliman, François Fages. A Graphical Method for Reducing and Relating Models in Systems Biology. Bioinformatics, 26(18):i575–i581, 2010.

Models of biochemical systems presented as a set of formal reaction rules with kinetic expressions can be interpreted with different semantics: as either deterministic Ordinary Differential Equations, stochastic continuous-time Markov Chains, Petri nets or Boolean transition systems. While the formal composition of reaction models can be syntactically defined as the (multiset) union of the reactions, the hybrid composition of models in different formalisms is a largely open issue.

In [7], we show that the combination of reaction rules with conditional events, as the ones already present in SBML, does provide the expressive power of hybrid automata and can be used in a non standard way to give meaning to the hybrid composition of heterogeneous models of biochemical processes. In particular, we show how hybrid differential-stochastic and hybrid differential-Boolean models can be compiled and simulated in this framework, through the specification of a high-level interface for composing heterogeneous models. This is illustrated by a hybrid stochastic-differential model of bacteriophage T7 infection, and by a reconstruction of the hybrid model of the mammalian cell cycle regulation of Singhania et al. as the composition of a Boolean model of cell cycle phase transitions and a differential model of cyclin activation.

6.7. Composition and Abstraction of Logical Influence Networks: Application to Multi-Cellular Systems

Participant: Grégory Batt.

Logical (Boolean or multi-valued) modelling is widely employed to study regulatory or signalling networks. Even though these discrete models constitute a coarse, yet useful, abstraction of reality, the analysis of large networks faces a classical combinatorial problem. In [4], we proposed to take advantage of the intrinsic modularity of inter-cellular networks to set up a compositional procedure that enables a significant reduction of the dynamics, yet preserving the reachability of stable states. To that end, we relied on process algebras, a well-established computational technique for the specification and verification of interacting systems.

We developed a novel compositional approach to support the logical modelling of interconnected cellular networks. First, we formalised the concept of logical regulatory modules and their composition. Then, we made this framework operational by transposing the composition of logical modules into a process algebra framework. Importantly, the combination of incremental composition, abstraction and minimisation using an appropriate equivalence relation (here the safety equivalence) yields huge reductions of the dynamics. We illustrated the potential of this approach with two case-studies: the Segment-Polarity and the Delta-Notch modules.

6.8. Identification of Biological Models from Single Cell Data: a Comparison between Mixed-Effects and Moment-based Inference

Participants: Grégory Batt, Andres Mauricio Gonzalez Vargas, Pascal Hersen, Artémis Llamosi, Jannis Uhlendorf.

Experimental techniques in biology such as microfluidic devices and time-lapse microscopy allow tracking of the gene expression in single cells over time. So far, few attempts have been made to fully exploit these data for modeling the dynamics of biological networks in cell populations. In [9], we compare two modeling approaches capable to describe cell-to-cell variability: Mixed-Effects (ME) models and the Chemical Master Equation (CME). We discuss how network parameters can be identified from experimental data and use real data of the HOG pathway in yeast to assess model quality.

For CME we rely on the identification approach proposed by Zechner et al. (PNAS, 2012), based on moments of the probability distribution involved in the CME. ME and moment-based (MB) inference will be also contrasted in terms of general features and possible uses in biology.

6.9. STL-based Analysis of TRAIL-induced Apoptosis Challenges the Notion of Type I/Type II Cell Line Classification

Participants: Grégory Batt, François Bertaux, Szymon Stoma.

Extrinsic apoptosis is a programmed cell death triggered by external ligands, such as the TNF-related apoptosis inducing ligand (TRAIL). Depending on the cell line, the specific molecular mechanisms leading to cell death may significantly differ. Precise characterization of these differences is crucial for understanding and exploiting extrinsic apoptosis. Cells show distinct behaviors on several aspects of apoptosis, including (i) the relative order of caspases activation, (ii) the necessity of mitochondria outer membrane permeabilization (MOMP) for effector caspase activation, and (iii) the survival of cell lines overexpressing Bcl2. These differences are attributed to the activation of one of two pathways, leading to classification of cell lines into two groups: type I and type II.

In [6] we challenge this type I/type II cell line classification. We encode the three aforementioned distinguishing behaviors in a formal language, called signal temporal logic (STL), and use it to extensively test the validity of a previously-proposed model of TRAIL-induced apoptosis with respect to experimental observations made on different cell lines. After having solved a few inconsistencies using STL-guided parameter search, we show that these three criteria do not define consistent cell line classifications in type I or type II, and suggest mutants that are predicted to exhibit ambivalent behaviors. In particular, this finding sheds light on the role of a feedback loop between caspases, and reconciliates two apparently-conflicting views regarding the importance of either upstream or downstream processes for cell-type determination. More generally, our work suggests that these three distinguishing behaviors should be merely considered as type I/II features rather than cell-type defining criteria. On the methodological side, this work illustrates the biological relevance of STL-diagrams, STL population data, and STL-guided parameter search implemented in the tool Breach. Such tools are well-adapted to the ever-increasing availability of heterogeneous knowledge on complex signal transduction pathways.

6.10. Single Cell Models and Models of Populations: A Mixed Effect Approach

Participants: Grégory Batt, Andres Mauricio Gonzalez Vargas, Pascal Hersen, Artémis Llamosi.

For a long time, experiments and models of gene expression were mainly based on the mean behavior of a population of cells. Although observed early, it is only recently that experimental technique allowed detailed investigation of variability in this process. Since the pioneering work of Elowitz and colleagues, a distinction is drawn between what is called intrinsic and extrinsic variability or noise. Intrinsic noise originates in the randomness of chemical reactions within a cell whether extrinsic noise is the variation in between cells at a given time. Extrinsic variability is associated with population heterogeneity in the concentrations of ribosomes or other molecular players or processes relevant to gene expression (RNAPolII concentration, degradation and dilution rates etc.).

In this work, we propose a modelling framework for gene expression based on a system of ODEs with random parameters following a distribution across the population of cells. In this context, each cell has its own identity which is represented by the value of its parameters. With this model we ask how much of the long term variability can be explained by extrinsic variability alone. We produced long term, time lapse and single-cell data of repeated gene induction in Saccharomyces cerevisiae. One experiment was treated as learning set whereas two were used as test sets. From the learning set, we are able to infer single cell parameters and population distributions which represent accurately in terms of mean and variance the variability in the population. These learned population distributions allowed good predictions on both the learning and test sets.

Our study demonstrates also that the way inference of single cell parameters and distributions is performed is crucial to achieve good performance. Best results being found by joint estimation of the parameters for single cells and for the whole population. With this technique, we noted that very decent fits of the population dynamics can be obtained by estimating only on a very limited number of cells. Concerning the quality of single cell parameters inferred, we validated the presence of an expected significant correlation between the dilution rate and the measured single cell growth rate. This motivates the use of this tool in order to investigate the origins of extrinsic noise, by correlating single cell parameters with measured candidate factors of gene expression variability such as cell density, cell size or age.

6.11. Coupled Model of the Cell Cycle and Circadian Clock

Participants: François Fages, Sylvain Soliman, Denis Thieffry, Pauline Traynard.

Recent advances in cancer chronotherapy techniques support the evidence that there exist important links between the cell cycle and the circadian clock genes. One purpose for modeling these links is to better understand how to efficiently target malignant cells depending on the phase of the day and patient characterictics. This is at the heart of our participation in collaboration with the EPI BANG in the EraNet SysBio project C5Sys, follow up of the former EU STREP project TEMPO.

This year we have pursued the investigation of the effect of transcription inhibition during mitosis, as a reverse coupling from the cell cycle to the circadian clock. We use quantitative temporal logic constraints and the parallel version of BIOCHAM for parameter search, running on the Jade cluster of 10000 processors at the GENCI CINES, to couple dynamical models in high dimension and fit models to experimental data time series obtained in Franck Delaunay's lab in Nice, CNRS. We are defining a series of common temporal logic patterns and *ad hoc* schemes for computing their validity domain on a given trace, more efficiently than by the generic method implemented in BIOCHAM.

6.12. Solving Mixed Shapes Packing Problems by Continuous Optimization with the CMA Evolution Strategy

Participants: François Fages, Thierry Martinez, Lumadaiara Do Nascimento Vitorino.

Bin packing is a classical combinatorial optimization problem which has a wide range of real-world applications in industry, logistics, transport, parallel computing, circuit design and other domains. While usually presented as discrete problems, in [12] we consider continuous packing problems including curve shapes, and model these problems as continuous optimization problems with a multi-objective function combining non-overlapping with minimum bin size constraints. More specifically, we consider the covariance matrix adaptation evolution strategy (CMA-ES) with a nonoverlapping and minimum size objective function in either two or three dimensions. Instead of taking the intersection area as measure of overlap, we propose other measures, monotonic with respect to the intersection area, to better guide the search. In order to compare this approach to previous work on bin packing, we first evaluate CMA-ES on Korf's benchmark of consecutive sizes square packing problems, for which optimal solutions are known, and on a benchmark of circle packing problems. We show that on square packing, CMA-ES computes solutions at typically 14% of the optimal cost, with the time limit given by the best dedicated algorithm for computing optimal solutions, and that on circle packing, the computed solutions are at 2% of the best known solutions. We then consider generalizations of this benchmark to mixed squares and circles, boxes, spheres and cylinders packing problems, and study a real-world problem for loading boxes and cylinders in containers. These hard problems illustrate the interesting trade-off between generality and efficiency in this approach.

6.13. Railway Time Tabling Optimization with CMA-ES and Greedy Heuristics

Participants: François Fages, David Fournier, Thierry Martinez, Sylvain Soliman.

The problem of reducing energy consumption in public transportation has received increasing attention over the last years. Most metros have energy regenerative braking systems, which allow them to produce electric energy when they brake. We study the problem of optimizing the energy consumption of a metro line by modifying the timetable, in order to maximize the actual reuse of the regenerative energy. This is achieved by synchronizing the braking and acceleration phases of the metros, through slight modifications of the stopping times in stations. In an article in preparation, we present a constraint-based model of the electric network of the line, which is used to evaluate the energy consumption at each instant, and to compute a distribution matrix for approximating the potential energy transfers between metros. The optimization of the timetable is then performed by an evolutionary algorithm using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES from Nikolaus Hansen, EPI TAO). On real data, this strategy shows energy savings ranging from 2.38% to 4.54%. Furthermore, these savings are shown to be robust with respect to perturbations of the dwell times.

DREAMPAL Team

6. New Results

6.1. Language-Independent Symbolic Execution, Program Equivalence, and Program Verification

A significant part of our research project consists in applying formal techniques for symbolically executing and formally verifying HiHope programs, as well as for formally proving the equivalence of HiHope programs with the corresponding HoMade assembly and machine-code programs obtained by compilation of HiHope.

- Symbolic execution will detect bugs (e.g., stack undeflow) in HiHope programs. Additionaly, symbolic execution is the natural execution manner of HiHope programs as soon as they contain (typically, underspecified) hardware IPs;
- program verification will guarantee the absence of bugs (with respect to specified properties, e.g., no stack underflow, no invocation of unavailable IPs, ...);
- program equivalence will guarantee that such above-mentioned bugs are also absent from the HoMade assembly and machine-code programs obtained by compilation of HiHope source code.

Since these languages (especially HiHope) are not completely defined yet, we decided to work (together with our colleagues from Univ. Iasi, Romania) on language-independent symbolic execution, program-equivalence, and program-verification techniques. In this way, when all the languages in our project become stable, we will be readily able to instantiate the above generic techniques on (the K formal definitions of) the languages in question. We note that all the techniques described below are also independent of K: they are applicable to other language-definition frameworks that use similar rewriting-based formal operational semantics.

6.1.1. Symbolic Execution

In [9] 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 automatically generated, which has the same syntax as the original one but whose semantics extends data domains with symbolic values and adapts semantical rules to deal with these values. Then, the symbolic execution of concrete programs is, by definition, the execution of programs with the new symbolic semantics, on symbolic input data. We prove that the symbolic execution thus defined has the properties naturally expected from it. A prototype implementation of our approach was developed in the K framework. We demonstrate the genericity of our tool by instantiating it on several languages, and show how it can be used for the symbolic execution and model checking of several programs.

6.1.2. Program Equivalence

In [12] we propose a logic and a deductive system for stating and automatically proving the equivalence of programs in deterministic languages having a rewriting-based operational semantics. 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 program-equivalence problem as we state it. We show that our approach is suitable for proving the equivalence of both terminating and non-terminating programs, and also the equivalence of both 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 in one shot the equivalence of (possibly, infinitely) many concrete programs obtained by replacing the variables by concrete statements or expressions. We also report on a prototype implementation of the proposed deductive system in the K framework.

6.1.3. Program Verification

In [14] we present an automatic and language-independent program verification approach based on symbolic execution. The specification formalism we consider is Reachability Logic, a language-independent logic that constitutes an alternative to Hoare logics. Reachability Logic has a sound and relatively complete deduction system, which offers a lot of freedom (but no guidelines) for constructing proofs. Hence, we propose symbolic execution as a strategy for proof construction. We show that, under reasonable conditions on the semantics of programming languages, our symbolic-execution based Reachability-Logic formula verification is sound. We present a prototype implementation of the resulting language-independent verifier as an extension of a generic symbolic execution engine that we are developing in the K framework. The verifier is illustrated on programs written in languages also formally defined in K.

6.2. Master-Slave Control Structure for MP-SoC Architectures

Our Synchronous Communication Asynchronous Computation (SCAC) model is a data-parallel execution model dedicated to the Massively Parallel System-on-Chip. This model proposes a novel control structure, referred to as master-slave control [11]. Its concept departs from the centralized configuration. However, instead of a uni-processor master controlling a set of parallel processing elements (PE), the master cooperates with a grid of parallel slave controllers which supervises the activities of cluster of PEs.

The control structure in SCAC model is presented by two hierarchical control levels:

- The Master Control Unit (MCU), which controls the order execution in the whole system. It is a simple processor, which fetches and decodes program instruction and broadcasts execution orders to Slave Control Unit. It controls the end execution to establish synchronous communication.
- The Slave Control Unit (SCU), which controls: local node and PEs activities, parallel instructions execution and synchronous communication. It is a crucial component in the master-slave control structure. The SCUs grid allows independent parallel execution.

The hardware architecture is composed of a single MCU and multiple Slave controllers (SCUs) combined with local processing element (PE) (or a cluster of 16 PEs), known collectively as Nodes. The MCU and SCU array are connected through single level hierarchical bus and the SCUs are connected together through X-net interconnection network [2]. This network is clocked synchronously with the SCUs and respectively with the PEs. SCU controllers in the grid care for the instruction execution activities that involve a large degree of parallelism and the communication activities that need to coordinate all the PEs in the grid. The structure of master-slave control should be distinguished from other hierarchical or clustered approaches proposed for parallel computing. Such proposals are usually motivated by memory latency considerations and the desire to build a scalable system. The use of two control levels is therefore visible to the user in its effect on the communication between various processors. With master-slave control structure, the PEs in massively parallel system can execute independently and then can communicate synchronously. Such a construction has the advantage of allowing the designer to optimize distinct processors for their intended tasks and to implement simple interconnection network without additionally buffers and complex routing algorithms.

The aim of these last works is to design a master-slaves control structure for SCAC architecture to allow autonomous processing with simple and regular communication. This control structure based on IP blocks which offers good flexibility and scalability was implemented in synthesizable VHDL code. It is simulated and synthetized for Xilinx Virtex q6 (XC6VLX240T) board. The difficulty of designing a master-slave structure is a compromise between an optimal execution time and high flexibility, while reducing power consumption and silicon area.

6.3. Toward a Massively Parallel and Reflective Execution Model

FPGAs are undoubtedly suited to the definition of what could be called a DSHA (Domain Specific Hardware Architecture). Similarity with the DSSA (Domain Specific Software Architecture) an assembly of functional components performs basic transformations on data, while a software / hardware infrastructure ensures the

ordering of these transformations. The HoMade processor is designed with this in mind: it can be seen as an IP integrator offering a mechanism for interprocess communication IPs via a battery and a scheduler of IPs via dedicated instructions for flow control. In this control we find two particular instructions for flow control designed for a massively parallel execution model for SPMD, and a new instruction can make HoMade reflexive . With this instruction, you can at runtime change the behavior of a virtual component by dynamically associating it to a particular HoMade instruction sentence and in particular IP triggering instructions. Same components can successively after applying this instruction, trigger a hardware IP, a software function which itself can trigger a flow of execution of hardware IPs. This intercession [2] feature , parts of HoMade core, is valid for one processor or for all HoMade slave components in a massively parallel architecture. We demonstrated on a FPGA board which computes the Fibonacci sequence with three different methods, but always through a single call to a unique Virtual Component.

6.4. Power Estimation at System-Level for MPSoC Based Platforms

Shifting the design entry point up to the system level is the most important countermeasure adopted to manage the increasing complexity of Multiprocessor System on Chip (MPSoC). The reason is that decisions taken at this level, early in the design cycle, have the greatest impact on the final design in terms of power and energy efficiency. However, taking decisions at this level is very difficult, since the design space is extremely wide and it has so far been mostly a manual activity. Efficient system-level power estimation tools are therefore necessary to enable proper Design Space Exploration (DSE) based on power/energy and timing. We propose a tool based on efficient hybrid system level power estimation methodology for MPSoC. In this methodology, a combination of Functional Level Power Analysis (FLPA) and system level simulation technique are used to compute the power of the whole system. Basically, the FLPA concept is proposed for processor architecture in order to obtain parameterized arithmetic power models depending on the consumption of the main functional blocks. In this work, FLPA is extended to set up generic power models for the different parts of the platform. In addition, a simulation framework is developed at the transactional level to evaluate accurately the activities used in the related power models. The combination of the above two parts leads to a hybrid power estimation, that gives a better trade-off between accuracy and speed. The proposed methodology has several benefits: It considers the power consumption of the embedded system in its entirety; and Leads to accurate estimates without a costly and complex material. The proposed methodology is also scalable for exploring complex embedded architectures. Based on the proposed methodology, our Power Estimation Tool at System-Level (PETS) is developed. The usefulness and effectiveness of our PETS tool is validated through a typical mono-processor and multiprocessor embedded system designed around the TI OMAP (3530 and 5912) and the Xilinx Virtex II Pro FPGA boards. This methodology is demonstrated and evaluated by using a variety of basic programs to complete media benchmarks. Estimated power values are compared to real board measurements for both simple and multiprocessor architectures. Our obtained power estimation results provide less than 3% of error for mono-processor, 3.8% for homogeneous multiprocessor system and 4.3% for heterogeneous multiprocessor system and 70x faster compared to the state-of-the-art power estimation tools. These results have been presented in the PhD of Santhosh Kumar Rethinagiri [2] and published in [4].

6.5. Dynamically reconfigurable CPU/FPGA architecture for the testing and simulation of avionic systems

Real-time computing systems are increasingly used in aerospace and avionic industries. In the face of power wall and real-time requirements, hardware designers are directed towards reconfigurable computing with the usage of heterogeneous CPU/FPGA systems. However, there is a lack of real-time environments able to deal with the execution of applications on such heterogeneous systems dedicated to avionic Testing and Simulation (T&S). This year, we addressed the problem of soft real-time environments for CPU/FPGA systems and we proposed first a high-performance hardware architecture used to implement intimately coupled hardware and software avionic models. Second, we developed an efficient real-time software environment for the model's

[2] Wikipedia definition: intercession is the ability of a program to modify its own execution state or alter its own interpretation or meaning.

execution, the multi-core CPU monitoring and the runtime task re-allocation to avoid the timing constraint violation. Experimental results underpin the industrial relevance of the presented approach for avionic T&S systems with real-time support. These results are presented in the PhD of George Afonso [1] and in different publications [7] [10] [8].

6.6. A custom reconfiguration controller for partial and dynamic reconfiguration in HoMade based systems

In all Xilinx devices supporting dynamic reconfiguration, such a functionality is realized using a hardware reconfiguration port called ICAP, that moves bitstreams from the reconfiguration memory to the programmable logic. ICAP is initialized by a Xilinx HW controller driven exclusively by a Microblaze processor and thus connected to a PLB or AXI bus.

This makes the partial and dynamic reconfiguration a very tedious task, as it implies using several Xilinx tools (XPS, ISE, PlanAhead,..etc). PDR becomes also resources and time consuming due to the fact that it uses very large interfaces and a static Xilinx architecture (in addition to the system that we want to design) including specific processors, buses, controllers,..etc.

Our contribution is the design of a custom ICAP controller, driven only by a HoMade processor, without any additional processors, buses or controllers. This ensures that our HoMade reconfigurable systems consumes fewer resources on the FPGA and does not require other tools than the standard ISE and PlanAhead tools in order to be designed.

6.7. Hardware control for partially and dynamically reconfigurable systems: from modelling to implementation

This work proposes a control design methodology for FPGA-based reconfigurable systems aiming at increasing control design productivity and guaranteeing implementation efficiency. This methodology is based on a semi-distributed control model [5] composed of a set of modular distributed controllers executing each observation, decision-making and reconfiguration tasks for a reconfigurable region of the system, and a coordinator between the distributed controllers decisions in order to respect global systems constraints and objectives. This semi-distributed decision-making is based on the mode-automata formalism. The proposed combination between modularity, control splitting and formalism-based design allows to enhance the flexibility, reusability and scalability of the control design. Another point that can be added to this combination, to enhance design productivity, is design automation. For this, the proposed methodology is based on Model-Driven Engineering approach [5] allowing to automate code generation from high-level models. This approach makes use of the UML MARTE (Modeling and Analysis of Real-Time and Embedded Systems) standard profile, allowing to make low-level technical details transparent to designers and to automate the VHDL code generation for hardware implementation of the modeled control systems in order to guarantee their performance. The generated control systems were validated using simulation. Synthesis results showed an acceptable time and resource overhead for systems having different numbers of controllers. A control system composed of four controllers and a coordinator was also validated through physical implementation in an FPGA system for an image processing application.

6.8. A model-based approach for dynamically reconfigurable systems design: from MARTE to RecoMARTE

This work is done in the context of the ANR FAMOUS project. It proposes a co-design methodology of dynamically reconfigurable systems based on FPGA. Our methodology is based on the Engineering Model Driven approach (MDE) and the models specification is done in the UML MARTE profile. It aims at ensuring flexibility, reusability and automation to facilitate the work of the designer and improve his productivity. The first contribution related is identifying parts of dynamically reconfigurable FPGA that can be modeled at the high abstraction levels. So, we defined a design flow based on the MDE to ensure the automation of code generation. According to this flow, several models are created mainly through MARTE profile concepts.

However, the modeling concepts of dynamic reconfiguration on FPGAs required extensions in MARTE. Thus, we identified the missing concepts to be integrated in a new profile that extends MARTE called RECOMARTE. The second contribution allows the chain automation and experimental validation. To integrate our design flow and to automate code generation, a processing chain was used. The final model resulting from MARTE proposed design flow is given as input to this chain.

We thereby move from MARTE to RECOMARTE models via an intermediate description according to the IP-XACT standard to finally generate files describing the complete system in the Xilinx XPS environment. This automation will accelerate the design phase and avoid errors due to the direct manipulation of these details. Finally, an example of application of image processing has been developed to demonstrate and validate our methodology.

INDES Project-Team

6. New Results

6.1. Security

Participants: Ilaria Castellani, Bernard Serpette [correspondant], José Santos.

6.1.1. Stateful Declassification Policies for Event-Driven Programs

We propose a novel mechanism for enforcing information flow policies with support for declassification on event-driven programs. Declassification policies consist of two functions. First, a projection function specifies for each confidential event what information in the event can be declassified directly. Second, a stateful release function specifies the aggregate information about all confidential events seen so far that can be declassified. We provide evidence that such declassification policies are useful in the context of JavaScript web applications. An enforcement mechanism for our policies is presented and its soundness and precision are proven. Finally, we give evidence of practicality by implementing and evaluating the mechanism in a browser.
Report and mechanization can be found in http://people.cs.kuleuven.be/~mathy.vanhoef/declass.

6.1.2. A Monitor Inlining Compiler for Securing JavaScript Programs

JavaScript applications can include untrusted code dynamically loaded from third party code providers (such as online advertisements). This issue raises the need for enforcement mechanisms to ensure security properties for JavaScript programs. The dynamic nature of the JavaScript programming language makes it a hard target for static analysis. Hence, research on mechanisms for enforcing security properties for JavaScript programs has mostly focused on dynamic approaches, such as runtime monitoring and program instrumentation. We design and implement a novel compiler that inlines a security monitor and we formally prove it correct with respect to an information flow security property. To the best of our knowledge, it is the first proven correct information flow monitor inlining transformation for JavaScript programs.
Report can be found in http://www-sop.inria.fr/indes/ifJS. See also software section.

6.1.3. Modular Extensions of Security Monitors for Web APIs: The DOM API Case Study

JavaScript programs often interact with the web page on which they are included, as well as with the browser itself, through external APIs such as the DOM API, the XMLHttpRequest API, and the W3C Geolocation API. The continuous emergence and heterogeneity of different external APIs renders the problem of precisely reasoning about JavaScript security particularly challenging. To tackle this problem, we propose a methodology for extending arbitrary sound JavaScript monitors. The methodology allows us to prove noninterference for external APIs in a modular way. Thus, when considering new external APIs, the noninterference property of the security monitor still holds. We present two groups of DOM interfaces that illustrate how to extend a noninterferent monitor model with: (1) basic DOM methods, for which we have discovered new information leaks not explored in previous work; (2) live collections, which are special features of the DOM API with an unconventional semantics that can lead to several previously unknown information leaks. Finally, we inline an extensible noninterferent JavaScript monitor that handles (1) and (2), and we make it available online
Report can be found in http://www-sop.inria.fr/indes/ifJS.

6.1.4. A Certified Lightweight Non-Interference Java Bytecode Verifier

We propose a type system to verify the non-interference property in the Java Virtual Machine. We verify the system in the Coq theorem prover.

Noninterference guarantees the absence of illicit information flow throughout program execution. It can be enforced by appropriate information flow type systems. Much of the previous work on type systems for non-interference has focused on calculi or high-level programming languages, and existing type systems for low-level languages typically omit objects, exceptions and method calls. We define an information flow type system for a sequential JVM-like language that includes all these programming features, and we prove, in the Coq proof assistant, that it guarantees non-interference. An additional benefit of the formalisation is that we have extracted from our proof a certified lightweight bytecode verifier for information flow. Our work provides, to the best of our knowledge, the first sound and certified information flow type system for such an expressive fragment of the JVM.

This work appeared in the journal of Mathematical Structures in Computer Science [9].

6.1.5. Session types for liveness and security

Within the COST Action BETTY, we have started studying the interplay between liveness properties and secure information flow properties in session calculi, in collaboration with a colleague from Torino University. Recent developments in static analysis techniques have shown that behavioural types, and in particular session types, may be used to enforce liveness properties of communicating systems. Examples of such properties are deadlock freedom, eventual message delivery and session termination. Because secure information flow in communicating systems depends on the observation of messages, there is a clear connection between information flow analysis and the liveness properties of the systems under consideration. We have been examining the joint application of liveness enforcement and secure information flow analysis in session calculi. It appears that, by strengthening the assumptions on the liveness of systems, it is possible to relax the conditions under which a system satisfies secure information flow properties. This is ongoing work, which is expected to continue within the BETTY Action.

6.1.6. Noninterference in reactive synchronous languages

We defined two properties of Reactive Noninterference (RNI) for a core synchronous reactive language called CRL formalising secure information flow. Both properties are time-insensitive and termination-insensitive. Again, coarse-grained RNI is more abstract than fine-grained RNI.

Finally, a type system guaranteeing both security properties was presented. Thanks to a design choice of CRL, which offers two separate constructs for loops and iteration, and to refined typing rules, this type system allows for a precise treatment of termination leaks, which are an issue in parallel languages.

This work has been presented at the International Symposium on Trustworthy Global Computing (TGC 2013) [11]. It is also described in Attar's PhD thesis pejman:tel-00920152.

6.2. Models, semantics, and languages

Participants: Pejman Attar, Gérard Berry, Gérard Boudol, Ilaria Castellani, Johan Grande, Cyprien Nicolas, Tamara Rezk, Manuel Serrano [correspondant].

6.2.1. Formalization and Concretization of Ordered Networks

Overlay networks have been extensively studied as a solution to the dynamic nature, scale and heterogeneity of large computing platforms, and are a fundamental layers of most existing peer-to-peer networks. The basic mechanism offered by an overlay network, is routing, i.e., the mechanism enabling the delivery of messages from any node to any other node in the network. On top of routing are built crucial functionalities of peer-to-peer networks, such as networks maintenance (nodes joining and leaving the network) and information distribution and retrieval. Over the years, different topologies and routing mechanisms have been proposed in literature. However, there is a lack of formal works unifying these different designs and establishing their correctness. This paper proposes a formal common basis, partially validated with the Coq theorem prover, with the nice property of only requiring the definition of a total order on the nodes. We investigate how such a basic design can be used to build deadlock/livelock-free algorithms for routing, node insertion, and node deletion in the fault-free environment. The genericity of our design is then explored through the construction of orders on

nodes corre- sponding to different topologies commonly encountered in the peer-to-peer domain. To validate the methodology proposed, a simulator tool was developed. This tool is able, given the definition of an order and the definition of shortcuts, to simulate the corresponding overlay network and to explore its performance.

6.2.2. Absence Prediction in Esterel

We have formally proved, with the Coq system, the correctness of an absence prediction of Esterel's signals. For this we have formalised in Coq the static analysis and the interpreter written in Scheme (see the previous activity report). With this formal specification, we prove the correctness of the analysis: if a signal is considered absent by the evaluator at one instant, then this signal will be not emitted during this instant. This work is described in a currently submitted paper.

6.2.3. Reactive Synchronous Languages

CRL: We have studied the security property of noninterference in a synchronous Core Reactive Language (CRL). In the synchronous reactive paradigm, programs communicate by means of broadcast events, and their parallel execution is regulated by a notion of instant.

We have first shown that CRL programs are indeed reactive, namely that they always converge to a state of termination or suspension ("end of instant") in a finite number of steps. This property is important as it also entails the reactivity of a program to its environment, namely its capacity to input events from the environment at the start of instants, and to output events to the environment at the end of instants. While classical in synchronous languages, this property required to be established afresh in CRL, since this language makes use of a new asymmetric parallel operator.

We defined two bisimulation equivalences on CRL programs, corresponding respectively to a fine-grained and to a coarse-grained observation of programs. We showed that coarse-grained bisimilarity is more abstract than fine-grained bisimilarity, as it is insensitive to the order of generation of events and to repeated emissions of the same event during an instant.

DSLM :

We have finalised our work on the language DSLM (Dynamic Synchronous Language with Memory), which is an extension of CRL with memory and distribution. There are now several sites, and agents may migrate between sites. Two main properties are established for DSLM: reactivity of each agent and absence of data-races between agents. Since DSLM uses the same asymmetric parallel operator as CRL, reactivity is proven in a similar way. Moreover, the language offers a way to benefit from multi-core and multi-processor architectures, by means of the notion of synchronized scheduler, which abstractly models a computing resource. Each site may be expanded and contracted dynamically by varying its number of synchronized schedulers. Moreover agents can be moved transparently from one scheduler to another one within the same site. In this way one can formally model the load-balancing of agents over a site. This work is part of Pejman Attar's PhD thesis, defended in December 2013.

6.2.4. Locking Fast

We have studied the integration of low-level locking mechanisms in programming language execution environments. We have shown that for a given low-level locking mechanism the performance of the applications may vary significantly according to decisions taken for integrating it in the runtime system. We have studied two different aspects. First, we have shown how to accelerate C IO locking by selecting at runtime the adequate implementation and by using spin locks instead of full-fledged mutexes. Second, we have presented a new schema for improving the slow path of Java-like synchronized blocks. It consists in lifting the exception handler that is installed on the stack and which is in charge of releasing a monitor up to the closest exception handler already installed on the stack. All these optimizations have been implemented in Hop, our Web programming language. We have conducted experiments that shows significant speed up (up to 30%) for applications using locks extensively.

The synchronization lifting technique could be generalized to all the exception handlers, not only the handlers of synchronized blocks. As lifting only modifies the interception of exceptions, not the way they are thrown, it is compatible with languages such as Java or JavaScript that store a description of the stack at the moment when the exception is thrown inside the exception handlers. The technique should thus be broadly applicable. Exploring this idea is left for future work.

This work is described in the paper that will be published in the proceedings of the SAC'14 conference [12].

6.2.5. JThread

The jthread library is a library for Hop offering threads and mutexes and whose main locking function implements deadlock avoidance. Our library offers structured locking (i.e., critical sections instead of explicit `lock`/`unlock` functions). It supports nested locking. Our library is implemented using the preexisting pthread library and is offered as an alternative to the latter.

Compared to usual locking functions, our primitive relies on the programmer to provide some supplementary information such as the set of mutexes that might be acquired while owning a first one. However, for this supplementary information we chose default values that limit the need for the programmer to actually write it to a minimum.

The syntax of our locking construct is as follows:

```
(synchronize* l [:prelock p]
    expr1
    expr2
    ... )
```

where `l` is the list of mutexes to lock and `p` is a list that contains (some of [1]) the mutexes that might be locked during the execution of the body of the construct.

The implementation of this function relies on the ability to lock n mutexes at once. We found an algorithm for this that is both deadlock-free and starvation-free. Our algorithm relies on a dynamic total ordering of threads; this is inspired by Lamport's bakery algorithm.

We wrote a starvation-freedom property that applies to our real-life language with dynamic thread creation and programs that run forever on purpose. To express the property we need to define the following relation over threads:

$t_1 prec t_2$ iff. $existsm.t_1$ owns m and t_2 is waiting to lock m.

Let $prec^*$ be the symmetric transitive closure of $prec$.

The property that we chose and proved for our algorithm is:

If each non-waiting thread eventually releases all the mutexes it owns **and if** for each waiting thread t the number of threads t' s.t. $t' prec^* t$ does not tend toward $+infty$ over time **then** each waiting thread eventually gets the mutexes it is waiting to lock.

We have implemented our library and integrated it to Hop. We haven't released it yet. An article is in preparation.

6.3. Web programming

Participants: Gérard Berry, Yoann Couillec, Ludovic Courtès, Cyprien Nicolas, Vincent Prunet, Tamara Rezk, Marcela Rivera, Bernard Serpette, Manuel Serrano [correspondant].

[1] According to a few rules that we impose

6.3.1. Colored λ-calculus

We have extended the bicolored λ-calculus to a polychromic one. With two colors, we were able to abstract the Hop language with its '$' and '~' annotations. With more than two colors, we can also modele embedded languages as a query based language, for example. As for the bicolored version, we have defined a static transformation aggregating expressions of the same color. We have formally proved, with the Coq system, the correctness, the confluence and the terminaison of the transformation. This work has been accepted for publication at the conference JFLA'14. [14].

6.3.2. Multitier Debugging

The distributed nature of Web applications makes debugging difficult. The programming languages and tools commonly used make it even more complex. Generally the server-side and the client-side are implemented in different settings and the debugging is treated as two separated tasks: on the one hand, the debugging of the server, on the other hand, the debugging of the client. Most studies and tools focus on this last aspect. They concentrate on the debugging of JavaScript in the browser. Although useful, this only addresses one half of the problem. Considering the debugging of Web applications as a whole raises the following difficulties:

- As the server-side and the client-side are generally implemented in different languages, debuggers for the Web do not capture the whole execution of the application. Programming the server and the client in the same language helps but is not sufficient to let the debugger expose a coherent view of the whole execution as this also demands a runtime environment that enforces consistent representations of data structures and execution traces.
- The JavaScript tolerant semantics tends to defer errors raising. For instance, calling a function with an insufficient number of arguments may lead to filling a data structure with the unexpected `undefined` value which, in turn, may raise a type error when accessed. The *distance* between the error and its actual cause may be arbitrarily long which can make the relation between the two difficult to establish.
- The JavaScript event loop used for the GUI splits the execution into unrelated callback procedures which get called upon event receipts. When an error occurs, the active stack trace only contains elements relative to the current callback invocation. It is oblivious of the context of the callback. Understanding the cause of the error is then not easy.

Pursuing our research on multitier programming for the Web, we have built a programming environment which eliminates most of these problems.

- When an error is raised, the full stack trace is reported. This stack trace might contain server stack frames, client stack frames, or both. We call this a *multitier stack trace*.
- When an error occurs, either on the client or on the server, its source location is reported by the debugger.
- In *debugging mode*, types, arities, and array bounds, are strictly enforced on the server and on the client. Hence, when the execution of the program deviates from the formal semantics of the language, an error is raised immediately.

A paper currently submitted presents this debugger and exposes the salient aspects of its implementation is under submission.

6.3.3. Hop and HipHop : Multitier Web Orchestration

Our aim is to help programming rich applications driven by computers, smartphones or tablets; since they interact with various external services and devices, such applications require orchestration techniques that merge classical computing, client-server concurrency, web-based interfaces, and event-based programming. To achieve this, we extend the Hop multitier web programming platform [5] by the new HipHop domain specific language (DSL), which is based on the synchronous language Esterel. HipHop orchestrates and synchronizes internal and external activities according to timers, events generated by the network, GUIs, sensors and devices, or internally computed conditions.

Like Esterel, Hiphop is a concurrent language based on the perfect synchrony hypothesis: a HipHop program repeatedly reacts in conceptual zero-delay to input events by generating output events; synchronization and communication between parallel statements is also performed in conceptual zero-delay. Perfect synchrony makes concurrent programs deterministic and deadlock-free, the only non-determinism left being that of the application environment. Its implementation is cycle-based, execution consisting of repeated atomic cycles "read inputs / compute reaction / generate outputs" in coroutine with the main Hop code. Concurrency is compiled away by static or dynamic sequential scheduling of code fragments. Cyclic execution atomicity avoids interference between computation and input-output, which is the usual source of unexpected non-determinism and synchronization problems for classical event-handler based programming.

While Esterel is limited to static applications, HipHop is designed for dynamicity. Its implementation on top of Hop makes it possible to dynamically build and run orchestration programs at any time using Hop's reflexivity facilities. It even makes it possible to modify a HipHop program between two execution cycles. It also simplifies the language by importing Hop's data definition facilities, expressions, modular structure, and higher-order programming features. It relies on the Web asynchronous concurrency and messaging already supported by Hop.

Using HipHop for real-life applications such as multimedia applications has been presented in an invited paper of the conference ICDCIT'14 [10].

This year, we extended the HipHop language with dynamic constructions, namely `genpar&` and `dyngenpar&`. These new constructs allow HipHop applications to parallelize treatment of event's values without knowing *a priori* the number of values carried by a given event. genpar& is alike a delayed parallel map execution, while dyngenpar& may create new parallel branches on-demand.

HipHop was also extended with listeners, alike HTML/DOM ones. The programmer can attach functions to any element of the HipHop programe that will be triggered when an instruction is started, suspended, resumed, terminated or aborted. These listeners enable us to trace a specific part of a program, easing its debugging.

6.3.4. Hop Programming Environment

In Linux-based environments, Hop is launched using the command line or using OS init scripts. This is inadequate for the Mac OS environment where graphical user interfaces are generally used to start, stop, and control applications. To fit the Mac OS users habits we have implemented a graphical front-end to Hop. It allows users to monitor and manage Hop processes. The implemented high level graphical interface is a XCode project which has been developed in objective-C for Mac OS X 10.7/10.8.

The main functionalities developed in this graphical front-end are:

- easily manage Hop processes. This GUI allows users to start, to stop and to restart the execution of Hop processes in a simplified manner. This process is executed in an independent thread in order to prevent impacts in the main program. Even though the process becomes independent, the main program can still control the execution of the Hop process by stopping or restarting it.
- Display messages in a user-frienly way. All messages as well as standard and error outputs, generated after the launch Hop server, are captured and redirected to be displayed in the main program. This allows the user to monitor the execution of the Hop process at any time. In this way, the user can search for a specific output or display pattern. All messages can be saved in external files for further analysis.
- Launch a Hop process with specific settings. It is possible to specify the port number on which the Hop server will accept connection. The verbosity and debugging level can also be specified according to the needs. At the same time the user can activate/deactivate specific options like Zeroconf and Webdav.
- Specify additional arguments to run a Hop process in a "command line like" way. In particular situations, advanced users could need to specify some options when launching the Hop server.

Additionally, a set of scripts have been developed to facilitate the generation and distribution of this work. A group of scripts allows one to compile and build the Hop GUI without needing a graphic Xcode interface. In this way, it is possible to generate an application bundle in a local machine as well as in a remote one without needing additional graphical interfaces.

The other group of scripts allows one to generate a "ready to use" dmg image containing Hop files. This dmg image can either include the graphical user interface (GUI) or not. Due to the continuous evolution of Hop based on new requirements and bugs fixed, the latter set of scripts provides a powerful tool to improve the releasing of new product versions.

This front-end has been integrated in the main Hop development tree. The MacOS pre-compiled version is publicly available on the Hop web site http://hop.inria.fr.

6.3.5. *Web of Data*

We are extending the Hop programming language in order to improve its data management: the amount of data it can access, the increasing number of sources of data and the heterogeneity of data it can accept. We have made an implementation of the SPARQL query language and the ORC orchestration language in Hop. We have written a configurable interpreter of the ORC language. The parallelism of the interpreter can be activated or not, for each operator of the ORC language. This specificity allows different executions of an ORC application, depending on the execution context or constraints, such as an execution on a client which disallows any parallelism. Within the X-Data project, we have participated in the development of a data intensive application in collaboration with Data Publica, the leading company of the project, and with the Inria Zenith research team. This application analyzed data sets provided by the French Insee institute to exhibit population commuting patterns. Our incentive for participating in this development was to acquire knowledge on programming data intensive applications. In the mid-term, we will rest on this expertise to create new data-aware programming languages or programming language extensions.

6.4. Web robotics

Participants: Ludovic Courtès, Cyprien Nicolas, Vincent Prunet [correspondant], Manuel Serrano.

6.4.1. *Cable driven robots*

The sound design of modern robotic applications demands for the configuration-time integration of various subsystems which together constitute a robot. APIs and protocols such as ROS (Robot Operating System) provide robot designers with tools to combine software and hardware components into a complete robot. In addition, more and more robots need to share information or interact with diffuse objects available in the robot neighborhood, and also with remote services, to log data (typically activity monitoring data in the case of an assistance robot), to send information messages (alarms or triggering events to some other infrastructure), to subscribe to services provided by objects or remote servers, to provide services that may help peer entities, to get new behaviors by downloading and installing applications within the robot. We develop tools and architectures to address these requirements using Hop as our main platform. We experiment software architectures involving robots, web objects, several integration models with third party components (hardware, software computation libraries for robotics, stand-alone robots), protocols, and libraries.

We pursued the joint work with Coprin Team about using Hop to coordinate a cable-driven robot. We changed the hardware on which Hop runs to a mini-PC instead of a standard laptop, plugged a wireless router, and used the wireless network from a tablet to move the robot. We also improved the robot hardware and software. The setup has been summarized in a paper [13] and presented at a national conference on robotics.

6.4.2. *Web Robotics*

Web Robotics is a two years Inria ADT project targeting the development of technical foundations (libraries and toolkits) and demos of web enabled robots. The project is led by Indes (Vincent Prunet, Manuel Serrano), software development is supported by Inria SED (Ludovic Courtès), robots are provided by Inria Coprin. A demonstrator and dissemination platform consisting of a cable robot and dedicated web services have been set up to enable people to interact with the robot through a web server (web http://webrobotics.inria.fr:8080/hop/welcome).

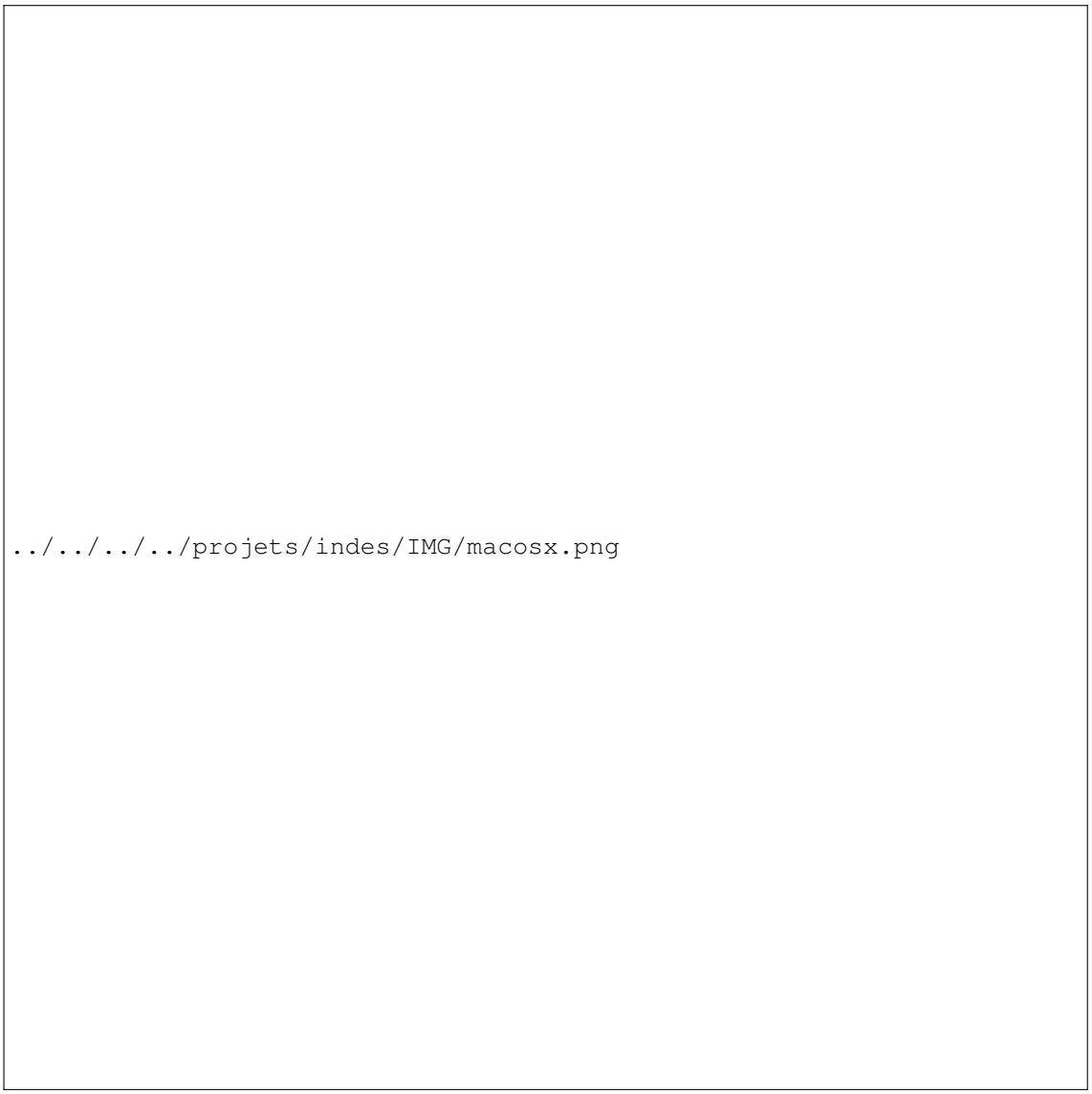

Figure 1. A screenshot of the MacOS X Hop graphical interface.

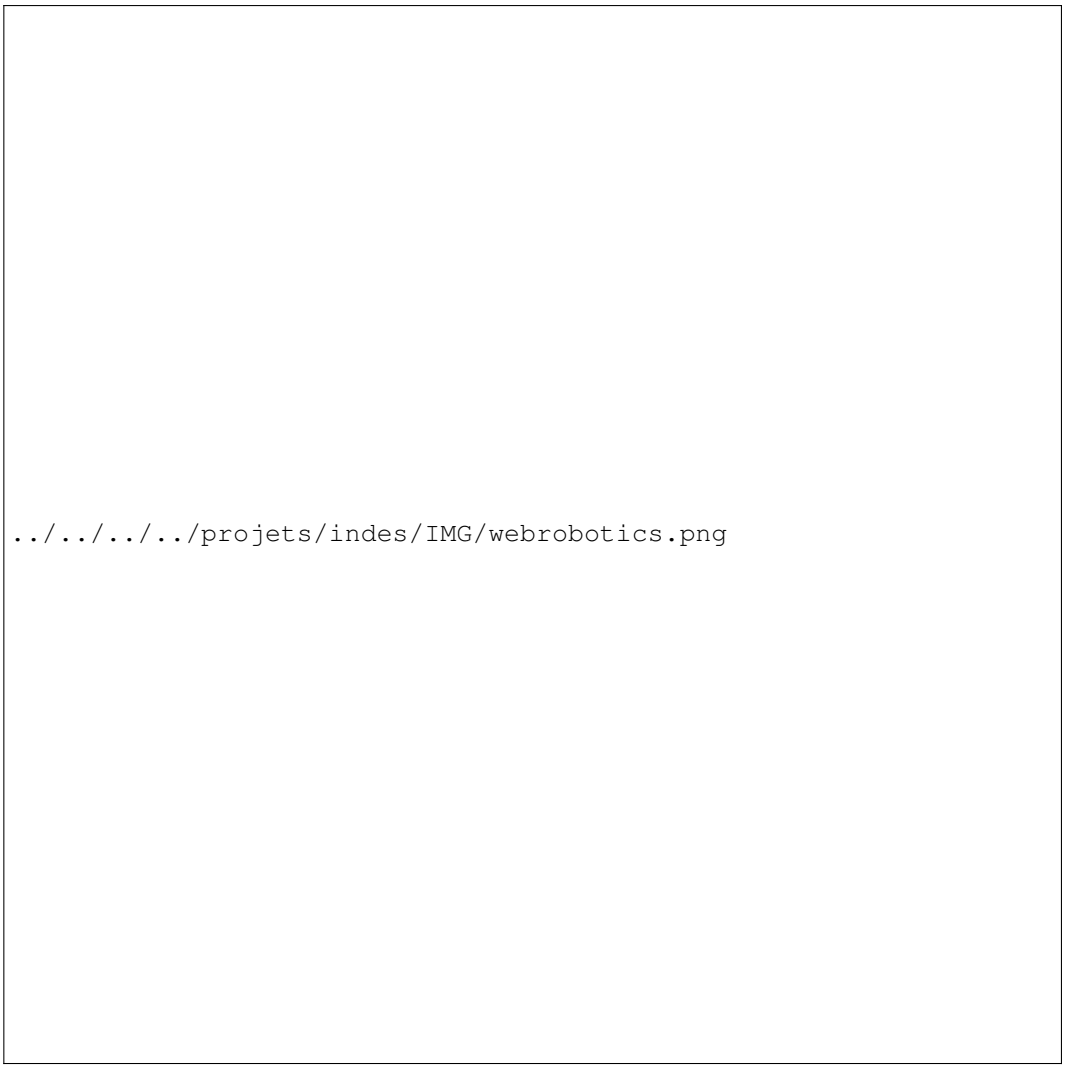

Figure 2. A web-controlled cable robot. The whole application is implemented with Hop.

The web robotics demo demonstrates:

- the programming of robot control functions within Hop
- simulation/real hardware abstraction
- hardware control (Phidget integration)
- integration of specialized robotics libraries
- multi server architecture
- management of multiple users and authenticated access to critical resources.

Hop added value is to provide:

- an unconstrained specification environment where data and services are easily shared among servers and clients;
- a seamless, plugin free, integration into standard web browsers.

Also in 2013, Indes has joined the PAL (Person Assisted Living) Inria project, to develop web enabled applications within the project.

PAREO Project-Team

6. New Results

6.1. Static analysis

Participant: Sergueï Lenglet.

6.1.1. Static analysis for control operators

Control operators allow programs to have access and manipulate their execution context. Abortive control operators, such as *call/cc* in Scheme or SML, capture the entire execution context (also called continuation), while delimited-control operators, such as *shift* and *reset* captures only a part of the continuation (delimited by reset). We want to prove properties (like equivalences between terms or termination) for languages with these operators, using static analysis.

In [9], [16], we study the behavioral theory of a language with delimited control. More precisely, we define environmental bisimilarities for the delimited-control operators shift and reset. We consider two different notions of contextual equivalence: one that does not require the presence of a top-level control delimiter when executing tested terms, and another one, fully compatible with the original CPS semantics of shift and reset, that does. For each of them, we develop sound and complete environmental bisimilarities, and we discuss up-to techniques.

In [8], we present new proofs of termination of evaluation in reduction semantics (i.e., a small-step operational semantics with explicit representation of evaluation contexts) for System F with control operators. We introduce a modified version of Girard's proof method based on reducibility candidates, where the reducibility predicates are defined on values and on evaluation contexts as prescribed by the reduction semantics format. We address both abortive control operators (*callcc*) and delimited-control operators (*shift* and *reset*) for which we introduce novel polymorphic type systems, and we consider both the call-by-value and call-by-name evaluation strategies.

6.1.2. Polymorphism and higher-order functions for XML

In [11], we define a calculus with higher-order polymorphic functions, recursive types with arrow and product type constructors and set-theoretic type connectives (union, intersection, and negation). We study the explicitly-typed version of the calculus in which type instantiation is driven by explicit instantiation annotations. In particular, we define an explicitly-typed λ-calculus with intersection types and an efficient evaluation model for it. In a companion paper [21], we define a local type inference system that allows the programmer to omit explicit instantiation annotations, and a type reconstruction system that allows the programmer to omit explicit type annotations. The work presented in the two articles provides the theoretical foundations and technical machinery needed to design and implement higher-order polymorphic functional languages for semi-structured data.

6.2. Model Transformations

Participants: Jean-Christophe Bach, Pierre-Etienne Moreau.

Model Driven Engineering is a technique that has been applied quite successfully for the design of complex systems. Such systems cannot be released and embedded without complying with the certification required by the application domain: EN 50128 for railways, DO-178C for aeronautics, or ISO 26262 for automotive for instance.

Recently we have developed an extension of *Tom* to support the development of Model Transformations and the generation of traces which are needed to give confidence in the quality of the implemented transformation.

In [12], we present a method, a language and dedicated tooling to ease and to speed up software development based on models transformations. Our approach aims to bridge the gap between general purpose languages and domain specific ones in order to take benefit from both of the two worlds, and to increase software quality. Our approach uses the Tom language which is a shallow extension of general purpose languages. Our proposal allows to write modular transformations whose code is reusable, and which are traceable.

6.3. Property based testing

Participants: Horatiu Cirstea, Pierre-Etienne Moreau, Cosay Topaktas.

Quality is crucial for software systems and several aspects should be taken into account. Formal verification techniques like model checking and automated theorem proving can be used to guarantee the correctness of finite or infinite systems. While these approaches provide a high level of confidence they are sometimes difficult and expensive to apply. Software testing is another approach and although it cannot guarantee correctness it can be very efficient in finding errors.

We have proposed a property based testing framework for the *Tom* language inspired from the ones prosed in the context of functional programming. In the current version relatively simple properties can be already expressed and tested on *Tom* programs. It consists of an exhaustive approach testing all possible input values and guaranteeing that the discovered counter-examples are the smallest ones (the size of the inputs is clearly limited by the execution time) and a random approach where inputs of bigger size could be tested but the minimal counter-example is not guaranteed. A relatively simple shrinking method which searches a smaller counter-example starting from an initial relatively complex one has been also proposed. There is ongoing work on the expressiveness of the property language and the efficiency of the shrinking method. The library is available at http://gforge.inria.fr/projects/tom.

6.4. Nominal Theory

Participant: Christophe Calvès.

Nominal unification is proven to be quadratic in time and space. It was so by two different approaches, both inspired by the Paterson-Wegman linear unification algorithm, but dramatically different in the way nominal and first-order constraints are dealt with.

To handle nominal constraints, Levy and Villaret introduced the notion of replacing while Calvès and Fernández use permutations and sets of atoms. To deal with structural constraints, the former use multi-equation in a way similar to the Martelli-Montanari algorithm while the later mimic Paterson-Wegman.

In [10] we abstract over these two approaches and genralize them into the notion of modality, highlighting the general ideas behind nominal unification. We show that replacings and environments are in fact isomorphic. This isomorphism is of prime importance to prove intricate properties on both sides and a step further to the real complexity of nominal unification.

TASC Project-Team

6. New Results

6.1. Solvers

Participants: Nicolas Beldiceanu, Rémi Douence, Narendra Jussien, Xavier Lorca, Eric Monfroy, Charles Prud'Homme.

- [14] presents some research directions wrt sustainable solver development based on the idea that solvers should be based/derived on data bases of combinatorial knowledge.
- [19] and [42] presents a solver independent language dealing both with variable-oriented and constraint-oriented propagation engines to enable the design of propagation engines.
- By observing the resolution process, [35] shows how to dynamically adapt the resolution while propagating constraints.

6.2. Filtering

Participants: Nicolas Beldiceanu, Alban Derrien, Jérémie Du Boisberranger, Jean-Guillaume Fages, Arnaud Letort, Xavier Lorca, Thierry Petit, Charlotte Truchet, Mohamed Wahbi.

- Given a matrix model, with the same constraint defined by a finite-state automaton on each row and a global cardinality constraint on each column, [12] exploits double counting to derive necessary conditions on the cardinality variables of the global cardinality constraints from the automata. (participants: Beldiceanu)
- By using the observation that most global constraints can be reformulated as a conjunction of a total function constraint together with a constraint that can be easily reified (e.g. a linear constraint involving two variables), [13] introduces a simple way for deriving reified global constraints. (participants: Beldiceanu)
- In the context of distributed constraint solving [22], [25] introduce two filtering algorithms that extend Asynchronous Forward Checking (AFC). The last one outperforms AFC specially on sparse problems. (participants: Wahbi)
- We improve the energetic reasoning checker of the cumulative constraint by decreasing the number checked intervals by a factor seven. We prove this approach can be generalized to the ER filtering algorithm. Furthermore, in a context of makespan minimization of hard problems, our experiments demonstrate that associating this checker with a Time-Table propagator is more efficient than using the best state-of-the-art propagators, such as Time-Table Edge-Finding. This work is at the core of Alban Derrien's doctoral research (Alban is a PhD student of TASC). It was published at the doctoral program of CP2013 [28]. (participants: Derrien, Petit)
- [29] introduces a probabilistic model for the bound consistency algorithm of the alldifferent constraint in order to decrease the number of times the constraint is woken without making new deductions during constraint propagation. (participants: Du Boisberranger, Lorca, Truchet).
- Initially motivated by the shift minimisation personal task scheduling problem [30] shows how to integrate difference constraints into the AtMostNValue constraint in order to get a better estimation about the minimum number of distinct values. (participants: Fages, Lorca)
- Motivated by scalability issues, and based on the idea of accelerating the convergence to the fix-point by filtering several cumulative constraints in parallel, [33] and [47] presents a sweep based algorithm for a conjunction of cumulative constraints. (participants: Beldiceanu, Letort)
- [46] come up with a more efficient filtering algorithm than the one introduced for the cost regular constraint for dealing with constraints for which the set of solutions can be represented by an automaton with counters. (participants: Beldiceanu)

6.3. Continuous/discrete

Participants: Nicolas Beldiceanu, Gilles Chabert, Jean-Guillaume Fages, Charles Prud'Homme.

- While convexity (and some of its generalisations) is a key property used for dealing with continuous constraints it was not yet used in the context of discrete global constraints. In [34] we come up with a parametric filtering algorithm based on a form of convexity. It can handle in a uniform way various constraints such as deviation, spread or the conjunction of a linear inequality constraint and count constraint.
- Motivated by hybrid discrete continuous problems we come up in [44] with a simple and efficient interface for connecting a discrete constraint solver (Choco) and a continuous constraint solver (Ibex).

6.4. Learning Constraint Models

Participants: Nicolas Beldiceanu, Naina Razakarison.

- In the context of learning parametrized constraint models for highly structured problems we address in [38] the problem of finding the coefficients of polynomials in several variables from example parameter and function values.
- In the context of semi structured time series where the structural aspect is related to technological constraints we deal in [24] with the problem of extracting functional dependency constraints. The problem is motivated by extracting constraints from electricity production time series and is characterized by a larger set of samples (from 7 years and from 300 plants).

6.5. Meta Heuristics

Participants: Alejandro Reyes Amaro, Eric Monfroy, Florian Richoux, Charlotte Truchet.

- The aim is to develop and implement new algorithmical methods for constraint problems on massively parallel machines. We also conduct more theoretical studies about the parallelization of constraint problems. This year, we proposed a fairly sharp model to predict parallel speed-ups one can expect while parallelizing by a multi-walk parallel scheme any Las Vegas algorithm by just studying the distribution of sequential run-times [41]. This model shows a divergence of only 20% when predicting speed-ups over 256 cores, on very different benchmarks.
- To evaluate the scalability and parallelization of local search algorithms for SAT, [23] presents a statistical method based on the analysis of the runtime behavior of its sequential version.
- [26] and [26] deals with the use of metaheuristics for solving the resource constrained scheduling problem and the set covering problems.

6.6. Search and modelling

Participants: Eric Monfroy, Thierry Petit.

- In the context of autonomous search, [21] deals with the problem of automatically tuning a search strategy (i.e., variable value selection). For this purpose it uses so called *choice functions* which provide an evaluation of a strategy in term of a set of indicators. [36] and [31] go one step further by providing tuning and adaptation facilities at the level of the different components of a constraint solver.
- Using the MiniZinc modeling language, [32] shows how to model and solve the portfolio selection problem with constraint programming. Since more than ten year constraints for which the set of solutions can be matched to the language accepted by an automaton were introduced in many solvers (e.g., Choco, Gecode, SICStus). [40] describes an interface for describing such constraints in a more convenient way.
- Many discrete optimization problems have constraints on the objective function. Being able to represent such constraints is fundamental to deal with many real world industrial problems. In this work, we go one step further in the concept of topologically concentrate high values in a sequence of cost variables. We refine the work we previously published in CP2012 thanks to three generalizations of the focus constraint. We experiment successfully the technique in scheduling, round-robin and musical benchmarks. This work has been published at IJCAI 2013 [37].

6.7. Miscellaneous

Participants: Eric Monfroy, Florian Richoux.

- [15] gives a complete characterization of the complexity of the existential positive first-order logic, that one can interpret as model checking on monotone csp. We exhibit a dichotomy criterion remaining the same on finite domains of every cardinality, as well as countable and uncountable infinite domains.
- We develop an artificial intelligence, AIUR, to play the real time strategy game $StarCraft^{tm}$, using both machine learning and constraint-based techniques. AIUR finished 3$^{\text{rd}}$ to $StarCraft^{tm}$ AI competitions organized at the conferences AIIDE 2013 and CIG 2013. [18] presents a survey on AI techniques applied on $StarCraft^{tm}$.

ESPRESSO Project-Team

6. New Results

6.1. A pivot in between synchrony and asynchrony

Participants: Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

Our time modeling framework requires a pivot specification paradigm to materialise a spectrum of models of computation and communication ranging from synchrony to asynchrony, from software to hardware, and accommodate with (abstractions of) software behaviors (software, functional blocks, tasks) and requirements (temporal properties, contracts, regular expressions) through logical, periodic, multi-periodic or affine time. We aim at developing a framework comprising dataflow networks (communications) and synchronous automata (computations) controlled by synthesised wrapper enforcing abstractions of specified constraints from the software viewpoint (timing requirements).

Relations between Kahn networks and classes of synchronous dataflow graphs (SDF) as well as synchronous languages have been studied in the past (e.g. Lustre), yet never in the full generality of relating the domain-theoretic model of Kahn networks to it most general synchronous incarnation (one that at least allows to express several clock domains) [17]. We are currently elaborating such a model to characterise morphisms between untimed asynchronous networks and multi-clocked, synchronous, dataflow networks. In this prospect, we developed the first constructive operational semantics of Signal [21], which opens to further investigations on its full abstraction relation with a denotational characterisation using Kahn networks over a polychronous domain.

6.2. New functionalities of Polychrony

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic.

We have developed and integrated in the Signal toolbox some clock computations useful for optimizations: *assignment clocks* and *utility clocks*. These information may be used to reduce the frequency of the computations and the communications for distributed code generation.

Assignment clock. A given signal is supposed to be computed at the instants of its clock, defined by the clock of the expression of its definition. For a signal x, the expression of its definition can always be rewritten as $x := (E_1 \texttt{ when } h_1) \texttt{ default } ... \texttt{ default } (E_{n-1} \texttt{ when } h_{n-1}) \texttt{ default } (x \$ \texttt{ when } k)$. If we assume that the signal keeps, between two consecutive instants, the last computed value, the assignment of $(x \$)$ to x is unnecessary. Then, the assignment clock is then defined by $h_1 \hat{+} ... \hat{+} h_{n-1}$, smaller than the clock of x defined by $(h_1 \hat{+} ... \hat{+} h_{n-1}) \hat{+} k$.

Utility clock. The utility clock defines the instants at which a signal is necessary. For a signal x, the utility clock, $hu(x)$ is defined by:

- the clock of x if x is an input, an output, a memory, or if it is used to define an undersampling clock ($\texttt{when } f(x)$);
- otherwise, it is defined, for $x \rightarrow y_1 \texttt{ when } h_1$, ..., $x \rightarrow y_n \texttt{ when } h_n$, by $\sum_{i=1,n} (hu(y_i) \hat{*} h_i)$.

If we rewrite the Signal program by sampling the signals (except for inputs/outputs) by their utility clock, the new Signal program is equivalent to the previous one, with respect to its behavior with the external world. Note that the utility clock can be used only when this transformation does not introduce cycles in the graph.

6.3. Formal Verification of Synchronous Dataflow Program Transformations Toward Certified Compilers

Participants: Van-Chan Ngo, Jean-Pierre Talpin, Thierry Gautier, Paul Le Guernic, Loïc Besnard.

Translation validation [49], [48] is a technique that attempts to verify that program transformations preserve the program semantics. A compiler generally involves several phases during its compilation process. For instance, the Signal compiler [2], [8], in its first two phases, *calculates the clock information*, makes *Boolean abstraction*, and makes *static scheduling*. The final phase is the executable code generation. One can try to prove globally that the input program and its final transformed program have the same semantics. However, we believe that a better approach consists in separating the concerns and proving for each phase the preservation of different kinds of semantic properties. In the case of the Signal compiler, the preservation of the semantics can be decomposed into the preservation of clock semantics, data dependence, and value-equivalence of variables.

Translation Validation for Clock and SDGs Transformations. This work focuses on proving the preservation of clock semantics in the first two phases of the Signal compiler. In order to do that we encode the clock semantics and data dependence as *clock models* and *synchronous dependence graphs* (SDGs). Then we show that a transformation is correct if and only if there exist *refinements* between clock models, and between SDGs, written as $\Phi(P_2) \sqsubseteq_{clk} \Phi(P_1)$ and $SDG(P_2) \sqsubseteq_{dep} SDG(P_1)$ [15]. We delegate the checking of the preservation to a SMT-solver [38], [54].

Translation Validation of Polychronous Dataflow Specifications: from Signal to C using Synchronous Dataflow Value-Graphs. In this work, we build a validator for the synchronous dataflow compiler of Signal. This validator tries to match the value-graph [53] of each output of the original program and its transformed counterpart. That ensures that every output of the original program and its counterpart in the transformed program have the same value whenever they are present. Our validator does not require any instrumentation and modification of the compiler, nor any rewriting of the source program.

The Signal program and its generated C program have been represented in the same shared synchronous dataflow value-graph (SDVG), in which the nodes for the same structures (variables, constants, operators) have been shared. For instance, the values of input signals and their corresponding variables in the generated C code are represented by the same nodes in the shared graph. Then, the shared graph is transformed following *predefined rules* to show that all output signal values in the Signal program and their counterparts in the generated C code are rooted at the same subgraph.

Consider the following process, where $IR(P)$ is the compiled code of the program P and TV(SDVG(P,IR(P))) is *true* when all output signal values in P and their counterparts in $IR(P)$ are the same:
if (Cp(P) is Error) then output Error; else
if ((Φ(IR(P)) $\sqsubseteq_{clk}$ Φ(P)) and (SDG(IR(P)) $\sqsubseteq_{dep}$ SDG(P)) and (TV(SDVG(P,IR(P))))) then output IR(P); else output Error.
This will provide formal guarantee as strong as that provided by a formally certified compiler w.r.t. the clock semantics and the data dependence in case the validator is certified formally.

Implementation and Experiments. At a high level, our tool *SigCert* [47] developed in OCaml checks the correctness of the compilation of the Polychrony Signal compiler w.r.t clock semantics, data dependence, and value-equivalence as shown in Figure 8 .

6.4. Exploring system architectures in AADL via Polychrony and SynDEx

Participants: Huafeng Yu, Loïc Besnard, Thierry Gautier, Jean-Pierre Talpin, Paul Le Guernic.

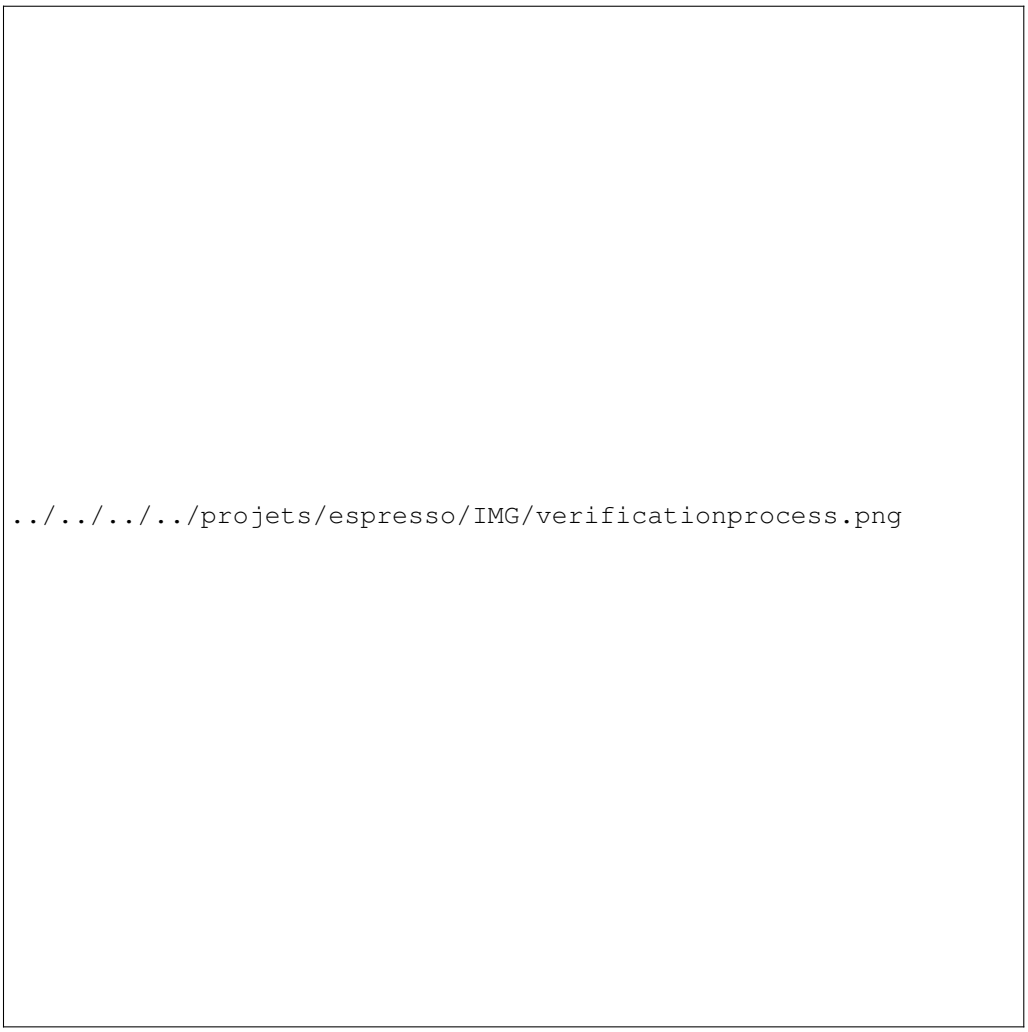

Figure 8. An overview of our integration within Polychrony toolset.

Architecture analysis & design language (AADL) has been increasingly adopted in the design of embedded systems, and corresponding scheduling and formal verification have been well studied. However, little work takes code distribution and architecture exploration into account, particularly considering clock constraints, for distributed multi-processor systems. Our approach [20], [16], [17] handles these concerns within the toolchain AADL-Polychrony-SynDEx. First, in order to avoid semantic ambiguities of AADL, the polychronous/multiclock semantics of AADL, based on a polychronous model of computation, is considered. Clock synthesis is then carried out in Polychrony, which bridges the gap between the polychronous semantics and the synchronous semantics of SynDEx [42]. The same timing semantics is always preserved in order to ensure the correctness of the transformations between different formalisms. Code distribution and corresponding scheduling is carried out on the obtained SynDEx model in the last step, which enables the exploration of architectures originally specified in AADL. Our contribution provides a fast yet efficient architecture exploration approach for the design of distributed real-time and embedded systems. The approach has been illustrated using an avionic case study.

6.5. A synchronous annex for the AADL

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

We propose a synchronous timing annex for the SAE standard AADL. Our approach consists of building a synchronous model of computation and communication that best fits the semantics and expressive capability of the AADL and its behavioral annex and yet requires little to know (syntactic) extension to it, i.e. to identify a synchronous core of the AADL (which prerequisites a formal definition of synchrony at hand) and define a formal design methodology to use the AADL in a way that supports formal analysis, verification and synthesis.

Our approach first identifies the core AADL concepts from which time events can be described. Then, is considers the behavior annex (BA) as the mean to model synchronous signals and traces through automata. Finally, we consider elements of the constraint annex to reason about abstractions of these signals and traces by clocks and relations among them. To support the formal presentation of these elements, we define a model of automata that comprises a transition system to express explicit transitions and constraints, in the form of a boolean formula on time, to implicitly constraint its behavior. The implementation of such an automaton amounts to composing its explicit transition system with that of the controller synthesised from its specified constraints.

6.6. Ongoing activities and results for integration of Polychrony with the P toolset

Participants: Christophe Junke, Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

Current state of P. The P language is still under definition, notably for the software/hardware architectural description of systems. In late october 2013, technical partners (headed by Adacore) released the first beta version of the toolset. The main activites of the ESPRESSO team can be splitted in analysis and development activities:

- The analysis activites consisted in understanding what tasks shall be performed ultimately by the P toolset w.r.t. code generation and architecture, and how Polychrony could be used in the proposed workflow.
- The development activities consisted in introducing a modified block sequencing algorithm in P and starting the development of the P to Signal converter.

Co-modeling in P. First, P should import functional behavior from Simulink, Stateflow and UML class, activity and state machine diagrams. Those represent a strictly sequential semantics: “the code generated from functional behaviour language will be strictly sequential and void of tasking features” (P specification [1]).

[1] https://forge.open-do.org/plugins/moinmoin/p/

Second, imported architectural description languages are likely to be SysML, MARTE and AADL, which present concurrent semantics. Hence, "the code generated from architectural description languages may include concurrent semantics (thread, shared resources...)" (*ibid*). However, code generation from architectural description languages will consist of invocations to an underlying real-time API. The current target of code generation is the APEX ARINC-653 API, which provides real-time services like inter/intra-partition communication channels as well as task scheduling. Real-time properties of imported architectural elements, like task periods and scheduling policy, are used to configure those services.

Code distribution. Code generation should be able to distribute the functional blocks among architectural elements (processors/threads and buses/queues).

Polychrony offers a way to distribute Signal processes among different locations [31]. In general, such code distribution may lead to the synthesis of new input and output ports: when expressing synchronous communication with asynchronous protocols, some clock information might need to be added to resynchronize data-flows. Moreover, the computation model of Signal allows to order asynchronous read and write operations to avoid communication deadlock. The extended input/output interfaces of blocks could be reimported back to P in order to ensure the correctness of code distribution.

It appears however that the subset of Simulink that is imported in P, and the execution model of P functional models that is enforced by the P compiler, can be viewed as a composition of endochronous multi-rate nodes (all inputs of a node are computed before all of its outputs; this avoids deadlock problems when composing nodes). This model ends up being similar to a Lustre model of computation, where code distribution can be performed without adding communication flows and where read/write operations can be setup in a general way without introducing deadlocks [41].

Despite the above observations, it might be possible to extend the input/output interfaces of existing P models thanks to Polychrony. One approach is to ensure that block dependencies between Simulink blocks are effectively respected after code distribution. Indeed, functional blocks can be partially ordered thanks to user-defined priorities. If other partners see an interest with this approach, it could be possible to establish communication links between ordered blocks, so that the global execution order of blocks in a distributed setting is the same as the one modeled originally in the simulation environment.

Model clustering. Alternatively, it would be interesting from a Signal point of view to loosen the synchronization assumptions made by both Simulink and P so that only algebraic dependencies are taken into account (e.g. interpret all Simulink subsystems as virtual, ignore all non-strictly required dependencies...), while respecting clock constraints (e.g. sample time, controlled and enabled blocks...). In that case, the Signal compiler could perform code distribution for simulation purposes, or simply to provide another compilation scheme for P. Another step could be to apply an automatic code distribution mechanism into so-called *clusters*, and export those clusters back to P as architectural elements. The resulting P model would end-up being having possibly more tasks/threads and smaller functional blocks, which might be interesting. Those design decisions are still under consideration and must be discussed with other partners.

From P to Signal. The development activities in the P project currently consist in adding a P to Signal translator. It is being developed as a backend of the P toolset, which provides a number of facilities to access and perform computations on P models. The current prototype must be completed and refined according to what are the actual needs in the project, but can already be tested with input models.

In order to validate the approach, the existing test models of the P projects are all checked with this exporter (over two hundreds small models, a couple of big ones). The resulting SSME files are then converted to Signal files: this step required to generate a command-line version of the Eclispe Polychrony product, as well a a batch converter from SSME to Signal (this converter is integrated in the Polychrony environment). In addition to convert SSME files to Signal, this converter also validates the model against Ecore constraints. Finally, the resulting Signal files are compiled with the original `C++` Signal compiler to check typing and clock relationships (those tests are not performed at the SSME level). The resulting test toolchain gives useful feedbacks for the iterative development of the translator.

Partial orders in P. The exporter also needs to export block dependencies from functional models. Since Polychrony is also able to infer a total order while taking into account code distribution, it was not satisfactory to export the existing total order computed by the P toolset: it is more sensible to export the subset that is strictly necessary (or desired). In agreement with technical partners, we modified the existing sequencer so that it could be parameterized with block ordering criteria (for example, we might want to take into account dataflow dependencies as well as user-defined priority in Polychrony, but nothing more). The outcome is a single package responsible for computing partial and total orders inside the P toolset. This prevents other tools, like the P to Signal exporter, to compute partial order by themselves.

The implementation of the sequencer is based on a dependency matrix that helps computing the transitive closure of dependencies (to quickly check whether two blocks are dependent on each other) while keeping track of their transitive reduction (in order to export only the minimal set of relationships). Now that the first version of the P toolset is released, the sequencer will hopefully be integrated in the P toolset.

6.7. Real-Time Scheduling of Dataflow Graphs

Participants: Adnan Bouakaz, Jean-Pierre Talpin.

The ever-increasing functional and nonfunctional requirements in real-time safety-critical embedded systems call for new design flows that solve the specification, validation, and synthesis problems. Ensuring key properties, such as functional determinism and temporal predictability, has been the main objective of many embedded system design models. Dataflow models of computation (such as KPN [44], SDF [46], CSDF [34], etc.) are widely used to model stream-based embedded systems due to their inherent functional determinism. Since the introduction of the (C)SDF model, a considerable effort has been made to solve the static-periodic scheduling problem [28]. Ensuring boundedness and liveness is the essence of the proposed algorithms in addition to optimizing some nonfunctional performance metrics (e.g. buffer minimization, throughput maximization, etc.). However, nowadays real-time embedded systems are so complex that real-time operating systems are used to manage hardware resources and host real-time tasks. Most of real-time operating systems rely on priority-driven scheduling algorithms [51], [37] (e.g. RM, EDF, etc.) instead of static schedules which are inflexible and difficult to maintain. Our work [12], [18], [19] [35] addresses the real-time scheduling problem of dataflow graph specifications; i.e., transformation of the dataflow specification to a set of independent real-time tasks w.r.t. a given priority-driven scheduling policy such that the following properties are satisfied: (1) channels are bounded and overflow/underflow-free; (2) the task set is schedulable on a given uniprocessor (or multiprocessor) architecture. This problem requires the synthesis of scheduling parameters (e.g. periods, priorities, processor allocation, etc.) and channel capacities. Furthermore, our work considers two performance optimization problems: buffer minimization and throughput maximization.

6.8. Structure-Preserved Distribution of Synchronous Programs

Participants: Ke Sun, Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

We propose an automatically structure-preserved distribution method, which is based on synchronous guarded actions [50] and component calls in an intermediate representation [36]. The guarded actions describe the local behavior. The component calls preserve the modular structure information of synchronous programs. Using this method, the designer can naturally blend the distribution design into the whole system design procedure: following the modular structure, a globally asynchronous locally synchronous (GALS) network over distributed nodes can be automatically constructed. Each node corresponds to a component and contains:

- a computing element, computing (as sender) or reacting to (as receiver) scheduling commands;
- a controlling element, called adaptor, adjusting the asynchronous communication between nodes.

The computing element focuses on the functional behaviors (i.e., value computation) in synchronous runs, which can be perfectly described by synchronous guarded actions. On the other hand, the controlling element mainly considers the temporal constraints (i.e., clock relation) under asynchronous communications. Guarded actions are not suitable for specifying clock relations, then we use polychronous specifications [8] to define the inter-node communications.

A perspective for future work would be the structure-preserved distribution of synchronous programs with multi-interaction. Multiple interactions in one logical instant are desynchronized and projected onto finer grained instants. Owing to this extension, it would provide more convenience for the designer to express the parallelism among components.

S4 Project-Team

6. New Results

6.1. New result 1

TRIO Team

6. New Results

6.1. Probabilistic real-time systems

Participants: Liliana Cucu-Grosjean, Adriana Gogonel, Codé Lo, Dorin Maxim and Cristian Maxim.
The arrival of complex hardware responding to the increasing demand for computing power in next generation systems exacerbates some of the limitations of static timing analysis for the estimation of the worst-case execution time (WCET) estimation. In particular, the effort of acquiring (1) detail information on the hardware to develop an accurate model of its execution latency as well as (2) knowledge of the timing behaviour of the program in the presence of varying hardware conditions, such as those dependent on the history of previously executed instructions. These problems are also known as the timing analysis walls. The probabilistic timing analysis, a novel approach to the analysis of the timing behaviour of next-generation real-time embedded systems, provides answers to timing analysis walls. In [7], [13], [11] timing analysis attacks the timing analysis walls. We have also presented experimental evidence that shows how probabilistic timing analysis reduces the extent of knowledge about the execution platform required to produce probabilistically-safe and tight WCET estimations.

Based on existing estimations of WCET or minimal inter-arrival time [16], we may propose different probabilistic schedulability analyses [6], [12].

2013 was also the year when through several invited talks [8], [10], [9], we had the opportunity to underline historical misunderstandings on probabilistic real-time systems. The most common is related to the notion of independence that is used with a wrong meaning by different papers.

AOSTE Project-Team

6. New Results

6.1. Process Networks with routing for parallel architectures

Participants: Robert de Simone, Emilien Kofman, Jean-Vivien Millo.

In the past we developed a dedicated Process Network (PN) formalism with explicit static switching/routing schemes for data flow. This year we considered the practical use of our formalism to model data-streams in specific applicative contexts.
In a first direction we considered the case of stencil algorithms, usually modeled with cellular automata (CA) (as in heat or gas propagation models for instance). In that case, the application itself is modeled in a way strongly similar to a physical architecture consisting of a regular mesh/array of parallel processors (MPPA). Mapping can seem to be straighforward then, *safe that* the neighborhood and connection topology may differ from the CA model to the MPPA. Our results consider efficient routing and propagation schemes on a given MPPA interconnect fabric, so as to match all-to-all broadcast paterns up to a given distance (on the CA topology). They are described in [20], and were implemented on Kalray MPPA256 prototype architecture.
A similar modeling effort was conducted, this time on FFT algorithm models (again described as parallel pipe-lined tasks). Again switching/routing schemes were provided in our formal PN model to map the virtual logical dependences onto concrete connection patterns in a MPPA256 model. This was the subject of Emilien Kofman internship, of which preliminary results were presented in a junior workshop [36].

6.2. Formal analysis of MARTE Time Model and CCSL

Participants: Frédéric Mallet, Robert de Simone, Yuliia Romenska, Jean-Vivien Millo, Ling Yin.

We have worked on building analysis methods and tools for running exhaustive analyses on MARTE/CCSL specifications. This was done by endowing CCSL with a State-Based semantics [51]. Each operator is described as a boolean state machine, some operators require an infinite number of states. When this is the case we rely on a lazy representation technique to capture symbolically the infinite number of states [45]. The semantics of a CCSL specification is then expressed as the synchronized product of the (infinite) state machines for each operator. Even though the operators are infinite, their composition can sometimes be bounded. When the synchronized product has only a finite number of reachable states, it is said to be safe. We have identified a set of representative and frequently used examples where this is the case [38]. When the product is not finite, our (semi-)algorithm to build the product does not terminate, therefore it is important to be able to know in advance whether or not the product is safe. We have thus proposed an algorithm to decide whether a CCSL specification is safe [37]. It relies on an intermediate representation called Clock Causality Graph and uses results from marked graph theory.

Building the product for a CCSL specification is exponential in the number of clocks and is not practical for large specifications. So, to avoid building explicitly the product we have proposed another technique to explore symbolically the state-space of a CCSL specification [49]. This relies on a liveness condition where no conflict may prevent an infinite clock from ticking infinitely often. Branches that may lead to states where an infinite clock dies are pruned by a fix-point algorithm.

These two solutions focus on the logical and discrete aspects of MARTE/CCSL, which was devised to unify logical and physical time constraints. An attempt to support verification of the physical time constraints of MARTE/CCSL was conducted through the use of UppAal timed automata and model-checker [46]. The proposed technique combines the logical clocks of CCSL with the real-valued clocks of timed automata. Synchronous/Polychronous aspects are solved with TimeSquare 5.1 while the UppAal model-checker is used to explore the space derived from the real-valued clocks.

6.3. Logical time in Model-Driven Engineering of embedded systems

Participants: Frédéric Mallet, Julien Deantoni, Robert de Simone, Marie-Agnès Peraldi Frati, Matias Vara-Larsen, Arda Goknil.

In the context of our approach based on logical time to specify causalities and synchronizations on models, 3.2 , we developed an extension of the OMG OCL Object Constraint Language. Named ECL (Event Constraint Language) it provides such specifications of causalitity and synchronization at syntactic language level, which enabled then automatic generation of semantic logical time constraints for any model that conforms the language.
This year, we extended to a new challenge, using logical time constraints to coordinate models of *several distinct* languages used jointly for a large heterogeneous system description. This work is reported in [25], [52].
It was illustrated in practice in the automotive domain by coordinating together the Timed Augmented Description Language (TADL2) and the EAST-ADL language [34], [32] (the formalisms are rather similar, but still with clear distinctions at places) .
Finally, we proposed a pattern to assemble the (possibly concurrent) semantics of a language associating our logical time constraints (based on pure clocks) with a syntactic action language (providing behavior content). By reifying events and constraints, this specification of the semantics is amenable to its composition [25]. Such approach has been, again, recently used for a first attempt to coordinate distinct behavioral models [47].

As part of our collaboration in the DAESD associated-team with ECNU Shone-SEI in Shanghai we studied the coupling of discrete-logical with continuous-physical time models, ending with a proposal of Hybrid MARTE statecharts [19] specified in a style much like a combinaison of MARTE state diagrams and timed automata.

In another setting we presented a new model of scenarios [21], dedicated to the specification and verification of system behaviours in the context of software product lines (SPL). The formalism uses the logical time modeling aproach, with a strong link to synchronous semantics. We draw our inspiration from some techniques that are mostly used in the hardware community, and we show how they could be applied to the verification of software components and product line variability. We point out the benefits of synchronous languages and models to bridge the gap between both worlds.

6.4. Multiview modeling and power intent in Systems-on-chip

Participants: Carlos Gomez Cardenas, Ameni Khecharem, Emilien Kofman, Frédéric Mallet, Julien Deantoni, Robert de Simone.

Power models for embedded architectures (where power consumption is highly constrained) provide an ideal example of a non-functional modeling framework with strong interactions with the functional and performance models: more speed in computation comes at the cost of larger energy consumption. There was also a demand for a framework allowing combinaison of models, each representing a distinct view of the system. We demonstrated as part of the HeLP ANR project 8.2.1.1 , followed by the newly started HOPE ANR project 8.2.1.2 , how such multiview modeling could be done, and how it could be connected down to more concrete simulation code or model, as in SystemC, Docea Power AcePlorer, or Scilab code. The multiview modeling applied to power intent and power managers was described in [35], and led to the PhD defense of Carlos Gomez Cardenas in December 2013 [16].

6.5. Performance variability analysis on manycore architectures

Participants: Sid Touati, Amin Oueslati, Franco Pestarini, Robert de Simone, Emilien Kofman.

In the context of the collaboration with Kalray (see 7.1.1), we conducted a systematic benchmarking campaign to test the stability (or low variability) of the performances of the MPPA256 prototype manycore processor. We first addressed issues of memory access and network latency, then programmed a distributed verson of the classical ALL_PAIRS_SHORTEST_PATH parallel algorithm with an hybrid OpenMP/MPI style. This was the objectif of Amin Oueslati Master2 internship. Results were encouraging, and showed stability of performance over a large set of runs.

This work is currently extended during the International Internship grant of Franco Pescarini. Specific on-chip communication modes offered by the MPPA256 processor (namely *portal* and *channel* communication modes) are being extensively benchmarked. Results show time predictability on the case of light on-chip communication traffic, but stability gets degraded as performance decreases in presence of heavy traffic and congestion (various runs show quite different execution time).
In another effort we conducted during the internship period of Emilien Kofman an experiment on MPPA256 quite similar to the work conducted as part of the collaboration with Kontron (see 7.1.3), exploring various mapping options of FFT algorithm variants, with the goal of figuring how to best map (in the future) several such algorithms onto the computation fabric of the many-cores available.

6.6. Off-line (static) mapping of real-time applications onto NoC-based many-cores

Participants: Thomas Carle, Manel Djemal, Dumitru Potop Butucaru, Robert de Simone, Zhen Zhang.

Modern computer architectures are increasingly relying on multi-processor systems-on-chips (MPSoCs, also called chip-multiprocessors), with data transfers between cores and RAM banks managed by on-chip networks (NoCs). This reflects in part a convergence between embedded, general-purpose PC, and high-performance computing (HPC) architecture designs. In past years we have identified and compared the hardware mechanisms supporting precise timing analysis and efficient resource allocation in existing NoCs. We determined that the NoC should ideally provide the means of enforcing a global communications schedule that is computed off-line and which is synchronized with the scheduling of computations on CPU cores (and we have built such a NoC).

This year we have focused on the problem of mapping applications onto NoC-based MPSoCs (discussed in this section) and on the associated problem of timing analysis of the resulting parallel implementations (discussed in section 6.7). On-chip networks used in MPSoCs pose significant challenges to both on-line and off-line real-time scheduling approaches. They have large numbers of potential contention points, have limited internal buffering capabilities, and network control operates at the scale of small data packets. Therefore, precise schedulability analysis requires scalable algorithms working on hardware models with a level of detail that is unprecedented in real-time scheduling.

We considered an off-line scheduling approach, and we targeted massively parallel processor arrays (MPPAs), which are MPSoCs with large numbers (hundreds) of processing cores. We proposed a novel allocation and scheduling method capable of synthesizing such global computation and communication schedules covering all the execution, communication, and memory resources in an MPPA. To allow an efficient use of the hardware resources, our method takes into account the specificities of MPPA hardware and implements advanced scheduling techniques such as pre-computed preemption of data transmissions and pipelined scheduling.

Our method has been implemented within the Lopht tool presented in section 5.4 , and first results are presented in [54]. One of the objectives of the collaboration with Kalray SA is the evaluation of the possibility of porting Lopht onto the Kalray MPPA platform.

6.7. WCET estimation for parallel code

Participant: Dumitru Potop Butucaru.

This is joint work with Isabelle Puaut, Inria, EPI ALF.

Classical timing analysis techniques for parallel code isolate micro-architecture analysis from the analysis of synchronizations between cores by performing them in two separate analysis phases (WCET – worst-case execution time – and WCRT – worst-case response time analyses). This isolation has its advantages, such as a reduction of the complexity of each analysis phase, and a separation of concerns that facilitates the development of analysis tools. But isolation also has a major drawback: a loss in precision which can be significant. To consider only one aspect, to be safe the WCET analysis of each synchronization-free sequential code region has to consider an undetermined micro-architecture state. This may result in overestimated WCETs, and consequently on pessimistic execution time bounds for the whole parallel application.

The contribution of this work [56], [44] is an *integrated* WCET analysis approach that considers at the same time micro-architectural information and the synchronizations between cores. This is achieved by extending a state-of-the-art WCET estimation technique and tool to manage synchronizations and communications between the sequential threads running on the different cores. The benefits of the proposed method are twofold. On the one hand, the micro-architectural state is not lost between synchronization-free code regions running on the same core, which results in tighter execution time estimates. On the other hand, only one tool is required for the temporal validation of the parallel application, which reduces the complexity of the timing validation toolchain.

Such a holistic approach is made possible by the use of deterministic and composable software and hardware architectures (many-cores with no cache sharing and time-predictable interconnect, static assignment of the code and data to the memory banks). Such code can be written by hand or automatically synthesized using the Lopht tool 5.4 or other automatic parallelization techniques.

6.8. Real-time scheduling and code generation for time-triggered platforms

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

We have continued this year the work on real-time scheduling and code generation for time-triggered platforms. This work was mainly carried out as part of a bilateral collaboration with Astrium Space Transportation (now part of Airbus Defence and Space), which co-funded with the CNES the post-doctorate of Raul Gorcitz (started in September).

The work focused this year on the improvement of the real-time scheduling and code generation (the PhD work of T. Carle), and on determining their adequacy to Astrium's industrial needs (the post-doc of Raul Gorcitz). We have improved our specification, mapping, and code generation technique at all levels. We have extended the Lopht tool to allow automatic mapping and code generation for single-processor and multi-processor partitioned targets (using an ARINC 653-compliant OS).

6.9. Uniprocessor Real-Time Scheduling

Participants: Yves Sorel, Falou Ndoye, Daniel de Rauglaudre.

6.9.1. Formal Proofs of Uniprocessor Real-Time Scheduling Theorems

We continued writing a monograph about three formal proofs, done in 2011/2012, in Coq on scheduling of fixed priority real-time preemptive tasks: one about the scheduling conditions of strict periodicity and two about the worst response time in the case of preemptive deadline monotonic scheduling. This document contains about 120 pages for the moment.

6.9.2. Real-Time Scheduling with Exact Preemption Cost

We proposed a new schedulability condition for dependent tasks executed on a uniprocessor which takes into account the exact preemption cost. Unlike the work presented in [10] which achieves that goal only for fixed priority tasks, our schedulability condition considers fixed as well as dynamic priorities tasks. Thus, we can overcome priority inversions involved by data dependent tasks. The schedulability analysis based on this schedulability condition led to an off-line scheduler [42] described by a scheduling table. Therefore, we have proposed an on-line time-trigger scheduler which implements this scheduling table. Compared to classical on-line schedulers, the proposed approach has two benefits. On the one hand the cost of the task selection amounts only to read the task to be executed in the scheduling table built off-line, rather than using on-line a scheduling algorithm like RM, DM, EDF, etc. On the other hand this cost is fixed since it does not depend on the number of ready tasks. In addition, with our on-line scheduler we do not need to synchronize, on-line, the utilization of the shared memory data, due to dependences, because this synchronization is performed during the off-line schedulability analysis.

6.10. Multiprocessor Real-Time Scheduling

Participants: Yves Sorel, Laurent George, Dumitru Potop-Butucaru, Falou Ndoye, Aderraouf Benyahia, Cécile Stentzel, Meriem Zidouni.

6.10.1. Multiprocessor Partitioned Scheduling with Exact Preemption Cost

We finalized the work started in previous years on multiprocessor scheduling of preemptive independent real-time tasks with exact preemption cost [43].

This year we proposed a heuristic for the multiprocessor scheduling of preemptive dependent real-time tasks with exact preemption cost. We chose the partitioned approach that avoids migration of tasks and allows the utilization of the uniprocessor schedulability condition, previously proposed, that takes into account the exact preemption cost. In addition, this schedulability condition takes into account the inter-processor communications and guarantees that no data is lost. The result of such an off-line scheduling provided by the heuristic, is a scheduling table for every processor which includes also inter-processor communication tasks. We compared our multiprocessor scheduling heuristic with a Branch & Bound exact algorithm using the same schedulability condition. Our heuristic provides similar results and is very much faster.

6.10.2. Multiprocessor Semi-Partitioned Mixed Criticality Scheduling

We mainly focused on the mixed criticality scheduling problem applied to semi-partitioned scheduling considering a static pattern of migration for jobs. We have studied this problem in the context of Mixed Criticality (MC) scheduling, a promising approach that can be used to take into account applications of different criticality levels on the same platform. The goal of MC approach is to better utilize computing resources by allowing low criticality tasks to execute in conjunction with high criticality tasks when the system criticality is not high.

6.10.3. Gateway with Modeling Languages for Certified Code Generation

This work was carried out in the P FUI project 8.2.2 . We defined a SynDEx UML profile for functional specifications. We developed a gateway between the P pivot formalism and SynDEx. This gateway deals with the data-flow modeling part of the P formalism which is compliant with the Simulink subset blocks supported by the P project, except for the IF, FOR, MERGE and MUX blocks. Presently, we enhance the gatetway to include these blocks and we colloborate with the other partners to define the architectural part of the P formalism. This part is intended to replace the non functional specifications, presently described with the UML profile MARTE (Modeling and Analysis of Real-Time Embedded Systems).

6.10.4. SynDEx updates with new results

We released an alpha version of SynDEx V8. This version is based on a new textual language whose compiler may be launched with commandes-lines featuring various options. In Syndex V8, the adequation heuristic which performs the multiprocessor real-time schedulability analysis on multi-periodic applications, is based on the theorems and algorithms provided in the Mohamed Marouf's thesis defended last year in the team. These algorithms have been deeply improved for better consideration of data dependencies in the case of multiprocessor architectures. On the other hand, the new heuristic generates a scheduling table composed of, in addition to the usual permanent phase, a transient phase that takes into account the distribution constraints defined by the user in the multi-periodic applications as well as in the mono-periodic applications.

6.11. Probabilistic Real-Time Systems

Participants: Liliana Cucu-Grosjean, Adriana Gogonel, Codé Lo, Dorin Maxim, Cristian Maxim.

The adventof complex hardware, in response to the increasing demand for computing power in next generation systems, exacerbates some of the limitations of static timing analysis for the estimation of the worst-case execution time (WCET) estimation. In particular, the effort of acquiring (1) detail information on the hardware to develop an accurate model of its execution latency as well as (2) knowledge of the timing behaviour of the program in the presence of varying hardware conditions, such as those dependent on the history of previously executed instructions. These problems are also known as the timing analysis walls. The probabilistic timing analysis, a novel approach to the analysis of the timing behaviour of next-generation real-time embedded systems, provides answers to timing analysis walls. In [17], [48], [31] timing analysis attacks the timing analysis walls. We have also presented experimental evidence that shows how probabilistic timing analysis reduces the extent of knowledge about the execution platform required to produce probabilistically-safe and tight WCET estimations.

Based on existing estimations of WCET or minimal inter-arrival time, one may propose different probabilistic schedulability analyses [39]. These results were reported in the (PhD thesis of Dorin Maxim, mostly conducted in the Inria TRIO team (before its completion and the move to Aoste in Sept 2013).

2013 was also the year when through several invited talks [26], [28], [27], we had the opportunity to underline historical misunderstandings on probabilistic real-time systems. The most common is related to the notion of independence that is used with a wrong meaning by different papers.

CONVECS Project-Team

6. New Results

6.1. New Formal Languages and their Implementations

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.

6.1.1. Translation from LNT to LOTOS

Participants: Hubert Garavel, Frédéric Lang, Wendelin Serwe.

The LNT2LOTOS, LNT.OPEN, and LPP tools convert LNT code to LOTOS, thus allowing the use of CADP to verify LNT descriptions. These tools have been used successfully for many different systems (see § 6.5 and § 9.1).

In 2013, in addition to 15 bug fixes, the following enhancements have been brought to these tools:

- The list of predefined functions that can be generated automatically for list and set types has been enriched, so as to include all operations commonly found in programming languages.
- A new "sorted set" type was added to LNT, where the automatically generated insertion function preserves the invariant that all elements in the set are sorted in ascending order and have a single occurrence.
- The use of range and predicate types has been facilitated, by translating explicit type annotations by type conversions if necessary.
- An implicit type conversion is applied by assignments to a variable; this helps the type-checker to solve overloaded definitions.
- The generated LOTOS and C code has been modified to avoid spurious warning messages from the LOTOS and C compilers.
- The demo examples demo_19, demo_30, and demo_35 have been enhanced with an LNT version.

6.1.2. 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 2013, in addition to fixing four bugs, the type checking algorithm of CAESAR and CAESAR.ADT was entirely revised to display less and better messages in case of typing errors, avoiding cascading error messages, e.g., when an undefined variable or constant is used, or when an overloaded function is improperly used in a context where a unique type is expected.

Also, the CAESAR compiler found a new usefulness as a means to easily produce large-size, realistic Petri nets that can be used as benchmarks by the Petri net community. To make this possible, a new option was added to CAESAR to export the BPN (*Basic Petri Net*) file generated from a LOTOS specification. The definition of the BPN format was made more precise by adding semantic constraints. The CAESAR.BDD tool of CADP was enhanced with two new options, one that checks whether a BPN file satisfies all semantic constraints, and another one that converts a BPN file into PNML (*Petri Net Markup Language*) format.

This work has been done in coordination with Fabrice Kordon and Lom-Messan Hillah (UPMC/LIP6, Paris, France) for the MCC (*Model Checking Contest*) workshop [5]. H. Garavel was in charge of redesigning the model forms used for this contest. One Petri net generated using CAESAR was selected as a benchmark for MCC'2013 and five Petri nets generated using CAESAR have been submitted to MCC'2014.

6.1.3. Translation from an Applied Pi-Calculus to LNT

Participants: Radu Mateescu, Gwen Salaün.

The π-calculus is a process algebra defined by Milner, Parrow, and Walker two decades ago for describing concurrent mobile processes. Despite a substantial body of theoretical work in this area, only a few verification tools have been designed for analysing π-calculus specifications automatically. Our objective is to provide analysis features for the π-calculus by reusing the verification technology available for process algebras without mobility. For this purpose, we extended the original polyadic π-calculus with the data types and functions of LNT. This yields a general-purpose applied π-calculus, which is suitable for specifying mobile value-passing concurrent systems belonging to various application domains. Our approach is based on a novel translation from the finite control fragment of π-calculus to LNT, making possible the analysis of applied π-calculus specifications using all verification tools of CADP. This translation is fully automated by the PIC2LNT translator (see § 5.3).

In 2013, we continued our work on the applied π-calculus and its translation to LNT. This resulted in a new version PIC2LNT 3.0 of the tool, which fixes several bugs and brings the following improvements:

- A bounded replication operator was added to the language, which expresses the parallel execution of a fixed number of π-calculus agents. This operator is translated into LNT by instantiating the appropriate number of corresponding processes.
- A type Chan representing channel names was implemented, which can be freely combined with ordinary data types. This increases the versatility of the language by allowing, e.g., the definition of agents parametrized by sets of channel names.
- Several options were added to the tool for enhancing its ergonomy and tuning the state space generation (specify the set of private channels that can be created, generate the state space of a particular agent).

A paper describing this work has been published in an international conference [16].

6.1.4. Translation from EB3 to LNT

Participants: Frédéric Lang, Radu Mateescu.

In collaboration with Dimitris Vekris (University Paris-Est Créteil), we considered a translation from the EB3 language [39] for information systems to LNT. EB3 has a process algebraic flavor, but has the particularity to contain so-called *attribute functions*, whose semantics depend on the history of events. We have proposed a formal translation scheme, which ensures the strong equivalence between the LTSs corresponding to an EB3 specification and to the LNT code generated. A prototype translator has been developed at University Paris-Est Créteil, which enables EB3 specifications to be formally verified using CADP.

In 2013, a paper has been published in an international conference [19].

6.1.5. Coverage Analysis for LNT

Participants: Gwen Salaün, Lina Ye.

In the classic verification setting, the designer has a specification of a system in a value-passing process algebra, a set of temporal properties to be verified on the corresponding LTS model, and a data set of examples (test cases) for validation purposes. At this stage, building the set of validation examples and debugging the specification is a complicated task, in particular for non-experts.

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

In 2013, we proposed a new framework for debugging value-passing process algebraic specifications by means of coverage analysis and we illustrated our approach with LNT. We define several coverage notions before showing how to instrument the specification without affecting its original behavior. Our approach helps the specifier to find dead code, ill-formed conditional structures, and other errors in the specification, but also to improve the quality of a data set of examples used for validation purposes. We have implemented a prototype tool, named CAL, for automating the verification of coverage analysis, and we applied it to several real-world case studies in different application areas. A paper has been submitted to an international conference.

6.1.6. Other Compiler Developments

Participants: Soraya Arias, Hubert Garavel, Frédéric Lang, Wendelin Serwe.

- In co-operation with Jérôme Hugues (ISAE, Toulouse), we investigated the translation of AADL (*Architecture Analysis and Design Language*) into LNT. An AADL example was manually tackled, leading to the conclusion that LNT could be a suitable target language for translating a large fragment of AADL.

  In co-operation with Holger Hermanns (Saarland University, Germany) and Joost-Pieter Katoen (RWTH Aachen, Germany), we prepared a contribution for the AADL standardization committee to detail semantics issues of the GSPN (*Generalized Stochastic Petri Nets*) model.
- We continued our work on the FLAC tool, which translates the Fiacre intermediate language into LOTOS to enable verification using CADP. In 2013, we eliminated spurious compilation warnings, we removed the definitions of integer operations *div* and *mod*, which have been added to a standard LOTOS library, and we improved the encoding of integer numbers. These changes have led to revisions 76 to 79 of the FLAC code, which is available on the development forge dedicated to Fiacre compilers [6].
- In co-operation with Holger Hermanns, we started studying the PseuCo language that is being defined and implemented at Saarland University. Developed from an educational perspective as a means to teach concurrency theory to bachelor students, PseuCo combines features from Java and Go, the language promoted by Google for concurrent programming. PseuCo supports both message-passing and shared-memory concurrency in a way that is easy to use and that can readily be transferred to Java, Go, or other mainstream languages. PseuCo has been awarded with the 2013 German national "*Preis des Fakultätentages Informatik*" for its innovative role in undergraduate education.

  In 2013, we undertook the manual translation of various PseuCo sample programs into LNT and started enhancing LNT with features that would enable automated PseuCo-to-LNT translation. We also reviewed a PseuCo-to-CCS translator recently developed at Saarland University and wrote an evaluation report for this software.

6.2. Parallel and Distributed Verification

6.2.1. Manipulation of Partitioned LTSs

Participants: Hubert Garavel, Radu Mateescu, Wendelin Serwe.

For distributed verification, CADP provides the PBG format, which implements the theoretical concept of *Partitioned LTS* [45] and provides a unified access to an LTS distributed over a set of remote machines.

In 2013, we continued the development of the prototype tool PBG_OPEN, which is an OPEN/CAESAR-compliant compiler for the PBG format, enabling the use of all CADP on-the-fly verification tools on a partitioned LTS. The main advantage of PBG_OPEN is that it can use the memory of several machines to store the transition relation of a partitioned LTS. Therefore, PBG_OPEN can explore on-the-fly large partitioned LTSs that could not be explored using other tool combinations. To reduce the amount of communications, PBG_OPEN can use a cache to store already encountered states, together with their outgoing transitions.

[6] http://gforge.enseeiht.fr/projects/fiacre-compil

We also developed another prototype tool, named PBG_INVERT, which changes the storage of the transitions of a partitioned LTS, transforming a partitioned LTS where each fragment stores the transitions leading to the states of the fragment (as generated by DISTRIBUTOR) into a partitioned LTS where each fragment stores the transitions going out from the states of the fragment. Adding this transformation step yields a reduction of up to 25% of the overall execution time, when verifying the partitioned LTS with PBG_OPEN. We experimented all these tools on the Grid'5000 computing infrastructure [31] using up to 512 distributed processes. These experiments confirmed the good scalability of our distributed LTS manipulation approach. A paper describing this work has been published in an international conference [13].

6.2.2. Distributed Code Generation for LNT

Participants: Hugues Evrard, Frédéric Lang.

Rigorous development and prototyping of a distributed verification algorithm in 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.

In 2013, we formally modelled and verified several existing synchronization protocols. This revealed an error in one of them, which led to a publication in an international conference [12]. Based on this study, we selected a suitable protocol and adapted it to the LNT synchronization operators.

Using this protocol, we developed a prototype distributed code generator, taking as input the model of a distributed system, described as a set of LNT processes and their parallel composition written in EXP. The LNT.OPEN and CAESAR tools are used to obtain the sequential implementation of each LNT process, and the EXP.OPEN tool is used to compute the possible interactions between processes. Then, our prototype generates the corresponding implementation of the distributed synchronization protocol and all necessary glue code between processes and the protocol. Our prototype automatically performs all these steps, such that a complete and runnable distributed implementation can easily be obtained from the original model.

So far, our prototype manages synchronizations with no data or data of enumerated types only, in which case the implementation checks that data values and types match before allowing a synchronization.

6.3. Timed, Probabilistic, and Stochastic Extensions

Participants: Hubert Garavel, Frédéric Lang, Radu Mateescu.

Process calculi provide a suitable formal framework for describing and analyzing concurrent systems, but need to be extended to model refined aspects of these systems. For instance, it may be necessary to represent probabilistic choices (in addition to deterministic and nondeterministic choices) as well as delays and latencies governed by probability laws. Many such extensions have been proposed in the literature, some of which have been implemented in software tools and applied to nontrivial problems. In particular, two of these extensions (namely, *Interactive Markov Chains* and *Interactive Probabilistic Chains*) are implemented in CADP. Despite these achievements, the state of the art is not satisfactory as the extended languages primarily focus on the probabilistic and stochastic aspects, leaving away the expressive and user-friendly features that process calculi provide for describing conventional concurrent systems.

In 2013, we did the following steps to progress our agenda of bridging the gap between functional verification and quantitative evaluation:

- We equipped CADP with a new tool named BCG_CMP, which enables to compare quantitative models modulo probabilistic and stochastic variants of strong bisimulation and branching bisimulation. Such comparison relations were not available in the BISIMULATOR tool that already existed in CADP.

- We investigated the feasibility of creating interconnections between mainstream verification tools for probabilistic and stochastic systems. In a first step, we focused on the DTMC (*Discrete-Time Markov Chain*) model and on three mainstream tools: CADP (Grenoble), MRMC (Aachen), and PRISM (Birmingham-Oxford).

  We developed translation tools to perform conversions between the various formats of these tools (".aut" and ".bcg" for CADP, ".tra/.sta/.lab" for MRMC, ".pm" and ".tra/.sta/.lab" for PRISM). So doing, we reported one bug in MRMC and five minor issues in PRISM. By discussing with Dave Parker (University of Birmingham), we contributed to the introduction in PRISM 4.1 of two new options "-importmodel" and "-exportmodel" that greatly simplify exchanges of models between PRISM and other tools.

  We developed a generator of random DTMCs in CADP, MRMC, and PRISM formats, and undertook the construction of a collection of DTMCs, which we used to compare the performance and scalability of CADP and PRISM.
- We started to investigate the evaluation of temporal logic properties on extended DTMCs, in which transitions are labeled with probabilities and optional actions. For this purpose, we developed a new prototype XTL library (consisting of XTL and C code) encoding the PCTL (*Probabilistic CTL*) temporal logic [50]. This new PCTL library enables the specifier to combine data-based, discrete-time, and probabilistic properties of DTMCs in a uniform way.

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

6.4.1. Compositional Model Checking

Participants: Frédéric Lang, Radu Mateescu.

We have continued our work on partial model checking following the approach proposed in [26]. Given a temporal logic formula φ to be evaluated on a set S of concurrent processes, partial model checking consists in transforming φ into another equivalent formula φ' to be evaluated on a subset of S. Formula φ' is constructed incrementally by choosing one process P in S and incorporating into φ the behavioral information corresponding to P — an operation called quotienting. Simplifications must be applied at each step, so as to maintain formulas at a tractable size.

In 2013, we extended the approach to handle fairness operators of alternation depth two, and we conducted new experiments. This resulted in a new version of the PMC prototype tool (see § 5.4) supporting all features of the input language of EXP.OPEN 2.1. An article has been published in an international journal [5].

6.4.2. On-the-Fly Test Generation

Participants: Radu Mateescu, Wendelin Serwe.

In the context of the collaboration with STMicroelectronics (see § 6.5.1 and § 7.1), we studied techniques for testing if an implementation is conform to a formal model written in LNT. Our approach is inspired by the theory of conformance testing [68], as implemented for instance in TGV [53] and JTorX [30].

We developed two prototype tools. The first tool implements a dedicated OPEN/CAESAR-compliant compiler for the particular asymmetric synchronous product of the model and the test purpose, and uses slightly extended generic components for graph manipulation (τ-compression, τ-confluence reduction, determinization) and resolution of Boolean equation systems. The second tool generates the complete test graph, which can be used to extract concrete test cases or to drive the test of the implementation. The principal advantage of our approach compared to existing tools is the use of LNT for test purposes, facilitating the manipulation of data values.

In 2013, we continued the development of these tools, with a focus on reducing execution time. We also implemented a prototype tool to extract from a complete test graph one or all test cases of minimal depth. We experimented with these tools on two case-studies, namely the ACE coherence protocol (see § 6.5.1) and the EnergyBus (see § 6.5.5).

6.4.3. Equivalence Checking

Participant: Frédéric Lang.

Equivalence relations can be used for verification in two complementary ways: for the minimization of an LTS and the comparison of two LTSs.

In 2013, we worked along the following lines:

- We added observational equivalence (following a request from LAAS-CNRS) as well as divergence-sensitive branching bisimulation (together with its stochastic and probabilistic variants) in BCG_MIN.
- We improved the speed of BCG_MIN in the case of branching reduction applied to a graph with a high branching factor and many internal transitions, by correcting a function that has a quadratic complexity instead of a linear one.
- We added the new tool BCG_CMP, which takes as input two BCG graphs and checks whether they are equivalent modulo a relation chosen among strong and branching bisimulation (and their stochastic and probabilistic variants), divergence-sensitive branching bisimulation, or observational equivalence. BCG_CMP checks equivalence using the partition-refinement algorithm of BCG_MIN. We compared BCG_CMP and BISIMULATOR on the VLTS benchmark suite [7], showing that BCG_CMP is generally slightly less efficient than BISIMULATOR for comparisons yielding a FALSE result, but much more efficient than BISIMULATOR for comparisons yielding a TRUE result.
- The new tool BCG_CMP as well as the new equivalence relations added to BCG_MIN have been added to the EUCALYPTUS graphical user interface and to the SVL scripting language.

6.4.4. Other Software Developments

The OPEN/CAESAR environment was enhanced with a new generic library (named CAESAR_CACHE_1) for manipulating hierarchical caches, with 15 built-in replacement strategies and the possibility to define new ones.

We also maintained the CADP toolbox, taking into account the feedback received from numerous users in the world. In addition to fixing 41 bugs, we evolved CADP to support the latest versions of Windows, Cygwin, Mac OS X, and their corresponding C compilers. The documentation for installing CADP has been updated and shortened. Finally, support for Sparc, Itanium, and PowerPC processors was dropped at the end of 2013 based on the observation that these architectures are almost no longer used among the CADP user community.

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 [25]. In a first step, we developed a formal LNT model (about 3200 lines of LNT) of a system consisting of an ACE-compliant cache coherent interconnect, processors, and a main memory. The model is parametric and can be instantiated with different configurations (number of processors, number of cache lines, number of memory lines) and different sets of supported elementary ACE operations (currently, a representative subset of 15 operations), including an abstract operation that represents any other ACE operation. We handled the global requirements of the ACE specification using a constraint oriented programming style, i.e., by representing each global requirement as a dedicated process observing the global behaviour and inhibiting incorrect executions.

[7] http://cadp.inria.fr/resources/vlts

In a second step, we generated for several configurations the corresponding LTS (up to 100 million states and 350 million transitions). We wrote two liveness properties in MCL expressing that each read (respectively write) transaction is executed until its termination. We also wrote two properties expressing cache coherence and data integrity. This required to transform state-based properties into action-based properties, by adding information about the cache state to actions executed by the cache. For all considered configurations, we checked these properties using parametric SVL scripts (about 100 lines) and EVALUATOR. For some scenarios without the processes representing the global requirements, EVALUATOR generated counterexamples for the cache coherence and data integrity. We are currently using these counterexamples to derive test cases for the architecture under design at STMicroelectronics.

This work led to publications [21], [15].

6.5.2. Choreography-based Communicating Systems

Participants: Radu Mateescu, Gwen Salaün, Lina Ye, Kaoutar Hafdi.

Choreographies are contracts specifying interactions among a set of services from a global point of view. These contracts serve as reference for the further development steps of the distributed system. Therefore, their specification and analysis is crucial to avoid issues (e.g., deadlocks) that may induce delays and additional costs if identified lately in the design and development process.

In 2013, we have obtained the following results:

- In collaboration with Meriem Ouederni (University of Toulouse) and Tevfik Bultan (University of California at Santa Barbara), we have proposed a branching definition of the synchronizability property, which identifies systems whose interaction behavior remains the same when asynchronous communication is replaced with synchronous communication. We have also shown how these results can be used for checking the compatibility of a set of asynchronously communicating components [17].
- In collaboration with Matthias Güdemann (Systerel), we have defined sufficient conditions for checking the repairability property, which indicates whether realizability can be enforced for choreography-based communicating systems using distributed controllers. A paper has been submitted to an international conference.
- We have proposed an approach for computing the degree of parallelism of BPMN processes using model checking techniques. A paper has been submitted to an international conference.
- In collaboration with Pascal Poizat (University of Paris Ouest Nanterre), we have been working on the development of the VerChor platform, which aims at assembling all the verification techniques and tools automating the analysis of choreography specifications [14].

6.5.3. Deployment and Reconfiguration Protocols for Cloud Applications

Participants: Rim Abid, Gwen Salaün.

We collaborated with Noël de Palma and Fabienne Boyer (University Joseph Fourier), Xavier Etchevers and Thierry Coupaye (Orange Labs, Meylan, France) in the field of cloud computing applications, which are complex distributed applications composed of interconnected software components running on distinct virtual machines. Setting up, (re)configuring, and monitoring these applications involves intricate management protocols, which fully automate these tasks while preserving application consistency as well as some key architectural invariants.

In 2013, we focused on the reliability of the self-configuration protocol [23]. This protocol always succeeds in deploying a cloud application, even when facing a finite number of virtual machine or network failures. Designing such highly parallel management protocols is difficult, therefore formal modelling techniques and verification tools were used for validation purposes. These results were accepted for publication in an international conference [11]. Also, an experience export on the verification tasks for such (re)configuration protocols has been published in an international journal [8].

We have also worked on the design and verification of a reconfiguration protocol, where virtual machines interact altogether using a publish-subscribe messaging system. The verification of this protocol with CADP helped to refine several parts of the protocol and correct subtle bugs. These results have been published in an international conference [10]. In collaboration with Francisco Durán (University of Málaga), we have also worked on the design of a variant of this reconfiguration protocol, where the virtual machines interact via FIFO buffers. A paper has been submitted to an international conference.

6.5.4. Networks of Programmable Logic Controllers

Participants: Hubert Garavel, 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 a pivot language that will enable the connection to testing and verification tools covering the synchronous and asynchronous aspects. Our work focuses on the translation from the pivot language to LNT, which will provide a direct connection to all verification functionalities of CADP, in particular model checking and equivalence checking.

In 2013, we studied the existing approaches and languages that address formal modeling and verification of GALS systems. We concluded that the current landscape lacks general-purpose, flexible, and formal representation of GALS systems suitable for efficient verification. To fulfill this requirement, we have designed GRL (*GALS Representation Language*), a language with user-friendly syntax and formal semantics, to efficiently model GALS systems for the purpose of formal verification. GRL targets GALS systems consisting of networks of synchronous systems interacting with their environments and communicating via asynchronous media. GRL draws mainly from two foundations. Regarding asynchronous concurrency, GRL builds upon process calculi (in particular LNT). Thereby, it leverages process calculi expressiveness, versatility, and verification efficiency. Regarding synchronous features, GRL holds a dataflow-oriented model based on the dataflow diagram model (also called block-diagram model). The GRL synchronous model inherits from the simplicity and modularity of the block-diagram model.

We defined the lexical and the abstract syntax of GRL (about 80 grammar rules), its static semantics (about 150 binding, typing, and initialization rules), and its dynamic semantics (about 20 structured operational semantics rules). Using the SYNTAX and LOTOS NT compiler construction technology, we started the development of a prototype translator GRL2LNT (about 8000 lines). The tool currently performs the lexical and syntactic analysis of GRL programs, together with some static semantic checks. A database containing about 30 examples of GRL programs has been constructed and used for non-regression testing of GRL2LNT. A reference manual for GRL (130 pages up to now) containing the definition of the language and its translation to LNT has been written. A paper presenting the GRL language has been submitted to an international conference.

Regarding the analysis of PLC networks by equivalence checking, we defined variants of classic equivalence relations (strong, τ*.a, and branching) for comparing the Mealy machine corresponding to a PLC network with the Moore machine corresponding to its external behaviour. We reformulated the verification problem as the resolution of a Boolean equation system, and we developed a prototype tool, based on the CAESAR_SOLVE_1 library, for the on-the-fly comparison of a Mealy and a Moore machine modulo the strong or the τ*.a equivalences.

6.5.5. EnergyBus Standard for Connecting Electric Components

Participants: Hubert Garavel, Wendelin Serwe.

The EnergyBus [8] 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[9] 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 [49], with the active collaboration of CONVECS.

In 2013, CONVECS provided help in modelling using the LNT language and the TGV tool, and enhanced the CADP toolbox to address a number of issues reported by Saarland University. At present, this LNT specification (1670 lines) is used for generating test suites using the TGV tool [53]. The formal modelling prompted for modifications in the EnergyBus standard and the generated test suites revealed three unknown bugs in an industrial CANopen implementation.

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 co-operation with Gaëlle Calvary, Eric Ceret, 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 [35] designed to provide human operators with an overview of a running nuclear 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 status information and alarms resulting from faults, disturbances, or unexpected events in the plant. 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.

In 2013, CONVECS contributed to the following results. Based upon the available information published by EDF, a formal specification in LNT of this new-generation interface was developed (2600 lines). This specification not only encompasses 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, a few desirable properties of the interface have been expressed in the MCL language of CADP and verified on the LNT model.

So doing, three main difficulties have been faced. The description of the prototype available in the published literature is not exhaustive, which required us to provide those missing details needed to obtain a realistic model. Quite often, we faced a combinatorial explosion in the number of states of the model, which forced us to restrict the complexity of operator behaviour and fault models. Finally, this case study revealed several LNT-specific issues, which triggered enhancements in the LNT language and tools.

[8] http://www.energybus.org
[9] http://www.can-cia.org

Hycomes Team

5. New Results

5.1. Hybrid Systems Modeling

Participants: Albert Benveniste, Benoît Caillaud.

5.1.1. Type-Based Analysis of Causality Loops In Hybrid Systems Modelers

Explicit hybrid systems modelers like Simulink / Stateflow allow for programming both discrete- and continuous-time behaviors with complex interactions between them. A key issue in their compilation is the static detection of algebraic or causality loops. Such loops can cause simulations to deadlock and prevent the generation of statically scheduled code. We have addressed this issue for a hybrid modeling language that combines synchronous Lustre-like data-flow equations with Ordinary Differential Equations (ODEs) [6], [9]. We introduce the operator last(x) for the left-limit of a signal x. This operator is used to break causality loops and permits a uniform treatment of discrete and continuous state variables. The semantics relies on non-standard analysis, defining an execution as a sequence of infinitesimally small steps. A signal is deemed causally correct when it can be computed sequentially and only progresses by infinitesimal steps outside of discrete events. The causality analysis takes the form of a simple type system. In well-typed programs, signals are proved continuous during integration and can be translated into sequential code for integration with off-the-shelf ODE solvers. The effectiveness of this system is illustrated with several examples written in Zélus [9], a Lustre-like synchronous language extended with hierarchical automata and ODEs.

5.1.2. Semantics of multi-mode DAE systems

Hybrid systems modelers exhibit a number of difficulties related to the mix of continuous and discrete dynamics and sensitivity to the discretization scheme. Modular modeling, where subsystems models can be simply assembled with no rework, calls for using Differential Algebraic Equations (DAE). In turn, DAE are strictly more difficult than ODE. They require sophisticated pre-processing using various notions of index before they can be submitted to a solver. We have studied some fundamental issues raised by the modeling and simulation of hybrid systems involving DAEs [10]. The objective of this work is to serve for the evolution and the design of future releases of the Modelica language for such systems. We focus on the following questions:

- What is the proper notion of index for a hybrid DAE system?
- What are the primitive statements needed for a DAE hybrid systems modeler?

The differentiation index for DAE explicitly relies on everything being differentiable. Therefore, generalizations to hybrid systems must be done with caution. We propose to rely on non-standard analysis for this. Non-standard analysis formalizes differential equations as discrete step transition systems with infinitesimal time basis. We can thus bring hybrid DAE systems to their non-standard form, where the notion of difference index can be firmly used. From this study, general hints for future releases of Modelica can be drawn.

5.2. Surgical Process Mining with Test and Flip Net Synthesis

Participant: Benoît Caillaud.

Surgical process modeling aims at providing an explicit representation of surgical procedural knowledge. *Surgical process models* are inferred from a set of surgical procedure recordings, and represent in a concise manner concurrency, causality and conflict relations between actions. In the context of the S3PM project (Section 6.1), we have investigated the use of *test and flip* nets, a mild extension of flip-flop nets, to represent surgical process models. A test and flip net synthesis algorithm, based on linear algebraic methods in the $Z/2Z$ ring is detailed. Experimental results regarding the use of this synthesis algorithm to automate the construction of simple surgical process models are also presented.

[9] http://zelus.di.ens.fr

MUTANT Project-Team

6. New Results

6.1. Operational Timed Semantics

Participants: José Echeveste, Jean-Louis Giavitto, Florent Jacquemard, Arshia Cont.

One common use-case of real-time musical interactions between musicians and computers is *Automatic Accompaniment* where the system is comprised of a real-time machine listening system that in reaction to recognition of events in a score from a human performer, launches necessary actions for the accompaniment section. While the real-time detection of score events out of live musicians' performance has been widely addressed in the literature, score accompaniment (or the reactive part of the process) has been rarely discussed. In [13], we are trying to deal with this missing component in the literature from a programming language perspective. We show how language considerations would enable better authoring of time and interaction during programming/composing and how it addresses critical aspects of a musical performance (such as errors) in real-time. We sketch the real-time features required by automatic musical accompaniment seen as a reactive system and formalize the timing strategies for musical events taking into account the various temporal scales used in music. Various strategies for the handling of synchronization constraints and the handling of errors are presented.

The behavior of the system *Antescofo* have been formally modeled as a *network of parametric timed automata*. The model obtained provides operational semantics for the input scores, in particular the interaction between the instrumental and electronic parts and the timing and error handling strategies mentioned below. This approach enables better authoring of time and interaction during programing/composing, permitting to use state of the art software verification tools for the static analysis of Antescofo scores. It also provides means to address critical aspects of musical performances in real-time.

6.2. Timed Static Analysis of Interactive Music Scores

Participant: Florent Jacquemard.

It is well known that every musician performance of the same work will differ from another. It is therefore a challenging task to be able to predict the behavior of interactive music systems like *Antescofo* in response to any possible performance, and prevent unwanted outcomes. With Léa Fanchon, we have been working on a module for timing analysis of augmented scores that complements the real-time score authoring and performance in *Antescofo*, with the aim of exploring possible behavior of authored scores with respect to possible deviations in human musician performance.

For this purpose we have studied [24] the application of formal models and methods from the literature of real-time systems verification to the static analysis of interactive music systems. We have considered in particular the good parameters problem, which consists in synthesizing a set of timing parameter valuations (representing performances here) guarantying a good behavior of the system analyzed. The methods presented in [24] have been applied to *Antescofo*, providing the following input to users:

- Evaluation of robustness of the program with respect to the environment's (musician's performance) temporal variations,
- Feedback to programmers or artists on critical synchronization points for better programming.

This study is one of the first of this kind in computer music literature, and the methods presented are general enough to apply to the verification of other interactive multimedia applications.

6.3. Automating the Generation of Test Suites for Antescofo

Participants: Florent Jacquemard, Clément Poncelet.

Clément Poncelet has started to develop during his Master thesis [35] a framework for black box conformance testing of *Antescofo*. This work is pursued in a PhD supported by DGA and Inria. The most important task in this context is the generation of relevant test data for the system, given an augmented score in *Antescofo* language. This data includes input, containing musical events (notes, chords etc) together with their timings. In a sense, the input data simulates a musical execution of the score. The input data must then be passed to *Antescofo* for black-box execution, in order to observe the system's reactions and compare them the expected output. For the latter comparison task, we need to be able to define the expected output, hence to have a formal model of the expected behavior of the system on the given score. For this purpose, we are using models of the system made of timed automata, which are computed automatically from given music scores. Then, we use tools from the UPPAAL suite [40] in order to generate testing data, based on relevant covering criteria and a formal model of the environment (i.e. the musician). This work has been presented at the poster session of MSR 2013 (national colloquium on modeling reactive systems) and a journal paper is in preparation.

6.4. Synchronous Embedding of Antescofo DSL

Participants: Arshia Cont, Jean-Louis Giavitto, Florent Jacquemard.

Antescofo can been seen as the coupling of a listening machine and a real-time reactive system. Therefore, it faces some of the same major challenges as embedded systems. We have been working with Guillaume Baudart, Louis Mandel, and Marc Pouzet (EPI Parkas, ENS) in strengthening the ties between the reactive aspects of *Antescofo* and that of synchronous languages, in particular ReactiveML [44]. In [17], we present a synchronous semantics for the core language of Antescofo and an alternative implementation, based on an embedding inside the synchronous language ReactiveML [44]. The semantics reduces to a few rules, is mathematically precise and leads to an interpretor of a few hundred lines whose efficiency compares well with that of the current implementation. On all musical pieces we have tested, response times have been less than the reaction time of the human ear. Moreover, this embedding permitted the prototyping of several new programming constructs. Some examples are available, together with the ReactiveML source code at http://reactiveml.org/emsoft13/.

6.5. Tree Structured Representation of Symbolic Temporal Data

Participant: Florent Jacquemard.

In traditional music notation, in particular in the languages used for the notation of mixed music such as Antescofo DSL, the durations are not expressed by numerical quantities but by symbols representing successive subdivisions of a reference time value (the beat). For this reason, trees data structures are commonly used for the symbolic representation of rhythms in computer aided composition softwares such as OpenMusic (developed at Ircam).

Following this idea, we have been working on using several tree automata techniques for the challenging and long-standing problem of automatic transcription of rhythm (in traditional music notations) from symbolic input data (symbolic traces with timestamps in ms, like e.g. in MIDI format). To summarize, the main problem in rhythm transcription is to find an acceptable balance between timing precision (the goal is to minimize the loss obtained by transformation of ms timing values into fractions of beats) and the complexity of the notation obtained. The relative importance of these two measures may vary largely according to the user (composer), his workflow, and the musical style considered. It is therefore important to be able to control this balance during the transcription process, in order to adapt to the case of users. In traditional approaches, the transcription is done by an alignement of the input trace on a grid, and the two measures (precision of the grid and complexity) are either defined by parameters fixed a priori or hardcoded e.g. for a precise musical style and composition workflow. During two internships co-supervised by Jean Bresson (Ircam, main developer of OpenMusic) and Florent Jacquemard, we have been studying more flexible new approaches, based on computations on the tree representation of rhythms.

Pierre Donat-Bouillud (L3 ENS Rennes) [29] has worked on an approach by transformation of trees following some rewrite rules. The general idea is to start with a complex tree representing timings very close to the input data, and to simplify it by rewriting until an acceptable level of complexity is reached. The rewrite rules are either generic (defining an equational theory of rhythm notation) or user defined (defining approximations). This approach has been implemented in an OpenMusic library.

Adrien Maire (M1 ENS Cachan) has studied another very promising approach based on stochastic tree automata learning in an interactive authoring scenario. The generated automaton is supposed to represent (by the weighted tree langage it defines) the expected complexity of rhythm notations (i.e. the user's "style").

Moreover, we have following other work on several classes of tree recognizers and tree transformations which could be of interest in this context. With Luis Barguñó, Carlos Creus, Guillem Godoy, and Camille Vacher, [11] we define a class of ranked tree automata called TABG generalizing both the tree automata with local brother tests of Bogaert and Tison [37] and with global equality and disequality constraints (TAGED) of Filiot et al. [39]. TABG can test for equality and disequality modulo a given flat equational theory between brother subterms and between subterms whose positions are defined by the states reached during a computation. In particular, TABG can check that all the subterms reaching a given state are distinct. This constraint is related to monadic key constraints for XML documents, meaning that every two distinct positions of a given type have different values. We have proven decidability of the emptiness problem for TABG. This solves, in particular, the open question of decidability of emptiness for TAGED. We further extended our result by allowing global arithmetic constraints for counting the number of occurrences of some state or the number of different equivalence classes of subterms (modulo a given flat equational theory) reaching some state during a computation. We also adapt the model to unranked ordered terms. As a consequence of our results for TABG, we prove the decidability of a fragment of the monadic second order logic on trees extended with predicates for equality and disequality between subtrees, and cardinality.

With Michaël Rusinowitch (EPI Cassis), we have introduced in [25] an extension of unranked tree automata called bi-dimensional context-free hedge automata. The languages they define are context free in two dimensions: in the the sequence of successors of a node and also along paths. This formalism is useful for the static type-checking of tree transformations such as XML updates defined in the W3C XQuery Update Facility. We have developed with the same author in the past years a general framework for the verification of unranked (XML) tree transformations based on tree automata techniques. It has been presented this year in an invited keynote [16]. We have also presented with Emmanuel Filiot and Sophie Tison a survey on tree automata with constraints [33] during a Dagstuhl Seminar (number 13192) on tree transducers and formal methods.

6.6. Online Automatic Structure Discovery of Audio Signals

Participants: Arshia Cont, Vincent Lostanlen [MS Internship].

Following recent team findings in [12] and the framework introduced in [4], we pursued the problem of automatic discovery of audio signals using methods of information geometry through a Masters Thesis undertaken by Vincent Lostanlen (MS ATIAM) [34]. This work introduces a novel way of representing and calculating *Similarity Matrices* for continuous multimedia signals and in real-time. In this approach, the signal is first segmented into homogeneous chunks using the change detection algorithm proposed by the team in [12], and proposes a method for constituting similarity relations between segments using *Bregman Information Geometry* and exploiting intersections between information balls.

Compared to traditional approaches to similarity matrix computing, the approach proposed in [34] is strictly on-line (thus suitable for real-time computing) and provides a sparse view of audio structures. We will pursue this project by increasing its robustness and evaluating results on larger databases including other timed-signals such as video.

6.7. Temporal Coherency Criterion for Alignment Inference Algorithms

Participants: Philippe Cuvillier [PhD Student], Arshia Cont.

The question of modeling time and duration is of utmost importance for stability and robustness of real-time alignment algorithms and constitute one of the major success factors for the *Antescofo* listening machine described in [2]. Meanwhile, regular algorithms undergo stability in highly uncertain environments where observations obtained from the signal are highly uninformative and temporal information is of crucial importance.

PhD student Philippe Cuvillier defined *Coherency Criteria* for such applications and attempted to formalize such criteria in terms of probabilistic models and inference algorithms in case of Hidden Semi-Markov Chains. The results show that not all probabilistic families meet such criteria including some commonly used by engineers and designers. Preliminary results are submitted for publications and experimental results are being pursued.

PARKAS Project-Team

6. New Results

6.1. Reactive Programming

Participants: Guillaume Baudart, Louis Mandel, Cédric Pasteur, Marc Pouzet.

ReactiveML is an extension of OCaml with synchronous concurrency, based on synchronous parallel composition and broadcast of signals. The goal is to provide a general model of deterministic concurrency inside a general purpose functional language to program reactive systems. It is particularly suited to program discrete simulations, for instance of sensor networks.

One of the current focus of the research is being able to simulate huge systems, composed of millions of agents, by extending the current purely sequential implementation in order to be able to take advantage of multi-core and distributed architectures. This goal has led to the introduction of a new programming construct, *reactive domain*, which allows to define local time scales. These domains help for the distribution of the code but also increase the expressiveness of the language. In particular, it allows to do time refinement. A paper on this new construct and the related static analysis has been published [20]. An extended version is under submission.

We continued the work on a new reactivity analysis which ensures that a process can not prevent the other ones to from executing. This analysis has published in [19]. An English version is under submission.

The runtime of ReactiveML has been cleanup and a multi-threaded implementation has been developed. A paper describing this new implementation will be published in [27].

All these novelties has been described precisely in the PhD thesis of Cédric Pasteur [1].

During the year, ReactiveML has also bee applied to *mixed music*. Mixed music is about live musicians interacting with electronic parts which are controlled by a computer during the performance. It allows composers to use and combine traditional instruments with complex synthesized sounds and other electronic devices. There are several languages dedicated to the writing of mixed music scores. Among them, the Antescofo language coupled with an advanced score follower allows a composer to manage the reactive aspects of musical performances: how electronic parts interact with a musician. However these domain specific languages do not offer the expressiveness of functional programming.

We defined a synchronous semantics for the core language of Antescofo and an alternative implementation based on an embedding inside ReactiveML [9]. The semantics reduces to a few rules, is mathematically precise and leads to an interpretor of only a few hundred lines. The efficiency of this interpretor compares well with that of the actual implementation: on all musical pieces we have tested, response times have been less than the reaction time of the human ear. Moreover, this approach offers to the composer recursion, higher order, inductive types, as well as a simple way to program complex reactive behaviors thanks to the synchronous model of concurrency on which ReactiveML is built [10].

6.2. n-Synchronous Languages

Participants: Albert Cohen, Adrien Guatto, Louis Mandel, Marc Pouzet.

Synchronous programming languages in the vein of Lustre were designed for critical real-time systems. They are, however, not that well adapted to embedded applications with more pressing computational needs, since the generated code will usually not contain loops or arrays.

An essential task of a Lustre compiler is to determine whether a program can be executed within bounded memory. This process is called the "clock calculus", and consists in mapping every item of each program stream to a logical date in a global, discrete time scale. For a given stream, the mapping itself is called a "clock", and is a strictly increasing function from stream positions to natural numbers representing ticks: two items cannot be computed at the same time. In practice, this function is represented as an infinite binary stream where the boolean b_i denotes presence (or absence) in the corresponding data stream at the i-th instant.

In recent work, Guatto, Cohen, Mandel and Pouzet considered the extension of the Lustre and Lucid Synchrone clock calculus to allow computing several values instantaneously. This simple idea has a deep impact on all aspects of the language: - its denotational semantics has to account for bursts of values; - the clock calculus now features integers rather than booleans: each integer denotes the size of the burst at the corresponding instant; - causality analysis has to take bursts into account when rejecting self-referential programs; - the code generation process translates bursts to arrays and clocks to counted loops.

A prototype implementation exploiting this idea and generating C code with loops is underway and a paper describing the base of the clock calculus will be published [26].

This work extends nicely the n-synchronous model that introduced a way to compose streams which have *almost the same clock* and can be synchronized through the use of a finite buffer.

6.3. Mechanization of AODV loop freedom proof

Participant: Timothy Bourke.

The Ad hoc On demand Distance Vector (AODV) routing protocol is described in RFC3561. It allows the nodes in a Mobile Ad hoc Network (MANET) to know where to forward messages so that they eventually reach their destinations. The nodes of such networks are *reactive systems* that cooperate to provide a global service (the sending of messages from node to node) satisfying certain correctness properties (namely 'loop freedom'—that messages are never sent in circles).

We have mechanized an existing formal but pen-and-paper proof of loop freedom of AODV in the interactive theorem prover Isabelle/HOL. While the process algebra model and the fine details of the original proof are quite formal, the structure of the proof is much less so. This necessitated the development of new framework elements and techniques in Isabelle. In particular, we adapted standard theory on inductive assertions to show invariants over individual reactive nodes and introduced machinery for assume/guarantee reasoning to lift these invariants to networks of communicating processes. While the original proof reasoned informally over traces, the mechanized proof is purely based on invariant reasoning, i.e., on reasoning over pairs of reachable states. Our combination of techniques works very well and is likely useful for modelling and verifying similar protocols in an interactive theorem prover.

We are currently finalising a paper describing this work for submission in January.

In collaboration with Peter Hofner (NICTA) and Robert J. van Glabbeek (UNSW/NICTA).

6.4. Hybrid Synchronous Languages

Participants: Timothy Bourke, Jun Inoue, Antoine Madet, Marc Pouzet.

During year 2013, we mainly worked on three directions: (a) the treatment of DAEs; (b) the design and implementation of a causality analysis for hybrid systems modelers; (c) the study of numerical techniques for *non-smooth dynamical systems*.

DAEs As part of our participation in the European project MODRIO and SYS2SOFT projects, we have been developing a prototype for simulating DAE (Differential-Algebraic Equations) systems. DAEs are the basis of the language Modelica and their interaction with discrete features — in particular the novel ones introduced in 2012, like hierarchical automata and clocks — raise difficult semantical and compilation issues. The goal is to precisely define the interaction between synchronous programming constructs and DAEs, in term of semantics and compilation. One strong difficulty at the moment is that existing techniques (index reduction, dymmy derivative) are not modular and force, either to (a)

write an interpretor where index reduction is done dynamically every time a mode change occurs or (b) statically enumerate all the modes, performing index reduction for every of those. While the first technique is too slow in practice (and it is not used in the most advanced Modelica compiler), the second one may explode in practice (putting n two-state automata in parallel lead to 2^n states to be enumerated). During year 2013, we have investigated a new approach for index reduction.

Work to-date has focused on implementing standard algorithms from the literature (notably Pantelides, Dummy Derivatives, Dynamic State Selection). Despite the importance of these algorithms to tools like Modelica, we found that important implementation details and "tricks" are not always well documented.

This work is developed hand-in-hand with the interface to the Sundials IDA solver.

Causality Analysis We have designed a causality analysis for a language that mix stream equations, hierarchical automata and ODEs and implemented it in the Zélus compiler. Its purpose is to give a sufficient condition for a hybrid program can be turned into statically scheduled code. Moreover, the analysis ensures that absence of discontinuities outside of declared zero-crossing events. This result is novel and the proof deeply rely on the use of *non standard analysis* introduced in our previous works. This new result has been accepted for publication at HSCC 2014.

Non Smooth Dynamical Systems In parallel, we collaborate with Bernard Brogliato and Vincent Acary (Inria team BIBOP, Grenoble) on non smooth dynamical systems. Beside general-purpose techniques for solving DAEs and implemented in Modelica compilers, there exist dedicated methods for systems with a lot of discontinuities and contacts (in mechanical system, electrical analogous circuits, etc.). They are far more efficient and numerically accurate than general-purpose techniques when the number of contact is important (e.g., transient in electrical circuits, a bag of marbles). They are based on a time stepping execution and do not have to stop at every zero-crossing event. The combination of those techniques with event detection ones (as used in the Simulink tool) is largely unknown. We are currently inverstigating the extension of our previous work to take Brogliato and Acary techniques into account. This is a novel but promising direction of research for the year to come.

In this research activity, we develop the new language Zélus used as a laboratory for experimenting novel programming constructs and compilation techniques. It serves to illustrate our research as Lucid Synchrone did in the past.

In collaboration with Benoit Caillaud and Albert Benveniste of the Inria HYCOMES team.

6.5. Fidelity in Real-Time Programming

Participants: Timothy Bourke, Guillaume Baudart.

We are close to completing a careful analysis of literature related to the quasi-synchronous model for real-time, distributed systems. We have extended existing results by increasing their precision, providing detailed proofs, and simplifying protocol descriptions. The work to-date is documented in a draft document which we expect will eventually become a technical report or journal article.

Quasi-synchronous architectures, sometimes termed Loosely Time-Triggered Architectures (LTTAs), are ubiquitious in the development of distributed, real-time systems. They represent a broad class of systems whose modelling and programming mixes elements of discrete time, physical time, and a notion of approximation. We expect that addressing these elements—in the Zélus programming language—will lead to insights and advances in a broader ambition to program in physical time.

6.6. A theory of safe optimisations in the C11/C++11 memory model and applications to compiler testing

Participants: Francesco Zappa Nardelli, Robin Morisset.

Compilers sometimes generate correct sequential code but break the concurrency memory model of the programming language: these subtle compiler bugs are observable only when the miscompiled functions interact with concurrent contexts, making them particularly hard to detect. In this work we design a strategy to reduce the hard problem of hunting concurrency compiler bugs to differential testing of sequential code and build a tool that puts this strategy to work. Our first contribution is a theory of sound optimisations in the C11/C++11 memory model, covering most of the optimisations we have observed in real compilers and validating the claim that common compiler optisations are sound in the C11/C++11 memory model. Our second contribution is to show how, building on this theory, concurrency compiler bugs can be identified by comparing the memory trace of compiled code against a reference memory trace for the source code. Our tool identified several mistaken write introductions and other unexpected behaviours in the latest release of the gcc compiler.

A paper on this work has been accepted in [22].

6.7. A verified compiler for relaxed-memory concurrency

Participant: Francesco Zappa Nardelli.

We studied the semantic design and verified compilation of a C-like programming language for concurrent shared-memory computation above x86 multiprocessors. The design of such a language is made surprisingly subtle by several factors: the relaxed-memory behaviour of the hardware, the effects of compiler optimisation on concurrent code, the need to support high-performance concurrent algorithms, and the desire for a reasonably simple programming model. In turn, this complexity makes verified (or verifying) compilation both essential and challenging. This project started in 2010. In 2013 an article, describing the correctness proof of all the phases of our CompCertTSO compiler (including experimental fence eliminations), appeared in the Journal of the ACM [7].

In collaboration with Jaroslav Sevcik (U. Cambridge), Viktor Vafeiadis (MPI-SWS), Suresh Jagannathan (Purdue U.), Peter Sewell (U. Cambridge).

6.8. Language design on top of JavaScript

Participant: Francesco Zappa Nardelli.

This research project aims at improving the design of the JavaScript language. In [23] we present a security infrastructure which allows users and content providers to specify access control policies over subsets of a JavaScript program by leveraging the con- cept of delimited histories with revocation. We implement our proposal in WebKit and evaluate it with three policies on 50 widely used websites with no changes to their JavaScript code and report performance overheads and violations. In [32] 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 and we report on performance and software engineering benefits.

With Gregor Richards and Jan Vitek (Purdue University).

6.9. Tiling for iterated stencils

Participants: Tobias Grosser, Sven Verdoolaege, Albert Cohen.

Time-tiling is necessary for the efficient execution of iterative stencil computations. Classical hyper-rectangular tiles cannot be used due to the combination of backward and forward dependences along space dimensions. Existing techniques trade temporal data reuse for inefficiencies in other areas, such as load imbalance, redundant computations, or increased control flow overhead, therefore making it challenging for use with GPUs.

We proposed a time-tiling method for iterative stencil computations on GPUs. Our method is the first tiling algorithm solving the following constraints simultaneously: it does not involve redundant computations, it favors coalesced global-memory accesses, data reuse in local/shared-memory or cache, avoidance of thread divergence, and concurrency, combining hexagonal tile shapes along the time and one spatial dimension with classical tiling along the other spatial dimensions. Hexagonal tiles expose multi-level parallelism as well as data reuse. Experimental results demonstrate significant performance improvements over existing stencil compilers.

Part of this work also involved our colleagues from the POLYFLOW associate-team at the Indian Institute of Science, Bangalore, India.

6.10. Compilation for scalable on-chip parallelism

Participants: Antoniu Pop, Feng Li, Sven Verdoolaege, Govindarajan Ramaswamy, Albert Cohen.

Task-parallel programming models are getting increasingly popular. Many of them provide expressive mechanisms for inter-task synchronization. For example, OpenMP 4.0 will integrate data-driven execution semantics derived from the StarSs research language. Compared to data-parallel and fork-join models of parallelism, the advanced features being introduced into task-parallel models in turn enable improved scalability through load balancing, memory latency mitigation, mitigation of the pressure on memory bandwidth, and as a side effect, reduced power consumption.

We developed a systematic approach to compile a loop nest into concurrent, dependent tasks. We formulated a partitioning scheme based on the tile-to-tile dependences, represented as affine polyhedra. This scheme ensures at compilation time that tasks belonging to the same class have the same, fully explicit incoming and outgoing dependence patterns. This alleviates the burden of a full-blown dependence resolver to track the readiness of tasks at run time. We evaluated our approach and algorithms in the PPCG compiler, targeting OpenStream, our experimental data-flow task-parallel language with explicit inter-task dependences and a lightweight runtime. Experimental results demonstrate the effectiveness of the approach.

Part of this work also involved our colleagues from the POLYFLOW associate-team at the Indian Institute of Science, Bangalore, India.

6.11. Correct and efficient runtime systems

Participants: Nhat Minh Lê, Robin Morisset, Adrien Guatto, Antoniu Pop, Francesco Zappa Nardelli, Albert Cohen.

User-space scheduling and concurrent first-in first-out queues are two essential building blocks of parallel programming runtimes. They are, however, rarely used together since typical schedulers are oblivious to the ordering constraints introduced by buffered communication.

Chase and Lev's concurrent deque is a key data structure in shared-memory parallel programming and plays an essential role in work-stealing schedulers. We provided the first correctness proof of an optimized implementation of Chase and Lev's deque on top of the POWER and ARM architectures: these provide very relaxed memory models, which we exploit to improve performance but considerably complicate the reasoning. We also studied an optimized x86 and a portable C11 implementation, conducting systematic experiments to evaluate the impact of memory barrier optimizations. Our results demonstrate the benefits of hand tuning the deque code when running on top of relaxed memory models.

Based on this early success, we started working on a more global solution using a new lock-free algorithm for stalling and waking-up tasks in a user-space scheduler according to changes in the state of the corresponding queues. The algorithm is portable and correct, since it is written and proven against the C11 memory model. We showed through experiments that it can serve as a keystone to efficient parallel runtime systems.

These efforts underline the parallelizing compilation research for n-synchronous languages, and the scalable parallel execution of OpenStream.

6.12. Checking Synchronous Compiler Correctness

Participants: Francesco Zappa Nardelli, Guillaume Chelfi, Marc Pouzet.

During year 2013, we have worked on the use of formal verification of compilation steps in the compiler of a Lustre-like synchronous language. Two main directions has been taken:

- The use of SMT-based k-induction techniques to verify the correctness of the successive steps of a synchronous compiler. We used the tool KIND developed by Cesare Tinelli (Iowa state Univ.) and applied it to the Heptagon compiler. The compiler does several source-to-source transformations upto sequential code and KIND was used to verify the equivalence between those successive steps. We came to the conclusion that for most programs, equivalence checking fails unless extra traceability information is added by the compiler.
- The development of a dedicated verification technique to prove the equivalence between a Lustre program and its sequential implementation. We plan to pursue this work during year 2014. Cesare Tinelli will be visiting professor for a month during June 2014.

SPADES Team

6. New Results

6.1. Components and Contracts

Participants: Gregor Goessler, Quentin Sabah, Jean-Bernard Stefani.

6.1.1. Analysis of logical causality

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". Surprisingly, the study of logical causality has so far received little attention in computer science, with the notable exception of [69] and its instantiations. However, this approach relies on a causal model that may not be known, for instance in presence of black-box components.

Improving on previous results, we have proposed in [21] an approach to enhance the fault diagnosis in black-box component-based systems, in which only events on component interfaces are observable. For such systems, we have described a causality analysis framework 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, and applicable to real-time systems. We have illustrated the analysis with a case study from the medical device domain.

In [5] we have proposed a formal framework for reasoning about causality, and blaming system-level failures on the component(s) that caused them. The framework is general in the sense that it applies to many different models of computation and communication (MoC), such as synchronous and asynchronous computation, and communication by messages or shared variables. We are currently instantiating the framework to specific MoC, in particular, to timed automata, and developing a refinement of our original approach that reduces the number of false positives.

6.1.2. Supporting isolation for actors in shared memory

The actor model of concurrency, as supported *e.g.*, by the Erlang programming language, is an appealing programming model for the construction of concurrent and distributed systems, and multicore programming in particular. Although much work has taken place in particular during the past ten years on efficient implementations of the actor model, the design space is far from being completely understood.

As part of Quentin Sabah's thesis [10], we have developed a variant of the actor model that, in contrast to previous works, ensures a strict isolation between actors while imposing no restriction on the form of data exchanged in messages. We have formally specified an abstract machine, called SIAAM (see Sec.5.4.5), for an extension of the Java language with our actor model, and implemented it as a modified Jikes virtual machine, a state of the art Java virtual machine. A combination of points-to and live variable analyses has been implemented using the Soot framework, that can be used to remove unnecessary read and write checks for isolation. A diagnosis tool built on top of the analyses helps programmers to pinpoint potential problems (exceptions raised indicating a potential violation of isolation). We have shown with artificial and small applicative benchmarks that, using our analyses to improve performance, our implementation is reasonably efficient and imposes low overhead for the benefit of strict isolation.

In addition, we have developed a Coq proof of the isolation property enforced by SIAAM, namely that no information between actors can take place outside of message exchanges, despite the presence of a shared heap between actors.

6.2. Real-Time multicore programming

Participants: Vagelis Bebelis, Gwenaël Delaval, Pascal Fradet, Alain Girault, Gregor Goessler, Bertrand Jeannet, Gideon Smeding, 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 [57]. 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 [90]. 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 [39].

Following our past results on the PRET-C programming language [35], 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 [22] (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.

This work has been conducted within the RIPPES associated team.

6.2.2. WCET analysis

Our past work on the WCET analysis of PRET-C programs has led us to design static analyses, for instance to prune unfeasible paths in the control flow graph [36]. In 2013, we have worked on how to take into account direct mapped instruction caches in WCET analysis. Instruction caches are essential to address if one wants to analyze large embedded programs. Our cache analysis technique offers the same precision as the most precise techniques [80], while improving analysis time by up to 240 times. This improvement is achieved by analyzing individual blocks of the control flow graph separately, and by proposing a tailored abstract domain to represent efficiently the cache state [14], [25]. In contrast with previous abstract analysis methods [88], [85], our analysis is able to offer the same precision as the concrete approaches [80].

6.2.3. Tradeoff exploration between reliability, power consumption, and execution time

For autonomous critical real-time embedded systems (*e.g.*, satellites), guaranteeing a very high level of reliability is as important as keeping the power consumption as low as possible. We have designed an off-line ready list scheduling heuristics which, from a given software application graph and a given multiprocessor architecture (homogeneous and fully connected), produces a static multiprocessor schedule that optimizes three criteria: its *length* (crucial for real-time systems), its *reliability* (crucial for dependable systems), and its *power consumption* (crucial for autonomous systems). Our tri-criteria scheduling heuristics, *TSH*, uses the *active replication* of the operations and the data-dependencies to increase the reliability, and uses *dynamic voltage and frequency scaling* to lower the power consumption [37], [38]. TSH implements a ready list scheduling heuristics, and we have formulated a new multi-criteria cost function such that we are able to prove rigorously that the static schedules we generate meet both the reliability constraint and the power consumption constraint [12].

By running TSH on a single problem instance, we are able to provide the Pareto front for this instance in 3D, therefore exposing the user to several tradeoffs between the power consumption, the reliability and the execution time. Thanks to extensive simulation results, we have shown how TSH behaves in practice. Firstly, we have compared TSH versus an optimal Mixed Linear Integer Program on small instances; the experimental results show that TSH behaves very well compared to the ILP. Secondly, we have compared TSH with the ECS heuristic (Energy-Conscious Scheduling [77]); the experimental results show that TSH performs systematically better than ECS.

This is a joint work with Ismail Assayad (U. Casablanca, Morocco) and Hamoudi Kalla (U. Batna, Algeria), who both visit the team regularly.

6.2.4. Modular distribution

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 Xavier Nicollin (Grenoble INP, VERIMAG lab).

We are working on type systems to formalize, in a uniform way, both the clock calculus and the location calculus of a synchronous data-flow programming language (the HEPTAGON language, inspired from LUCID SYNCHRONE [49]). On one hand, the clock calculus infers the clock of each variable in the program and checks the clock consistency: *e.g.*, a time-homogeneous function, like +, should be applied to variables with identical clocks. On the other hand, the location calculus infers the spatial distribution of computations and checks the spatial consistency: *e.g.*, a centralized operator, like +, should be applied to variables located at the same location. Compared to the PhD of Gwenaël Delaval [55], [56], the goal is to achieve *modular* distribution. 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. We make use of our uniform type system to express the computing locations as first-class abstract types, exactly like clocks. It allows us to compile a typed variable (typed by both the clock and the location calculi) into `if ... then ... else ...` structures, whose conditions will be valuations of the clock and location variables.

We currently work on an example of software-defined radio. We have shown on this example how to use a modified clock calculus to describe the localisation of values as clocks, and the architecture as clocks (for the computing resources) and their relations (for communication links).

6.2.5. Distribution of synchronous programs under real-time constraints

The goal of Gideon Smeding's PhD thesis [11] was to propose a quasi-synchronous framework encompassing constraints on the relative speed of clocks, together with a formalism for reasoning about clock-dependent properties within the model. This framework should provide a seamless link between synchronous models and their asynchronous implementation.

The quasi-synchronous approach developed in [11] considers independently clocked, synchronous components that interact via communication-by-sampling or FIFO channels. We have defined relative drift bounds on pairs of recurring events such as clock ticks or the arrival of a message. Drift bounds express constraints on the stability of clocks, *e.g.*, at least two ticks of one per three consecutive ticks of the other. We can thus move from total synchrony, where all clocks tick simultaneously, to global asynchrony by relaxing the drift bounds. As constraints are more relaxed, behavior diverges more and more from synchronous system behavior. In many systems, such as distributed control systems, occasional deviations of input and output signals of the controller from their behavior in the synchronous model may be acceptable as long as the frequency of such deviations is bounded. The approach of [11] takes as inputs a program written in a Lustre-like language extended with asynchronous communication by sampling, application requirements on the distribution in the form of weakly-hard constraints [45] bounding *e.g.*, the tolerated loss of data tokens, and platform assertions (*e.g.*, relative clock speeds, available communication resources), and verifies whether the program meets the requirements under the platform assertions.

6.2.6. 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.

Last year, we have introduced the *schedulable parametric data-flow (SPDF)* MoC for dynamic streaming applications [60]. SPDF extends the standard dataflow model by allowing rates to be parametric. SPDF was designed to be statically analyzable while retaining sufficient expressive power.

Following the same lines, we have recently proposed the *Boolean Parametric Data Flow (BPDF)* MoC which combines integer parameters (to express dynamic rates) and boolean parameters (to express the activation and deactivation of communication channels) [15], [26], [24]. High dynamism 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 which ensure statically the liveness and the boundedness of BDPF graphs. Our case studies are video decoders for high definition video streaming such as VC-1.

We have proposed a generic and flexible framework to generate parallel ASAP schedules targeted to the new STHORM many-core platform designed by STMicroelectronics [29], [23]. The parametric dataflow graph is associated with generic or user-defined specific constraints aimed at minimizing, timing, buffer sizes, power consumption, or other criteria. The scheduling algorithm executes with minimal overhead and can be adapted to different scheduling policies just by changing some constraints. The safety of both the dataflow graph and constraints can be checked statically and all schedules are guaranteed to be bounded and deadlock free. This parallel scheduling framework has been developed for a parametric MoC without booleans. We are now focusing on extending it to BPDF applications.

This research is the central topic of Vagelis Bebelis' PhD thesis. It is conducted in collaboration with STMicroelectronics.

6.2.7. Abstract Acceleration of general linear loops

We have investigated abstract acceleration techniques for computing loop invariants for numerical programs with linear assignments and conditionals. Whereas abstract interpretation techniques typically over-approximate the set of reachable states iteratively, abstract acceleration captures the effect of the loop with a single, non-iterative transfer function applied to the initial states at the loop head.

In contrast to previous acceleration techniques, our approach applies to any linear loop without restrictions. Its novelty lies in the use of the Jordan normal form decomposition of the loop body to derive symbolic expressions for the entries of the matrix modeling the effect of $n >= 0$ iterations of the loop. The entries of such a matrix depend on n through complex polynomial, exponential and trigonometric functions. Therefore, we introduced an abstract domain for matrices that captures the linear inequality relations between these complex expressions. This results in an abstract matrix for describing the fixpoint semantics of the loop. We also developed a technique to take into account the guard of the loop by bounding the number of loop iterations, which relies again on the Jordan normal form decomposition.

Our approach integrates smoothly into standard abstract interpreters and can handle programs with nested loops and loops containing conditional branches. We evaluate it over small but complex loops that are commonly found in control software, comparing it with other tools for computing linear loop invariants. The loops in our benchmarks typically exhibit polynomial, exponential and oscillatory behaviors that present challenges to existing approaches, that are either too unprecise (classical abstract interpretation) or limited to a restricted class of loops (*e.g.*, translation with resets in the case of abstract acceleration, or stable loops, in the sense of control theory, for ellipsoid methods). Our approach finds non-trivial invariants to prove useful bounds on the values of variables for such loops, clearly outperforming the existing approaches in terms of precision while exhibiting good performance.

A paper presenting this technique has been accepted to POPL'2014. An extended version has been published in arXiv [30].

6.2.8. 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.*, [87] 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 [82]. 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 [61].

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.

In [52], we have proposed a technique for the synthesis of safety controllers for switched systems using multi-scale abstractions that allow us to deal with fast switching while keeping the number of states in the abstraction at a reasonable level. The finest scales of the abstraction are effectively explored only when fast switching is needed, that is when the system approaches the unsafe set. We have implemented these results in the tool COSYMA (COntroller SYnthesis using Multi-scale Abstractions, see Sec. 5.4.2) [20]. The tool accepts a description of a switched system represented by a set of differential equations and the sampling parameters used to define an approximation of the state-space on which discrete abstractions are computed. The tool generates a controller — if it exists — for the system that enforces a given safety or time-bounded reachability specification.

In [19], we have presented an approach using mode sequences of given length as symbolic states for our abstractions. We have shown that the resulting symbolic models are approximately bisimilar to the original switched system and that an arbitrary precision can be achieved by considering sufficiently long mode sequences. The advantage of this approach over existing ones is double: first, the transition relation of the symbolic model admits a very compact representation under the form of a shift operator; second, our approach does not use lattices over the state-space and can potentially be used for higher dimensional systems. We have provided a theoretical comparison with the lattice-based approach and presented a simple criterion enabling to choose the most appropriate approach for a given switched system. We have applied the approach to a model of road traffic for which we have synthesized a schedule for the coordination of traffic lights under constraints of safety and fairness.

6.3. Language Based Fault-Tolerance

Participants: Dmitry Burlyaev, Pascal Fradet, Alain Girault, Jean-Bernard Stefani.

6.3.1. Automatic Transformations for Fault tolerant Circuits

In the recent 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 have designed a simple hardware description language inspired from LUSTRE and Lucid Synchrone. It is a core functional language manipulating synchronous boolean streams. We consider both single-event upsets (SEU) and single-event transients (SET) and all fault models of the form "at most 1 SEU or SET within n clock signals". The language's semantics as well as fault modes have been formalized in Coq and many basic (library) properties have been shown on that language.

We have expressed several variants of triple modular redundancy (TMR) as program transformations. We have proposed a verification-based approach to minimize the number of voters in TMR [16]. Our technique guarantees that the resulting circuit *(i)* is fault tolerant to the soft-errors defined by the fault model and *(ii)* is functionally equivalent to the initial one. Our approach operates at the logic level and takes into account the input and output interface specifications of the circuit. Its implementation makes use of graph traversal algorithms, fixed-point iterations, and BDDs. Experimental results on the ITC'99 benchmark suite indicate that our method significantly decreases the number of inserted voters which entails a hardware reduction of up to 55% and a clock frequency increase of up to 35% compared to full TMR. We address scalability issues arising from formal verification with approximations and assess their efficiency and precision.

We are currently studying the definition of other fault-tolerant techniques (*e.g.*, time redundancy, mixed time/space redundancy) as program transformations. We are also considering the use of the Coq proof assistant to certify that the transformations make the programs fault tolerant *w.r.t.* specific fault models. Our long term goal is to design an aspect-like language allowing users to specify and tune a wide range of fault tolerance techniques, while ensuring that the corresponding transformations ensure well-defined fault-tolerance properties. The advantage would be to produce fault-tolerant circuits by specifying fault-tolerant properties/strategies separately from their functional specifications.

6.3.2. Concurrent flexible reversibility

In the recent years, we have been investigating reversible concurrent computation, and investigated various reversible concurrent programming models, with the hope that reversibility can shed some light on the common semantic features underlying various forms of fault recovery techniques (including, exceptions, transactions, and checkpoint/rollback schemes).

As part of this research program, we have devised a reversible variant of the higher-order π-calculus, equipped with an imperative rollback operation that allows a concurrent program to be rolled back to a past execution state, and a primitive form of compensation to control (forward execution) after a rollback operation [18]. We have shown that these two extensions provide very powerful primitives for programming different forms of rollback/compensation schemes. We have shown in particular that they are powerful enough to provide a faithful encoding of a notion of communicating transaction proposed in the literature. We have started the development of a behavioral theory for this $\mathsf{croll}\pi$ calculus, and proved in particular a context lemma, similar to that of the π-calculus, although the reversible machinery makes its proof more involved.

This work was done in collaboration with Inria teams FOCUS in Bologna and CELTIQUE in Rennes, and as part of the ANR REVER project.

FORMES Team

6. New Results

6.1. Type and rewriting theory

Participants: Frédéric Blanqui, Jean-Pierre Jouannaud, Jianqi Li, Qian Wang.

Qian Wang and Bruno Barras have proved the strong normalization property of CoqMTU in presence of strong elimination, a major step towars the full certification of CoqMTU [16].

Jouannaud and Li have developped a new framework, Normal Abstract Rewriting Systems (NARS), that captures all known Church-Rosser results in presence of a termination assumption allowing to reduce equality of terms to a simpler equality on their normal forms. This result applies to the paticular case of higher-order rewriting for which it solved long-standing open problems [10].

Jouannaud and Liu have continued their investigation of Church-Rosser properties of non-terminating rewrite systems [10], showing recently first, that many results found in the litterature could be captured, and generalized, by using van Oostrom's decreasing diagram technique (accepted at Symposium on Algebraic Specifications, Kanazawa, Japan, April 2014). The next step, which has been recently completed, is a powerful result generalizing Knuth and Bendix confluence test to non terminating rewrite system (submitted).

Frédéric Blanqui, Jean-Pierre Jouannaud and Albert Rubio (Technical University of Catalonia) have developed a method aiming at carrying out termination proof for higher-order calculi. CPO appears to be the ultimate improvement of the higher-order recursive path ordering (HORPO) [25] in the sense that this definition captures the essence of computability arguments *à la* Tait and Girard, therefore explaining the name of the improved ordering. It has been shown that CPO allows to consider higher-order rewrite rules in a simple type discipline with inductive types, that most of the guards present in the recursive calls of its core definition cannot be relaxed in any natural way without losing well-foundedness, and that the precedence on function symbols cannot be made more liberal anymore. This result is submitted to journal, and has been concurrently generalized to higher-order calculi with dependent types by Jouannaud and Li (submitted).

Frédéric Blanqui worked on the formalization in the Coq proof assistant of various definitions of the notion of α-equivalence on pure λ-terms. In particular, he formalized and formally proved equivalent the definitions of Church (1932), Curry and Feys (1958), Krivine (1993), and Gabbay and Pitts (1999). This work is freely available from the CoLoR library released on December 13th.

Frédéric Blanqui worked with John Steinberger (Tsinghua University) on the formal verification in Coq of proofs of theorems on coset arrays and non-negative integer linear combinations.

6.2. Automated theorem proving

Participant: Kim-Quyen Ly.

Kim-Quyen Ly extended her formally-proved (in Coq) automated termination-certificate (for first-order rewrite systems) verifier Rainbow for dealing with certificates using arguments filtering [22] and other termination techniques.

6.3. Simulation

Participants: Vania Joloboff, Antoine Rouquette, Shenpeng Wang.

There exists very fast Loosely Timed simulators such as **SimSoC** that can run the application software to validate its functionality and possibly test real time software using timers. But such simulators do not provide good enough timings to evaluate the software performance. The idea of "Approximately Timed" simulation is to provide a fast simulation that can be used by software developers, and yet provide performance estimate. The goal of approximately timed simulation is to provide estimates that are within a small margin error from the real hardware performance, but at a simulation speed that is an order of magnitude faster than a cycle accurate one.

Modern processors have complex architectures. They can execute a certain number of instructions per clock cycle. There are however several cases where the instruction flow is disrupted, introducing delays in the computation. In order to make an Approximately Timed simulator, our idea is to simulate enough of the processes causing the delays, not simulating the exact hardware processes of the caches and pipe line and I/Os, but using a model with wich the delays can be computed with a reasonable approximation while maintaining fast simulation. Delays may also be related to bus arbitration and interconnect access. These delays are beyond the scope of our work, but can be captured by TLM (timed) transactions. In our work, we are considering only the processor model and we rely upon TLM interface to the interconnect for peripheral access to provides us with timing delays.

We have started to investigate a new approach to provide a fast Approximately Timed ISS, that does not simulate fully the hardware, yet provides good precision estimates, and does not use stastistical methods. Our approach consists in developing a higher abstraction model of the processor (than the CA models) that still executes instructions using fast SystemC/TLM code, but in parallel maintains some architecture state to measure the delays introduces by cache misses and pipe line stalls, although the pipe line is not really simulated.

6.4. Certification of a Simulator

Participants: Vania Joloboff, Jean-François Monin, Xiaomu Shi.

We have developed a correctness proof of a part of the hardware simulator **SimSoC**. This is not only an attempt to certify a simulator, but also a new experiment on the certification of non-trivial programs written in C. We have provided a formalized representation of the ARM instruction set and addressing modes in Coq. We also constructed a Coq representation of the ARM simulator in C, using the abstract syntax defined in **CompCert**.

From these two Coq representations, we have developed Coq proofs to prove the correctness of the C code, using the operational semantics of C provided by **CompCert**.

During this work, we have also improved the technology available in Coq for performing *inversions*, a kind of proof steps which heavily occurs in this context.

All of this work has been described in Xiaomu SHI PhD thesis dissertation, presented at University of Grenoble in July 2013, and at ITP 2013 conference[15].

SECSI Project-Team

6. New Results

6.1. Dishonest keys (Objective 2)

Participants: Hubert Comon-Lundh, Guillaume Scerri.

One of the main issues in the formal verification of the security protocols is the validity (and scope) of the formal model. Otherwise, it may happen that a protocol is proved and later someone finds an attack. This paradoxical situation may happen when the formal model used in the proof is too abstract.

A main stream of research therefore consists in proving full abstraction results (also called *soundness*): if the protocol is secure in the (symbolic) model, then an attack can only occur with negligible probability in a computational model. Such results have two main drawbacks: first they are very complicated, and have to be completed again and again for each combination of security primitives. Second, they require strong hypotheses on the primitives, some of which are not realistic. For instance, it is assumed that the attacker cannot forge his own keys (or that all keys come with their certificates, even for symmetric encryption keys).

Hubert Comon-Lundh, Véronique Cortier and Guillaume Scerri had proposed an extension of the symbolic model in 2012, and proved it computationally sound, without this restriction on the dishonest keys.

6.2. Deciding trace equivalence

Participants: David Baelde, Stéphanie Delaune, Rémy Chrétien, Lucca Hirschi.

Most existing results focus on trace properties like secrecy or authentication. There are however several security properties, which cannot be defined (or cannot be naturally defined) as trace properties and require the notion of indistinguishably. Typical examples are anonymity, privacy related properties or statements closer to security properties used in cryptography.

In the framework of the applied pi-calculus as in similar languages based on equational logics, indistinguishably corresponds to a relation called trace equivalence. Roughly, two processes are trace equivalent when an observer cannot see any difference between the two processes. Static equivalence applies only to observations on finite sets of messages, and do not take into account the dynamic behavior of a process whereas the notion of trace equivalence is more general and takes into account this aspect.

6.2.1. Static equivalence.

As explained above, static equivalence is a cornerstone to provide decision procedures for observational equivalence.

Stéphanie Delaune, in collaboration with Mathieu Baudet and Véronique Cortier, has designed a generic procedure for deducibility and static equivalence that takes as input any convergent rewrite system [12]. They have shown that their algorithm covers most of the existing decision procedures for convergent theories. They also provide an efficient implementation. This paper is a journal version of the work presented at RTA'09.

6.2.2. Trace equivalence.

When the processes under study do not contain replication, trace equivalence can be reduced to the problem of deciding symbolic equivalence [13]. Thanks to this reduction and relying on a result first proved by M. Baudet, this yields the first decidability result of observational equivalence for a general class of equational theories (for processes without else branches and without replication). Moreover, based on another decidability result for deciding equivalence between sets of constraint systems, we get decidability of trace equivalence for processes with else branch for standard primitives.

Even though there are some implementations of the procedures described above, this does not suffice to obtain practical tools. Current prototypes suffer from a classical combinatorial explosion problem caused by the exploration of many interleavings in the behaviour of processes. David Baelde, Stéphanie Delaune, and Lucca Hirschi revisit a work due to Mödersheim et al., generalize it and adapt it for equivalence checking. They obtain an optimization in the form of a reduced symbolic semantics that eliminates redundant interleavings on the fly. This work will be published as:

- D. Baelde, S. Delaune, and L. Hirschi. A Reduced Semantics for Deciding Trace Equivalence using Constraint Systems. In *Proc. 3rd Conference on Principles of Security and Trust (POST 2014)*, Grenoble, April 2014, France.

When processes under study contain replication, the approach relying on symbolic equivalence does not work anymore. Moreover, since it is well-known that deciding reachability properties is undecidable under various restrictions, there is actually no hope to do better for equivalence-based properties. Rémy Chrétien, Véronique Cortier, and Stéphanie Delaune provide the first results of (un)decidability for certain classes of protocols for the equivalence problem. They consider a class of protocols shown to be decidable for reachability properties, and establish a first undecidability result. Then, they restrained the class of protocols a step further by making the protocols deterministic in some sense and preventing it from disclosing secret keys. This tighter class of protocols was then shown to be decidable after reduction to an equivalence between deterministic pushdown automata. This work has been published at ICALP'13 [14].

To deal with replication, another approach has been studied by Vincent Cheval in collaboration with Bruno Blanchet. They propose an extension of the automatic protocol verifier ProVerif. ProVerif can prove observational equivalence between processes that have the same structure but differ by the messages they contain. In order to extend the class of equivalences that ProVerif handles, they extend the language of terms by defining more functions (destructors) by rewrite rules. These extensions have been implemented in ProVerif and allow one to automatically prove anonymity in the private authentication protocol by Abadi and Fournet. This work is part of Vincent Cheval's PhD thesis, and was published as:

- V. Cheval, B. Blanchet. Proving More Observational Equivalences with ProVerif. In *2nd Conference on Principles of Security and Trust (POST 2013)*. David Basin, John Mitchell, eds. Springer Verlag, Lecture Notes in Computer Science 7796, 2013.

6.3. Mobile ad-hoc networks

Participants: Rémy Chrétien, Stéphanie Delaune.

Mobile ad hoc networks consist of mobile wireless devices which autonomously organize their communication infrastructure: each node provides the function of a router and relays packets on paths to other nodes. Finding these paths in an a priori unknown and constantly changing network topology is a crucial functionality of any ad hoc network. Specific protocols, called *routing protocols*, are designed to ensure this functionality known as *route discovery*. Secured versions of routing protocols have been proposed to provide more guarantees on the resulting routes, and some of them have been designed to protect the privacy of the users.

Rémy Chrétien and Stéphanie Delaune propose a framework for analysing privacy-type properties for routing protocols. They use the notion of equivalence between traces to formalise three security properties related to privacy, namely indistinguishability, unlinkability, and anonymity. They study the relationship between these definitions and we illustrate them using two versions of the ANODR routing protocol. This work was published as:

- R. Chrétien, S. Delaune. Formal Analysis of Privacy for Routing Protocols in Mobile Ad Hoc Networks. *Principles of Security and Trust - Second International Conference, POST 2013*, held as Part of the *European Joint Conferences on Theory and Practice of Software, ETAPS 2013*, Rome, Italy, March 16-24, 2013. Proceedings. Springer 2013. Lecture Notes in Computer Science. ISBN 978-3-642-36829-5. Pages 1-20.

6.4. Composition results

Participant: Stéphanie Delaune.

Formal methods have proved their usefulness for analysing the security of protocols. However, protocols are often analysed in isolation, and this is well-known to be not sufficient as soon as the protocols share some keys.

Stéphanie Delaune, in collaboration with Céline Chevalier, Steve Kremer, and Mark Ryan, study whether password protocols can be safely composed, even when a same password is reused. More precisely, they present a transformation which maps a password protocol that is secure for a single protocol session (a decidable problem) to a protocol that is secure for an unbounded number of sessions. Their result provides an effective strategy to design secure password protocols: (i) design a protocol intended to be secure for one protocol session; (ii) apply their transformation and obtain a protocol which is secure for an unbounded number of sessions. Their technique also applies to compose different password protocols allowing one to obtain both inter-protocol and inter-session composition. This work was published as:

- C. Chevalier, S. Delaune, S. Kremer and M. Ryan. Composition of Password-based Protocols. *Formal Methods in System Design* 43(3), pages 369-413, 2013.

6.5. Unconditional Soundness (Objective 2)

Participants: Hubert Comon-Lundh, Guillaume Scerri.

Hubert Comon-Lundh, Véronique Cortier and Guillaume Scerri had shown in a 2012 CCS paper how one could drop one of the assumptions of computational soundness results. However, the proofs remain very complicated and there are still assumptions such as the absence of key cycles, or no dynamic corruption... that are still necessary for all these results.

Gergei Bana and Hubert Comon-Lundh investigated a completely different approach to formal security proofs in a 2012 POST paper, which does not make any such assumptions. The idea can be stated in a nutshell: whereas all existing formal models state the attacker's abilities, they propose to formally state what the attacker *cannot* do.

This makes a big difference, since the soundness need only to be proved formula by formula and only the very necessary assumptions are used for such formulas (for instance, no absence of key cycles is needed). This does not need to be proved again when a primitive is added.

Once the general setting is fixed, the question was how practical is the method. We studied the complexity of the consistency proofs in this setting and showed that we can complete such proofs in Polynomial Time for a wide class of axioms in

- H. Comon-Lundh, V. Cortier and G. Scerri. Tractable inference systems: an extension with a deducibility predicate. In CADE'13, LNAI 7898, pages 91-108. Springer, 2013

The development of a prototype implementation is under development. We expect to complete experiments on a number of protocols.

6.6. Static Analysis of Programs with Imprecise Probabilities

Participant: Jean Goubault-Larrecq [correspondant].

Static analyses allows one to obtain guarantees about the behavior of programs, without running them. Programs that handle numerical data such as feedback control loops pose a challenge in this area. This gets even harder when one considers programs that read numerical data from sensors, and write to actuators, as these data are imprecise, and are governed by probability distributions that may themselves be unknown, and only know to fall into some interval of distributions.

As part of the ANR projet blanc CPP, an efficient static analysis framework that deals with this kind of programs was proposed in 2011 by J. Goubault-Larrecq, O. Bouissou, E. Goubault, Sylvie Putot, based on P-boxes and Dempster-Shafer structures to handle imprecise probabilities.

The semantic foundations were made clearer, a new, improved algorithm was proposed, and new applications were examined in:

- A. Adjé, O. Bouissou, J. Goubault-Larrecq, E. Goubault and S. Putot. Static Analysis of Programs with Imprecise Probabilistic Inputs. In *VSTTE'13*, LNCS. Springer, 2013.

ABSTRACTION Project-Team

6. New Results

6.1. Analysis of Biological Pathways

We have improved our framework to design and analyze biological networks in KAPPA. This framework focuses on protein-protein interaction networks described as graph rewriting systems. Such networks can be used to model some signaling pathways that control the cell cycle. The task is made difficult due to the combinatorial blow up in the number of reachable species (*i.e.*, non-isomorphic connected components of proteins).

6.1.1. Semantics

Participants: Jonathan Hayman, Tobias Heindel [CEA-List].

Keywords: Graph rewriting, Single Push-Out semantics.

Domain-specific rule-based languages can be understood intuitively as transforming graph-like structures, but due to their expressivity these are difficult to model in 'traditional' graph rewriting frameworks.

In [16], we introduce pattern graphs and closed morphisms as a more abstract graph-like model and show how Kappa can be encoded in them by connecting its single-pushout semantics to that for Kappa. This level of abstraction elucidates the earlier single-pushout result for Kappa, teasing apart the proof and guiding the way to richer languages, for example the introduction of compartments within cells.

6.1.2. Causality Analysis

We use causal analysis so as to extract minimal concurrent scenarios that lead to the activation of given events.

6.1.2.1. Implementation

Participant: Jérôme Feret.

Keywords: Causality, Counter-examples, Compression.

This year, we have re-implemented in OPENKAPPA the strong compression method that is described in [48]. The new implementation is very efficient, it has been used to extract minimal scenarios from traces of several hundred of thousands causally related events, that were generated during the simulation of a model of the WnT signaling pathway.

6.1.2.2. Framework

Participant: Jonathan Hayman.

Keywords: Abstraction, Causality, Compression.

Standard notions of independence of rule applications fail to provide adequately concise causal histories, leading to the earlier formulation of strong and weak forms of trajectory compression for Kappa. In [15], we give a simple categorical account of how forms of compression can be uniformly obtained. This generalisation also describes a way for the user to specify their own levels of compression between weak and strong, which we call filtered compression. This is based on the idea of the user specifying the part of the type graph that represents the the structure which the compression technique should track through the trace.

6.1.3. Model Reduction

Participants: Ferdinanda Camporesi, Jérôme Feret, Jonathan Hayman.

Keywords: Context-sensitivity, Differential semantics, Model reduction.

Rule-based modeling allows very compact descriptions of protein-protein interaction networks. However, combinatorial complexity increases again when one attempts to describe formally the behaviour of the networks, which motivates the use of abstractions to make these models more coarse-grained. Context-insensitive abstractions of the intrinsic flow of information among the sites of chemical complexes through the rules have been proposed to infer sound coarse-graining, providing an efficient way to find macro-variables and the corresponding reduced models.

In [12], we propose a framework to allow the tuning of the context-sensitivity of the information flow analyses and show how these finer analyses can be used to find fewer macro-variables and smaller reduced differential models.

6.2. Andromeda: Accurate and Scalable Security Analysis of Web Applications

Participants: Omer Tripp [Tel Aviv University, Israël], Marco Pistola [University of Washington, Seattle, USA], Patrick Cousot, Radhia Cousot, Salvatore Guarnieri.

Keywords: Abstract interpretation, Security, Web.

Security auditing of industry-scale software systems mandates automation. Static taint analysis enables deep and exhaustive tracking of suspicious data flows for detection of potential leakage and integrity violations, such as cross-site scripting (XSS), SQL injection (SQLi) and log forging. Research in this area has taken two directions: program slicing and type systems. Both of these approaches suffer from a high rate of false findings, which limits the usability of analysis tools based on these techniques. Attempts to reduce the number of false findings have resulted in analyses that are either (i) unsound, suffering from the dual problem of false negatives, or (ii) too expensive due to their high precision, thereby failing to scale to real-world applications.

In [21], we investigate a novel approach for enabling precise yet scalable static taint analysis. The key observation informing our approach is that taint analysis is a demand-driven problem, which enables lazy computation of vulnerable information flows, instead of eagerly computing a complete data-flow solution, which is the reason for the traditional dichotomy between scalability and precision. We have implemented our approach in Andromeda, an analysis tool that computes data-flow propagations on demand, in an efficient and accurate manner, and additionally features incremental analysis capabilities. Andromeda is currently in use in a commercial product. It supports applications written in Java, .NET and JavaScript. Our extensive evaluation of Andromeda on a suite of 16 production-level benchmarks shows Andromeda to achieve high accuracy and compare favorably to a state-of-the-art tool that trades soundness for precision.

6.3. Backward analysis

6.3.1. Automatic Inference of Necessary Preconditions

Participants: Patrick Cousot, Radhia Cousot, Manuel Fähndrich [Microsoft Research, Redmond, USA], Francesco Logozzo [Microsoft Research, Redmond, USA].

Keywords: Abstract interpretation, Backward analysis, Static analysis, Necessary condition inference.

In [14], we consider the problem of automatic precondition inference for: (i) program verification; (ii) helping the annotation process of legacy code; and (iii) helping generating code contracts during code refactoring. We argue that the common notion of sufficient precondition inference (i.e., under which precondition is the program correct?) imposes too large a burden on call-sites, and hence is unfit for automatic program analysis. Therefore, we define the problem of necessary precondition inference (i.e., under which precondition, if violated, will the program always be incorrect?). We designed and implemented several new abstract interpretation-based analyses to infer necessary preconditions. The analyses infer atomic preconditions (including disjunctions), as well as universally and existentially quantified preconditions.

We experimentally validated the analyses on large scale industrial code.

For unannotated code, the inference algorithms find necessary preconditions for almost 64% of methods which contained warnings. In 27% of these cases the inferred preconditions were also sufficient, meaning all warnings within the method body disappeared. For annotated code, the inference algorithms find necessary preconditions for over 68% of methods with warnings. In almost 50% of these cases the preconditions were also sufficient. Overall, the precision improvement obtained by precondition inference (counted as the additional number of methods with no warnings) ranged between 9% and 21%.

6.3.2. Under-approximations to infer sufficient program conditions

Participant: Antoine Miné.

Keywords: Abstract interpretation, Backward analysis, Numerical abstract domains, Static analysis, Sufficient condition inference, Under-approximations.

In [9] we discuss the automatic inference of sufficient preconditions by abstract interpretation and sketch the construction of an under-approximating backward analysis. We focus on numeric properties of variables and revisit three classic numeric abstract domains: intervals, octagons, and polyhedra, with new under-approximating backward transfer functions, including the support for non-deterministic expressions, as well as lower widenings to handle loops. We show that effective under-approximation is possible natively in these domains without necessarily resorting to disjunctive completion nor domain complementation. Applications include the derivation of sufficient conditions for a program to never step outside an envelope of safe states, or dually to force it to eventually fail. We built a proof-of-concept prototype implementation based on the APRON numeric domain library and experimented it on simple examples (the prototype is available for download and usable on-line at http://www.di.ens.fr/~mine/banal).

6.4. Bisimulation metrics

6.4.1. Bisimulation for MDP through Families of Functional Expressions

Participants: Norman Ferns, Sophia Knight [LIX], Doina Precup [McGill University].

Keywords: Markov decision processes, Bisimulation, Metrics.

We have transfered a notion of quantitative bisimilarity for labelled Markov processes [54] to Markov decision processes with continuous state spaces. This notion takes the form of a pseudometric on the system states, cast in terms of the equivalence of a family of functional expressions evaluated on those states and interpreted as a real-valued modal logic. Our proof amounts to a slight modification of previous techniques [61], [60] used to prove equivalence with a fixed-point pseudometric on the state-space of a labelled Markov process and making heavy use of the Kantorovich probability metric. Indeed, we again demonstrate equivalence with a fixed-point pseudometric defined on Markov decision processes [57]; what is novel is that we recast this proof in terms of integral probability metrics [59] defined through the family of functional expressions, shifting emphasis back to properties of such families. The hope is that a judicious choice of family might lead to something more computationally tractable than bisimilarity whilst maintaining its pleasing theoretical guarantees. Moreover, we use a trick from descriptive set theory to extend our results to MDPs with bounded measurable reward functions, dropping a previous continuity constraint on rewards and Markov kernels.

This work is under submission.

6.4.2. Bisimulation Metrics are Optimal Value Functions

Participants: Norman Ferns, Doina Precup [McGill University].

Keywords: Markov decision processes, Bisimulation, Metrics.

We have proved that a behavioural pseudometric defined on the state space of a given Markov decision process and whose kernel is stochastic bisimilarity [57] can be expressed as the optimal value function of another Markov decision process. Furthermore, this latter process can be interpreted as an optimal coupling of two copies of the original model.

This work is under submission.

6.5. A Constraint Solver Based on Abstract Domains

Participants: Marie Pelleau [University of Nantes, LINA], Antoine Miné, Charlotte Truchet [University of Nantes, LINA], Frédéric Benhamou [University of Nantes, LINA].

Keywords: Abstract interpretation, Backward analysis, Numerical abstract domains, Static analysis, Sufficient condition inference, Under-approximations.

In [18] and [19] we apply techniques from abstract interpretation to constraint programming (which aims at solving hard combinatorial problems with a generic framework based on first-order logics). We highlight some links and differences between these fields: both compute fixpoints by iterations but employ different extrapolation and refinement strategies; moreover, consistencies in constraint programming can be mapped to non-relational abstract domains. We then use these correspondences to build an abstract constraint solver that leverages abstract interpretation techniques (such as relational domains) to go beyond classic solvers. We present encouraging experimental results obtained with our prototype implementation.

6.6. A Galois Connection Calculus for Abstract Interpretation

Participants: Patrick Cousot, Radhia Cousot.

Keywords: Abstract interpretation, Galois connection.

In [10], we introduce a Galois connection calculus for language independent specification of abstract interpretations used in programming language semantics, formal verification, and static analysis. This Galois connection calculus and its type system are typed by abstract interpretation.

6.7. Mechanically Verifying a Shape Analysis

Participant: Arnaud Spiwack.

Keywords: Program verification, Abstract interpretation, Static analysis, Shape analysis, Coq.

The result of a static analysis is only as good as the trust put into its correctness. For critical software, the standards are very high, and trusting a complex tool requires costly inspection of its implementation. Mechanically proving the correctness of static analysers is a way to lower these costs: the exigence of trust is moved from various complex dedicated tools to a single simpler general purpose one.

In this context, Arnaud Spiwack worked on an ongoing Coq implementation and certification of a shape abstract domain. The implementation, named Cosa, is based on Evan Chang and Xavier Rival's Xisa. It targets an intermediary language of Xavier Leroy's Compcert C, and interfaces with the domains of the Verasco project.

The development of Cosa lead Arnaud Spiwack to express the abstract interpretation correctness property in term of refinement calculus, which allowed to use interaction structures (a type theoretic variant of the refinement calculus) as a central structuring element of Cosa. Arnaud Spiwack started investigating how the technology of nominal sets could be leveraged to prove the correctness of unfolding (which involves choosing new names) in Cosa.

6.8. Modular Construction of Shape-Numeric Analyzers

Participants: Bor-Yuh Evan Chang [University of Colorado, Boulder, USA], Xavier Rival.

Keywords: Abstract interpretation, Memory abstraction, Shape abstract domains.

In [13], we discuss the modular construction of memory abstract domains.

The aim of static analysis is to infer invariants about programs that are precise enough to establish semantic properties, such as the absence of run-time errors. Broadly speaking, there are two major branches of static analysis for imperative programs. Pointer and *shape* analyses focus on inferring properties of pointers, dynamically-allocated memory, and recursive data structures, while *numeric* analyses seek to derive invariants on numeric values. Although simultaneous inference of shape-numeric invariants is often needed, this case is especially challenging and is not particularly well explored. Notably, simultaneous shape-numeric inference raises complex issues in the design of the static analyzer itself.

In this paper, we study the construction of such shape-numeric, static analyzers. We set up an abstract interpretation framework that allows us to reason about simultaneous shape-numeric properties by combining shape and numeric abstractions into a modular, expressive abstract domain. Such a modular structure is highly desirable to make its formalization and implementation easier to do and get correct. To achieve this, we choose a concrete semantics that can be abstracted step-by-step, while preserving a high level of expressiveness. The structure of abstract operations (i.e., transfer, join, and comparison) follows the structure of this semantics. The advantage of this construction is to divide the analyzer in modules and functors that implement abstractions of distinct features.

6.9. Reduced Product Combination of Abstract Domains for Shapes

Participants: Bor-Yuh Evan Chang [University of Colorado, Boulder, USA], Xavier Rival, Antoine Toubhans.

Keywords: Abstract interpretation, Memory abstraction, Shape abstract domains.

In [20], we discuss the construction of shape abstract domains by reduced product.

Real-world data structures are often enhanced with additional pointers capturing alternative paths through a basic inductive skeleton (e.g., back pointers, head pointers). From the static analysis point of view, we must obtain several interlocking shape invariants. At the same time, it is well understood in abstract interpretation design that supporting a separation of concerns is critically important to designing powerful static analyses. Such a separation of concerns is often obtained via a reduced product on a case-by-case basis. In this paper, we lift this idea to abstract domains for shape analyses, introducing a domain combination operator for memory abstractions. As an example, we present *simultaneous separating shape graphs*, a product construction that combines instances of separation logic-based shape domains. The key enabler for this construction is a static analysis on inductive data structure definitions to derive relations between the skeleton and the alternative paths. From the engineering standpoint, this construction allows each component to reason independently about different aspects of the data structure invariant and then separately exchange information via a reduction operator. From the usability standpoint, we enable describing a data structure invariant in terms of several inductive definitions that hold simultaneously.

6.10. Relational Thread-Modular Static Value Analysis

Participant: Antoine Miné.

Keywords: Abstract interpretation, Concurrency, Embedded software, Rely-guarantee methods, Run-time errors, Safety.

We study in [17] thread-modular static analysis by abstract interpretation to infer the values of variables in concurrent programs. We show how to go beyond the state of the art and increase an analysis precision by adding the ability to infer some relational and history-sensitive properties of thread interferences. The fundamental basis of this work is the formalization by abstract interpretation of a rely-guarantee concrete semantics which is thread-modular, constructive, and complete for safety properties. We then show that previous analyses based on non-relational interferences can be retrieved as coarse computable abstractions of this semantics; additionally, we present novel abstraction examples exploiting our ability to reason more precisely about interferences, including domains to infer relational lock invariants and the monotonicity of counters. Our method and domains have been implemented in the ASTRÉEA static analyzer (5.3) that checks for run-time errors in embedded concurrent C programs, where they enabled a significant reduction of the number of false alarms.

6.11. Static Analyzers on the Cloud

Participants: Michael Barnett [Microsoft Research, Redmond, USA], Mehdi Bouaziz, Francesco Logozzo [Microsoft Research, Redmond, USA], Manuel Fähndrich [Microsoft Research, Redmond, USA].

A cloud-based static analyzer runs as service. Clients issue analysis requests through the local network or over the internet. The analysis takes advantage of the large computation resources offered by the cloud: the underlying infrastructure ensures scaling and unlimited storage. Cloud-based analyzers may relax performance-precision trade-offs usually associated with desktop-based analyzers. More cores enable more precise and responsive analyses. More storage enables perfect caching of the analysis results, shareable among different clients, and queryable off-line. To realize these advantages, cloud-based analyzers need to be architected differently than desktop ones. In [11], we describe our ongoing effort of moving a desktop analyzer, Clousot, into a cloud-based one, Cloudot.

6.12. Termination

We have explored the analysis of program termination and the inference of sufficient conditions to ensure the definite termination of programs using abstract interpretation techniques. Following [40], we employ a backward analysis over an abstract domain of ranking functions.

6.12.1. Abstract Domain of Segmented Ranking Functions

Participant: Caterina Urban.

We present in [24] and [23] a parameterized abstract domain that infers sufficient conditions for program termination by automatically synthesizing piecewise-defined ranking functions over natural numbers. The analysis uses over-approximations but we prove its soundness, meaning that all program executions respecting these sufficient conditions are indeed terminating. The abstract domain is parameterized by a numerical abstract domain for environments and a numerical abstract domain for functions. This parameterization allows to easily tune the trade-off between precision and cost of the analysis. We describe an instantiation of this generic domain with intervals and affine functions. We define all abstract operators, including widening to ensure convergence. To experiment with this domain, we have implemented a research prototype static analyzer FUNCTION (5.6) that yielded interesting preliminary results.

6.12.2. Abstract Domain to Infer Ordinal-Valued Ranking Functions

Participants: Caterina Urban, Antoine Miné.

We observed that, in some important cases (such as programs with unbounded non-determinism), there does not exist any ranking function over natural numbers. In [22] and [29] we propose a new abstract domain to automatically infer ranking functions over ordinals. We extended the domain of piecewise-defined natural-valued ranking functions introduced in the previous section to polynomials in ω, where the polynomial coefficients are natural-valued functions of the program variables. The abstract domain is parametric in the choice of the maximum degree of the polynomial, and the types of functions used as polynomial coefficients. We have enriched the FUNCTION prototype analyzer (5.6) with an instantiation of our domain using affine functions as polynomial coefficients. We successfully analyzed small but intricate examples that are out of the reach of existing methods. To our knowledge this is the first abstract domain able to reason about ordinals. Handling ordinals leads to a powerful approach for proving termination of imperative programs, which in particular subsumes existing techniques based on lexicographic ranking functions.

CELTIQUE Project-Team

5. New Results

5.1. Information Flow Tracking

Participants: Frédéric Besson, Nataliia Bielova, Delphine Demange, Thomas Jensen, David Pichardie.

We investigate different approaches for dynamically tracking information flows.

The first track of work is motivated by web-browser security. In a survey [15], we have classified JavaScript security policies and their enforcement mechanisms in a web-browser. We have identified the problem of stateless web tracking (fingerprinting) and have proposent a novel approach to hybrid information flow monitoring by tracking the knowledge about secret variables using logical formulae. A logic formula quantifies the amount of knowledge stored in a variable. This knowledge representation helps to compare and improve precision of hybrid information flow monitors. We define a generic hybrid monitor parametrised by a static analysis and derive sufficient conditions on the static analysis for soundness and relative precision of hybrid monitors. We instantiate the generic monitor with a combined static constant and dependency analysis. Several other hybrid monitors including those based on well-known hybrid techniques for information flow control are formalised as instances of our generic hybrid mon- itor. These monitors are organised into a hierarchy that establishes their relative precision. The whole framework is accompanied by a formalisation of the theory in the Coq proof assistant [19].

Our second activity is related to SAFE, a clean-slate design for a highly secure computer system, with pervasive mechanisms for tracking and limiting information flows. At the lowest level, the SAFE hardware supports fine-grained programmable tags, with efficient and flexible propagation and combination of tags as instructions are executed. The operating system virtualizes these generic facilities to present an information-flow abstract machine that allows user programs to label sensitive data with rich confidentiality policies. We present a formal, machine-checked model of the key hardware and software mechanisms used to control information flow in SAFE and an end-to-end proof of noninterference for this model in the Coq proof assistant [17].

5.2. Towards efficient abstract domains for regular language based static analysis

Participants: Thomas Genet, Valérie Murat, 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 safety 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. However, when dealing with infinite-state systems, Regular Tree Model Checking approaches may have some difficulties to represent infinite sets of data. We proposed Lattice Tree Automata, an extended version of tree automata to represent complex data domains and their related operations in an efficient manner. Moreover, we introduce a new completion-based algorithm for computing the possibly infinite set of reachable states in a finite amount of time. This algorithm is independent of the lattice making it possible to seamlessly plug abstract domains into a Regular Tree Model Checking algorithm[27]. As a first instance, we implemented in Timbuk a completion with an interval abstract domain. We shown that this implementation permits to scale up regular tree model-checking of Java programs dealing with integer arithmetics. Now, we aim at applying this technique to the static analysis of programming languages whose semantics is based on terms, like functional programming languages [38].

5.3. Result Certification of Static Program Analysers with Automated Theorem Provers

Participants: Frédéric Besson, Pierre-Emmanuel Cornilleau, Thomas Jensen.

The automation of the deductive approach to program verification crucially depends on the ability to efficiently infer and discharge program invariants. In an ideal world, user-provided invariants would be strengthened by incorporating the result of static analysers as untrusted annotations and discharged by automated theorem provers. However, the results of object-oriented analyses are heavily quantified and cannot be discharged, within reasonable time limits, by state-of-the-art auto- mated theorem provers. In the present work, we investigate an original approach for verifying automatically and efficiently the result of certain classes of object-oriented static analyses using off-the-shelf automated theorem provers. We propose to generate verification conditions that are generic enough to capture, not a single, but a family of analyses which encompasses Java bytecode verification and Fähndrich and Leino type-system for checking null pointers. For those analyses, we show how to generate tractable verification conditions that are still quantified but fall in a decidable logic fragment that is reducible to the Effectively Propositional logic. Our experiments confirm that such verification conditions are efficiently discharged by off-the-shelf automated theorem provers [20].

5.4. Formal Semantics for Multi-threaded Java

Participants: Delphine Demange, Vincent Laporte, David Pichardie.

Recent advances in verification have made it possible to envision trusted implementations of real-world languages. Java with its type-safety and fully specified semantics would appear to be an ideal candidate; yet, the complexity of the translation steps used in production virtual machines have made it a challenging target for verifying compiler technology. One of Java's key innovations, its memory model, poses significant obstacles to such an endeavor. The Java Memory Model is an ambitious attempt at specifying the behavior of multithreaded programs in a portable, hardware agnostic, way. While experts have an intuitive grasp of the properties that the model should enjoy, the specification is complex and not well-suited for integration within a verifying compiler infrastructure. Moreover, the specification is given in an axiomatic style that is distant from the intuitive reordering-based reasonings traditionally used to justify or rule out behaviors, and ill suited to the kind of operational reasoning one would expect to employ in a compiler. We take a step back, and introduces a *Buffered Memory Model* (BMM) for Java [26]. We choose a pragmatic point in the design space sacrificing generality in favor of a model that is fully characterized in terms of the reorderings it allows, amenable to formal reasoning, and which can be efficiently applied to a specific hardware family, namely x86 multiprocessors. Although the BMM restricts the reorderings compilers are allowed to perform, it serves as the key enabling device to achieving a verification pathway from bytecode to machine instructions. Despite its restrictions, we show that it is backwards compatible with the Java Memory Model and that it does not cripple performance on TSO architectures.

5.5. Formal Verification of Static Analysis

Participants: Sandrine Blazy, Martin Bodin, Thomas Jensen, Vincent Laporte, André Oliveira Maroneze, David Pichardie, Alan Schmitt.

Static analyzers based on abstract interpretation are complex pieces of software implementing delicate algorithms. Even if static analysis techniques are well understood, their implementation on real languages is still error-prone.

Using the Coq proof assistant, we formalized of a value analysis (based on abstract interpretation), and a soundness proof of the value analysis. The formalization relies on generic interfaces. The mechanized proof is facilitated by a translation validation of a Bourdoncle fixpoint iterator. The work has been integrated into the CompCert verified C-compiler. Our verified analysis directly operates over an intermediate language of the compiler having the same expressiveness as C. The automatic extraction of our value analysis into OCaml yields a program with competitive results, obtained from experiments on a number of benchmarks and comparisons with the Frama-C tool [21]. The value analysis was applied to a loop bound estimation tool for WCET analysis [22] relying also on program slicing and loop bound calculation.

Moreover, we formalized static analyses for logic programming, relying on results about the relative correctness of semantics in different styles; forward and backward, top-down and bottom-up. The results chosen are paradigmatic of the kind of correctness theorems that semantic analyses rely on and are therefore well-suited to explore the possibilities afforded by the application of interactive theorem provers to this task, as well as the difficulties likely to be encountered in the endeavour [29].

We also study the development of certified information flow analyses based on a formal semantics of JavaScript. We have in particular presented a technique for deriving semantic program analyses from a natural semantics specification of the programming language. The technique is based on the pretty-big-step semantics approach applied to a language with simple objects called O'While. We have specified a series of instrumentations of the semantics that makes explicit the flows of values in a program. This leads to a semantics-based dependency analysis, at the core, e.g., of tainting or information flow analyses in software security [32].

5.6. Certified JavaScript Semantics

Participants: Martin Bodin, Alan Schmitt.

JavaScript is the most widely used web language for client-side applications. Whilst the development of JavaScript was initially just led by implementation, there is now increasing momentum behind the ECMA standardisation process. The time is ripe for a formal, mechanised specification of JavaScript, to clarify ambiguities in the ECMA standards, to serve as a trusted reference for high-level language compilation and JavaScript implementations, and to provide a platform for high-assurance proofs of language properties. We present JScert, a formalisation of the current ECMA standard in the Coq proof assistant, and JSref, a reference interpreter for JavaScript extracted from Coq to OCaml. We give a Coq proof that JSref is correct with respect to JScert and assess JSref using test262, the ECMA conformance test suite. Our methodology ensures that JScert is a comparatively accurate formulation of the English standard, which will only improve as time goes on. We have demonstrated that modern techniques of mechanised specification can handle the complexity of JavaScript [25], [24].

5.7. Concurrent Reversibility

Participant: Alan Schmitt.

Concurrent reversibility has been studied in different areas, such as biological or dependable distributed systems. However, only "rigid" reversibility has been considered, allowing to go back to a past state and restart the exact same computation, possibly leading to divergence. We present a concurrent calculus featuring *flexible reversibility*, allowing the specification of alternatives to a computation to be used upon rollback. Alternatives in processes of this calculus are attached to messages. We show the robustness of this mechanism by encoding more complex idioms for specifying flexible reversibility, and we illustrate the benefits of our approach by encoding a calculus of communicating transactions [30].

5.8. Non linear analysis: fast inference of polynomial invariants

Participants: Thomas Jensen, David Cachera, Arnaud Jobin.

We have proposed an abstract interpretation based method for inferring polynomial invariants.Our analysis uses a form of weakest precondition calculus which was already observed to be well adapted to polynomial disequality guards, and which we extend to equality guards by using parameterized polynomial division. We have shown that the choice of a suitable division operation is crucial at each iteration step in order to compute the invariant. Based on this analysis, we have designed a constraint-based algorithm for inferring polynomial invariants. We have identified heuristics to solve equality constraints between ideals, and implemented the whole analysis algorithm in Maple.A salient feature of this analysis, which distinguishes it from the approaches proposed so far in the literature, is that it does not require the use of Gröbner base computations, which are known to be costly on parameterized polynomials. Our benchmarks show that our analyzer can successfully infer invariants on a sizeable set of examples, while performing two orders of magnitude faster than other existing implementations [16].

DEDUCTEAM Exploratory Action

6. New Results

6.1. Dedukti

The version 2.0 of the Dedukti system, developed by Ronan Saillard, has been released in July 2013. It is based on an improved version of the $\lambda\Pi$-calculus modulo where rewrite rules are explicitly added [31], and where the conditions for typing the rewrite rules are weakened.

This version is fully written in OCaml. It is smaller, far more efficient than the previous version, and permits to type-check much bigger files.

New features include a better reporting of errors, an interactive mode, an export functionality from Dedukti to the MMT format [53], and non-linear pattern matching.

6.2. Embeddings in the $\lambda\Pi$-calculus modulo

A new version of Coqine has been developed by Ali Assaf. This version is designed using a Coq plugin architecture, which allows for a smoother integration with Coq's code base and alleviates problems of maintainability that affected the previous version.

The implementation of Holide has been improved, by Ali Assaf. This improved version incorporates sharing at the level of terms and types. This optimization allows to reduce the type-checking time of the OpenTheory standard library from more than 30 minutes to less than 1 minute.

Catherine Dubois and Raphaël Cauderlier have studied a translation in the $\lambda\Pi$-calculus modulo of features coming from object oriented programming languages, such as inheritance and late binding. This compilation scheme has been applied to produce a new back-end for FoCaLize called Focalide, through a compilation to Dedukti. This translation can benefit from the flexibility of Dedukti to deal with more dynamic object-oriented languages than FoCaLiZe; they are currently working on generalizing this translation using ζ-calculus as a theoretical foundation for objects.

Resolution and superposition are proof-search methods that are used in state-of-the-art first-order automated theorem provers such as iProver, Vampire, E or SPASS. A shallow embedding of resolution and superposition proofs in the $\lambda\Pi$-calculus modulo has been proposed by Guillaume Burel, thus offering a way to check these proofs in a trusted setting, and to combine them with other proofs. This embedding has been implemented in particular as a backend of iProver Modulo, therefore allowing to check proofs found by iProver Modulo using Dedukti [20].

A shallow embedding in Dedukti of the tableaux proofs generated by Zenon modulo has been designed and implemented by Frédéric Gilbert [22], [23]. The embedding is based on a refined version of previous double-negation translations, introducing as less as possible double negations. This optimization has shown that more than half of the proofs found by Zenon modulo are not using the excluded-middle law, therefore being purely intuitionistic.

6.3. Automated Theorem Proving

Mélanie Jacquel (*Siemens*) and David Delahaye developed *Super Zenon* [5], a generalization of the extension of *Zenon* to superdeduction to handle any first order theory. To do so, they designed heuristics able to automatically transform axioms of a theory into rewrite rules. This new tool has been tested over the first order problems of the TPTP library and a significant increase has been observed. A first distribution of this tool (under GPL licence) is planned in the first months of 2014. In addition, an integration to the *Rodin* platform is also planned with the help of Laurent Voisin (*Systerel*). This integration should allow us to apply this tool in the context of *Event-B*.

Pierre Halmagrand, David Delahaye, Damien Doligez, and Olivier Hermant developed *Zenon Modulo* [22], [23], an extension of *Zenon* to Deduction modulo. Like *Super Zenon*, this new tool is able to deal with any first order theory and relies on an heuristic able to automatically transform axioms of a theory into rewrite rules. This tool has also been tested over the first order problems of the TPTP library and a similar increase of performance (compared to *Super Zenon*) has been observed. Frédéric Gilbert has developed a *Dedukti* backend for this extension, which is based on a double-negation transformation that allows us to transform classical proofs produced by *Zenon Modulo* into intuitionistic proofs in *Dedukti*. This tool is intended to be applied in the framework of the *BWare* project in order to automatically verify proof obligations coming from the modeling of industrial applications. To do so, the idea is to manually transform the *B* set theory into a theory modulo and provide it to *Zenon Modulo* in order to verify the proof obligations of the *BWare* project.

Guillaume Burel and Simon Cruanes have designed a method to scan sets of first-order clauses in order to detect the presence of instances of axiomatic theories (group structures, total orderings, etc.), even during a saturation process (so that theories that only become apparent during the proof search can be detected) [21]. To this end, they introduced the concept of *meta-prover*, a Datalog system that reasons over properties of the problem, and communicates with the saturation prover. This technique made some applications possible, such as the use of generic lemma and an equational redundancy criterion for some theories, and was implemented in Zipperposition.

Simon Cruanes has been working on superposition modulo linear arithmetic, using Zipperposition as a test bed. The focus is on problems with rational or integer arithmetic mixed with first-order reasoning, an area in which SMT solvers struggle. The work is still preliminary, but shows promising results.

Depending on the logic for finite structures, which is defined by Gilles Dowek and Ying Jiang (Beijing), Kailiang Ji has extended the use of proof search algorithms in Deduction modulo to automatically prove some graph properties, such as (un)reachability, which can be described by CTL formulas. A technical report about this has been given on Locali 2013 in Beijing.

Together with Tayssir Touili (University Paris Diderot) Hugo Macedo has shown how to advance the performance of the application of model checking techniques in the domain of malicious software detection. The work consisted in leveraging the reachability analysis used in the model checking of pushdown systems to infer malicious behavior patterns from known malware. From such new application a malware detection tool was prototyped and put to the test with instances of "in the wild" (real world) malicious software. This work was published in a large security venue and the details about the technique follow in [29].

Kim-Quyen Ly extended her formally-proved (in Coq) automated termination-certificate (for first-order rewrite systems) verifier Rainbow for dealing with certificates using arguments filtering [39] and other termination techniques.

6.4. Proof theory

The conservativity of the embedding of pure type systems in the $\lambda\Pi$-calculus modulo was proved by Ali Assaf. This result extends those of Cousineau and Dowek [46] and further justifies the use of the $\lambda\Pi$-calculus modulo as a logical framework. This embedding is the basis for the automated translation tools Holide and Coqine.

Frédéric Blanqui, Jean-Pierre Jouannaud (Univ. Paris 11) and Albert Rubio (Technical University of Catalonia) have developed a method aiming at carrying out termination proofs for higher-order calculi. CPO appears to be the ultimate improvement of the higher-order recursive path ordering (HORPO) [45] in the sense that this definition captures the essence of computability arguments *à la* Tait and Girard, therefore explaining the name of the improved ordering. It has been shown that CPO allows to consider higher-order rewrite rules in a simple type discipline with inductive types, that most of the guards present in the recursive calls of its core definition cannot be relaxed in any natural way without losing well-foundedness, and that the precedence on function symbols cannot be made more liberal anymore.

Frédéric Blanqui worked on the formalization in the Coq proof assistant of various definitions of the notion of α-equivalence on pure λ-terms. In particular, he formalized and formally proved equivalent the definitions

of Church (1932), Curry and Feys (1958), Krivine (1993), and Gabbay and Pitts (1999). This work is freely available from the CoLoR library released on December 13th.

Alejandro Díaz-Caro and Gilles Dowek have introduced an extension of λ-calculus with pairs where isomorphic types are equated. Identifying some types requires to also identify some terms via an equivalence relation on terms, leading to an interesting calculus, which is related to several known non-deterministic and probabilistic calculi. A preliminary version of this work has been published on [24]. A complete version in simple types, with its proof of normalisation, is currently under review.

Together with Ying Jiang, Gilles Dowek has started to investigate the links between model-checking and proof-checking. This has materialized by an encoding of CTL for a finite model in predicate logic and by the definition of a proof-system for CTL.

Olivier Hermant has studied optimized versions of double-negation translations, that allow to switch between classical and intuitionistic logics. Such an algorithm has been implemented in Zenon's backend to Dedukti by Frédéric Gilbert. Gilles Dowek has given new version of Gödel's translation of classical logic into constructive logic. This translation is homomorphic, hence it can be seen as a mere definition of the classical connectives from the constructive ones.

6.5. Safety of aerospace systems

Pierre Néron has designed a method to transform straight line programs, such as those used in some aerospace systems into others that do not use some operations such as, square roots and divisions that cannot be performed exactly on decimal numbers. To this end he has defined a new notion of anti-unification, called *constrained anti-unification*, and a new anti-unification algorithm.

6.6. Models of Computation

Alejandro Díaz-Caro and Gilles Dowek have shown how to provide a structure of probability space to the set of execution traces on a non-confluent abstract rewrite system, by defining a variant of a Lebesgue measure on the space of traces. Then, they showed how to use this probability space to transform a non-deterministic calculus into a probabilistic one. As an example, they applied this technique to the previously introduced non-deterministic calculus. [25]

Ali Assaf and Alejandro Díaz-Caro, together with Simon Perdrix (Nancy), Christine Tasson (PPS) and Benoît Valiron (PPS) have determined the relationship between the algebraic λ-calculus, a fragment of the differential λ-calculus and the linear-algebraic λ-calculus, a candidate λ-calculus for quantum computation. Both calculi are algebraic: each one is equipped with an additive and a scalar-multiplicative structure, and their set of terms is closed under linear combinations. However, the two languages were built using different approaches: the former is a call-by-name language whereas the latter is call-by-value; the former considers algebraic equalities whereas the latter approaches them through rewrite rules. They have analysed how these different approaches relate to one another, proposing four canonical languages based on each of the possible choices: call-by-name versus call-by-value, algebraic equality versus algebraic rewriting. They have shown that the various languages simulate one another. Preliminary versions of this work where published in [47] and [41]. Now they are working on a journal version filling the gaps between these previous works.

Together with Pablo Arrighi (Grenoble) and Benoît Valiron (PPS), Alejandro Díaz-Caro has described a type system for the linear-algebraic lambda-calculus. The type system accounts for the linear-algebraic aspects of this extension of lambda-calculus: It is able to statically describe the linear combinations of terms that will be obtained when reducing the programs. This gives rise to an original type theory where types, in the same way as terms, can be superposed into linear combinations. They have proven that the resulting typed lambda-calculus is strongly normalising and features a weak subject reduction. In addition, they have shown how to naturally encode matrices and vectors in this typed calculus [34].

Gilles Dowek has investigated a new definition of the notion of a chaotic system that can be applied to discrete systems and that is compatible with the principle of a finite density of information.

The paper Call-by-value non-determinism in a linear logic type discipline by Alejandro Díaz-Caro, Giulio Manzonetto and Michele Pagani has been published [26].

The paper Universality in two dimensions of Gilles Dowek and Nachum Dershowitz has been published.

The paper Linear-algebraic lambda-calculus: higher-order, encodings and confluence of Pablo Arrighi and Gilles Dowek has been published.

The book *Lambda Calculus with Types* , written by Henk Barendregt, Wil Dekkers, Richard Statman, and 11 contributors, including Gilles Dowek, has been published.

6.7. Constraint solving

Catherine Dubois has extended the formally verified constraint solver (on finite domains) she has developed with Matthieu Carlier and Arnaud Gotlieb with a new local consistency property (bound-consistency).

GALLIUM Project-Team

6. New Results

6.1. Formal verification of compilers and static analyzers

6.1.1. The CompCert formally-verified compiler

Participants: Xavier Leroy, Jacques-Henri Jourdan, Robbert Krebbers.

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 part, translating the Cminor intermediate language to PowerPC assembly and reusable for source languages other than C [5], and a front-end translating 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 Caml 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 released three versions of CompCert. Version 1.13, released in March, improves conformance with the ISO C standard by defining the semantics of comparisons involving pointers "one past" the end of an array. Such comparisons used to be undefined behaviors in earlier versions of CompCert. Robbert Krebbers formalized a reasonable interpretation of the ISO C rules concerning pointers "one past" and adapted CompCert's proofs accordingly. CompCert 1.13 also features minor performance improvements for the ARM and PowerPC back-ends, notably for parameter passing via stack locations.

Version 2.0 of CompCert, released in June, re-architects the compiler back-end around the new register allocator described in section 6.1.2 . Besides improving the performance of generated code, this new allocator made it possible to add support for 64-bit integers, that is, the `long long` and `unsigned long long` data types of ISO C99. Most arithmetic operations over 64-bit integers are expanded in-line and proved correct, but a few complex operations (division, modulus, and conversions to and from floating-point numbers) are implemented as calls into library functions.

Moreover, conformance with Application Binary Interfaces was improved, especially concerning the passing of function parameters and results of type `float` (single-precision FP numbers).

Finally, CompCert 2.0 features preliminary support for debugging information. The `-g` compiler flag causes DWARF debugging information to be generated for line numbers and call stack structure. However, no information is generated yet for C type definitions and variable declarations.

Version 2.1, released in October, addresses several shortcomings of CompCert for embedded system codes, as identified by Airbus during their experimental evaluation of CompCert. In particular, CompCert 2.1 features the `_Alignas` modifier introduced in ISO C2011, to support precise control of alignment of global variables and structure fields, and uses this modifier to implement packed structures in a more robust fashion than in earlier releases. Xavier Leroy also implemented and proved correct the optimization of integer divisions by constants introduced by Granlund and Montgomery [40].

6.1.2. Register allocation with validation a posteriori

Participant: Xavier Leroy.

Register allocation (the placement of program variables in processor registers) has a tremendous impact on the performance of compiled code. However, advanced register allocation techniques are difficult to prove correct, as they involve complex algorithms and data structures. Since the beginning of the CompCert project, we chose to avoid some of these difficult proofs by performing validation *a posteriori* for part of register allocation: the IRC graph coloring algorithm invoked during register allocation is not proved correct; instead, its results are verified at every compiler run to be a correct coloring of the given interference graph, using a simple validator proved sound in Coq.

In CompCert 2.0, we push this validation-based approach further. The whole register allocator is now subject to validation a posteriori and no longer needs to be proved correct. The validator follows the algorithm invented by Rideau and Leroy [50] and further developed by Tassarotti and Leroy. It proceeds by backward dataflow analysis of symbolic equations between program variables, registers, and stack locations.

Consequently, the new register allocator for CompCert 2.0 is much more aggressive than that of CompCert 1: it features a number of optimizations that could not be proved correct in CompCert, including live-range splitting, better handling of two-address operations and other irregularities of the x86 instruction set, an improved spilling strategy, and iterating register allocation to place temporaries introduced by spilling. Moreover, the new register allocator can handle program variables of 64-bit integer types, allocating them to pairs of 32-bit registers or stack locations. The new register allocator improves the performance of generated x86 code by up to 10% on our benchmarks.

6.1.3. Formal verification of static analyzers based on abstract interpretation

Participants: Sandrine Blazy [EPI Celtique], Vincent Laporte [EPI Celtique], Jacques-Henri Jourdan, Xavier Leroy, David Pichardie [EPI Celtique].

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 should be able to handle the same large subset of the C language as the CompCert compiler; support a combination of abstract domains, including relational domains; and 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 codes.

This year, Jacques-Henri Jourdan worked on numerical abstract domains for the static analyzer. First, he designed, programmed and proved correct an abstraction layer that transforms any relational abstract domain for mathematical, arbitrary-precision integers into a relational abstract domain for finite-precision machine integers, taking overflow and "wrap-around" behaviors into account. This domain transformer makes it possible to design numerical domains without taking into account the finiteness of machine integers. Then, he implemented and proved sound non-relational abstract domains for intervals of integers and of floating-point numbers, supporting almost all CompCert arithmetic operations.

In collaboration with team Celtique, we studied which intermediate languages of the CompCert C compiler are suitable as source language for the static analyzer. Early work by Blazy, Laporte, Maroneze and Pichardie [36] performs abstract interpretation over the RTL intermediate language, a simple language with unstructured control (control-flow graph). However, this language is too low-level to support reporting alarms at the level of the source C program.

Later this year, we decided to use the C#minor intermediate language of CompCert as source language for analysis. This language has mostly structured control (if/then/else, C loops, and `goto`), and is much closer to the source C program. Then, Jacques-Henri Jourdan, Xavier Leroy and David Pichardie designed a generic abstract interpreter for the C#minor language, parameterized by an abstract domain of execution states, using structured fixpoint iteration for loops and a function-global iteration for `goto`. Jacques-Henri Jourdan is in the process of proving the soundness of this abstract interpreter in Coq.

6.1.4. Formalization of floating-point arithmetic

Participants: Sylvie Boldo [EPI Toccata], Jacques-Henri Jourdan, Xavier Leroy, Guillaume Melquiond [EPI Toccata].

Last year, we replaced the axiomatization of floating-point numbers and arithmetic operations used in early versions of CompCert by a fully-formal Coq development, building on the Coq formalization of IEEE-754 arithmetic provided by the Flocq library of Sylvie Boldo and Guillaume Melquiond. A paper describing this work was presented at the ARITH 2013 conference [15].

This year, we extended this formalization of floating-point arithmetic with a more precise modeling of "Not a Number" special numbers, reflecting the signs and payloads of these numbers into their bit-level, in-memory representation. We also proved correct more algebraic identities over FP computations, such as $x/2^n = x \times 2^{-n}$ if $|n| < 1023$, as well as nontrivial implementation schemes for conversions between integer and FP numbers, whose correctness rely on subtle properties of the "round to odd" rounding mode. These extensions are described in a draft journal paper under submission [29], and integrated in version 2.1 of CompCert.

6.1.5. Formal verification of hardware synthesis

Participants: Thomas Braibant, Adam Chlipala [MIT].

Verification of hardware designs has been thoroughly investigated. Yet, obtaining provably correct hardware of significant complexity is usually considered challenging and time-consuming. Hardware synthesis aims to raise the level of description of circuits, reducing the effort necessary to produce them. This yields two opportunities for formal verification: a first option is to verify (part of) the hardware compiler; a second option is to study to what extent these higher-level design are amenable to formal proof.

Continuing work started during a visit at MIT under the supervision of Adam Chlipala, Thomas Braibant worked on the implementation and proof of correctness of a prototype hardware compiler. This compiler produces descriptions of circuits in RTL style from a high-level description language inspired by BlueSpec. Formal verification of hardware designs of mild complexity was conducted at the source level, making it possible to obtain fully certified RTL designs. A paper describing this compiler and two examples of certified designs was presented at the CAV 2013 conference [16].

6.2. Language design and type systems

6.2.1. The Mezzo programming language

Participants: Jonathan Protzenko, François Pottier, Thibaut Balabonski, Armaël Guéneau, Cyprien Mangin.

In the past ten years, the type systems community and the separation logic community, among others, have developed highly expressive formalisms for describing ownership policies and controlling side effects in imperative programming languages. In spite of this extensive knowledge, it remains very difficult to come up with a programming language design that is simple, effective (it actually controls side effects!) and expressive (it does not force programmers to alter the design of their data structures and algorithms).

The Mezzo programming language aims to bring new answers to these questions.

This year, we:

- made significant progress on the proof of soundness, by rewriting it in a more modular fashion;
- improved the implementation, by formalizing the algorithms and rewriting significant parts of the type-checker;
- hosted two interns who explored arithmetic reasoning and modeling of the iterator protocol, respectively;
- formalized libraries for concurrent programming in Mezzo;
- wrote both an interpreter and a compiler for the language.

A paper on Mezzo appeared in the ICFP 2013 conference [21].

During the previous year (2012), François Pottier wrote a formal definition of Mezzo, and proved that Mezzo is type-safe: that is, well-typed programs cannot crash. The proof was machine-checked using Coq. This year, Thibaut Balabonski and François Pottier extended this formalization with support for concurrency and dynamically-allocated locks, and proved that well-typed programs not only cannot crash, but also are data-race free.

The structure of the proof was re-worked so as to make it more modular. A paper, which emphasizes this modularity, has been submitted for presentation at a conference.

The new concurrent features have been integrated in the core library of Mezzo by Thibaut Balabonski. Further concurrent libraries have been included to provide more communication primitives, such as channels for message passing.

Jonathan Protzenko worked on formalizing the type-checking algorithms currently used in the Mezzo prototype compiler. This led to practical results in the form of improvements to the type-checker: we now type-check more programs, and the success of the type-checker is more predictable as well. Some soundness bugs have been identified and fixed. The design of some of the language's features has been improved as well.

The formalization of the type-checker was presented at the IFL 2013 conference, and is to appear in the post-symposium proceedings in 2014.

We set out to promote Mezzo in the wild. Protzenko packaged the software to make it available widely via OPAM, wrote a tutorial for end-users [34], communicated through blog posts about the language, and released the source code online for others to contribute.

We also spread the word about Mezzo through various seminar talks and discussions with other teams (Carnegie-Mellon university, Cambridge Computer Lab, Aarhus University, Brasilia University), and by communicating in international conferences (ICFP'13, FSFMA'13).

This year, two interns worked with us on Mezzo. Armaël Guéneau (L3; June-July 2013) and Cyprien Mangin (M1; April-July 2013) explored several experimental aspects of the language. In particular, Armaël worked on an encoding of iterators in an object-oriented style, which involves transfers of ownership and typestate changes; while Cyprien improved the treatment of arrays and implemented an experimental extension of Mezzo with arithmetic assertions. Armaël presented his work at the workshop HOPE 2013. This work is also described in a short unpublished paper [33].

6.2.2. System F with coercion constraints

Participants: Julien Cretin, Didier Rémy.

Expressive type systems often allow non trivial conversions between types, which may lead to complex, challenging, and sometimes ad hoc type systems. Such examples are the extension of System F with type equalities to model GADTs and type families of Haskell, or the extension of System F with explicit contracts. A useful technique to simplify the meta-theoretical studies of such systems is to make type conversions explicit as "coercions" inside terms.

Following a general approach to coercions, we extended System F with a richer type-level language and a proposition language. Propositions contain a first-order logic, a coinduction mechanism, coherence assertions and coercion assertions. Types are classified by kinds and extended in order to handle lists of types. We introduce a particular kind restricting a previous kind to its types satisfying a proposition. Abstracting over such a kind means abstracting over arbitrary propositions, and thus enables coercion abstraction. Type abstraction must be coherent: the kind of the abstract type has to be inhabited by a witness type. This language, called Fcc, extends our previous language parametric F-iota and additionally subsumes Constraint ML.

We also extended Fcc with incoherent polymorphism in order to model GADTs. Unlike coercions and thus coherent polymorphism, incoherent polymorphism is not erasable. But in counterpart, incoherent abstraction does not require the kind to be inhabited. Since abstracting over incoherent types permits to write unsound terms, incoherent abstraction has to block the reduction of terms.

This work is part of Julien Cretin's Ph.D. dissertation [11], which will be defended in January 2014.

6.2.3. Type inference for GADTs

Participants: Jacques Garrigue [Nagoya University], Didier Rémy.

Type inference for generalized algebraic data types (GADTs) is inherently non monotone: assuming more specific types for GADTs may ensure more invariants, which may result in more general types. This is problematic for type inference and some amount of type annotations is required.

Moreover, even when types of GADTs parameters are explicitly given, they introduce equalities between types, which makes them inter-convertible but with a limited scope. This may create an ambiguity when leaving the scope of the equation: which element should be used for representing the equivalent forms? Idealy, one should use a type disjunction, but this is not allowed—for good reasons. Hence, to avoid arbitrary choices, these situations must be rejected as ambiguous, forcing the user to write more annotations to resolve the ambiguities.

We proposed a new approach to type inference with GADTs. While some uses of equations are unavoidable and create *real* ambiguities, others are gratuitous and create *artificial* ambiguities, To distinguish between the two we introduced *ambivalent types*, which are a way to trace unavoidable uses of equations within types themselves. We then redefined ambiguities so that only ambivalent types become ambiguous and should be rejected or resolved by a programmer annotation. Interestingly, this solution is fully compatible with unification-based type inference algorithms used in ML dialects.

This work was presented at the APLAS 2013 conference [20]. It is also implemented in the OCaml language since version 4.00.

6.2.4. GADTs and Subtyping

Participants: Gabriel Scherer, Didier Rémy.

Following the addition of GADTs to the OCaml language in version 4.00 released this year, we studied the theoretical underpinnings of variance subtyping for GADTs. The question is to decide which variances should be accepted for a GADT-style type declaration that includes type equality constraints in constructor types. This question exposes a new notion of decomposability and unexpected tensions in the design of a subtyping relation. A paper describing our formalization was presented at the ESOP 2013 conference [23].

6.2.5. Singleton types for code inference

Participants: Gabriel Scherer, Didier Rémy.

We continued working on the use of singleton types for code inference. If we can prove that a type contains, in a suitably restricted pure lambda-calculus, a unique inhabitant modulo program equivalence, the compiler can infer the code of this inhabitant. This opens the way to type-directed description of boilerplate code, through type inference of finer-grained type annotations. As this is still work in progress, there was no publication on this topic this year, but we presented our directions on three occasions: at the PLUME team in ENS Lyon, at the LIX team in École Polytechnique (whose proof-search research is highly relevant to our work), and at the Dependently Typed Programming workshop (satellite of the International Conference on Functional Programming) in Boston.

6.2.6. Open closure types

Participants: Gabriel Scherer, Jan Hoffmann [Yale University, FLINT group].

During a visit to Yale, Gabriel Scherer worked with Jan Hoffmann on a type system for program analysis of higher-order functional languages. Open closure types are a novel typing construct that lets the type system statically reason about closure variables present in the lexical context. This allows fine-grained analysis (e.g., for resource consumption or information-flow control) of functional programming patterns such as function currying. This work was presented at the LPAR 2013 conference [22] (Logic for Programming, Artificial Intelligence, and Reasoning) in October.

6.3. Shared-memory parallelism

6.3.1. Algorithms and data structures for parallel computing

Participants: Umut Acar, Arthur Charguéraud [EPI Toccata], Mike Rainey.

The ERC Deepsea project, with principal investigator Umut Acar, started in June and is hosted by the Gallium team. This project aims at developing techniques for parallel and self-adjusting computations 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 in the previous three years. 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 have recently been pursuing two main lines of work.

We have been developing an algorithm that is able to perform dynamic load balancing in the style of work stealing but without requiring atomic read-modify-write operations. These operations may scale poorly with the number of cores due to synchronization bottlenecks. We have designed the algorithm, proved it correct using a new technique for the x86-TSO weak memory model. We have evaluated our algorithm on a modern multicore machine. Although we use no synchronization operations, we achieve performance that is no more than a few percent slower than the industrial-strengh algorithm, even though the industrial-strength algorithm takes full advantage of synchronization operations. We have a soon-to-be-submitted research article describing our contributions [25].

The design of efficient parallel graph algorithms requires a sequence data structure that supports logarithmic-time split and concatenation operations in addition to push and pop operations with excellent constant factors. We have designed such a data structure by building on a recently introduced data structure called Finger Tree and by integrating a "chunking" technique. Our chunking technique is based on instantiating the leaves of the Finger Tree with chunks of contiguous memory. Unlike previous chunked data structures, we are able to prove efficient constant factors even in worst-case scenarios. Moreover, we implemented our data structure in C++ and OCaml and showed it to be competitive with state-of-the-art sequence data structures that do not support split and concatenation operations. We are currently writing a report on our results.

6.3.2. Weak memory models

Participants: Luc Maranget, Jacques-Pascal Deplaix, Jade Alglave [University College London].

Modern multicore and multiprocessor computers do not follow the intuitive "Sequential Consistency" model that would define a concurrent execution as the interleaving of the execution of its constituting 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 instruction and to the presence of sophisticated (and cooperating) caching devices between processors and memory.

In the last few years, Luc Maranget took part in an international research effort to define the semantics of the computers of the multi-core era. This research effort relies both on formal methods for defining the models and on intensive experiments for validating the models. Joint work with, amongst others, Jade Alglave (now at University College London) and Peter Sewell (University of Cambridge) achieved several significant results, including two semantics for the IBM Power and ARM memory models: one of the operational kind [52] and the other of the axiomatic kind [46]. In particular, Luc Maranget is the main developer of the **diy** tool suite (see section 5.3). Luc Maranget also performs most of the experiments involved.

In 2013, Luc Maranget pursued this collaboration. He mainly worked with Jade Alglave to produce a new model for Power/ARM. The new model is simpler than the previous ones, in the sense that it is based on fewer mathematical objects and can be simulated more efficiently than the previous models. The new model is at the core of a journal submission which is now at the second stage of reviewing. The submitted work contains in-depth testing of ARM devices which led to the discovery of anomalous behaviours acknowledged as such by our ARM contact, and of legitimate features now included in the model. The new model also impacted our **diy** tool suite that now includes a generic memory model simulator built by following the principles exposed in the submitted article. At the moment the new simulator is available as an experimental release (http://diy.inria.fr/herd). It will be include in future releases of the tool suite.

In the same research theme, Luc Maranget supervises the internship of Jacques-Pascal Deplaix (EPITECH), from Oct. 2013 to May 2014. The internship aims at extending **litmus**, our tool to to run tests on hardware: at the moment **litmus** accepts test written in assembler; Jacques-Pascal is extending **litmus** so that it accepts tests written in C. The general objective is to achieve conformance testing of C compilers and machines with respect to the new C11/C++11 standard.

6.4. The OCaml language and system

6.4.1. The OCaml system

Participants: Damien Doligez, Alain Frisch [Lexifi SAS], Jacques Garrigue [University of Nagoya], Fabrice Le Fessant, Xavier Leroy, Gabriel Scherer.

This year, we released version 4.01.0 of the OCaml system. This is a major release that fixes about 140 bugs and introduces 44 new features suggested by users. Damien Doligez acted as release manager for this version.

The major innovations in OCaml 4.01 are:

- The overloading of variant constructors and record field labels, resolved using typing information. Before this, programmers had to use globally unique field labels across all record types. The new typechecking algorithm enables programmers to use more natural names for fields in their data structures. The algorithm is carefully engineered to preserve principality of inferred types.
- New warnings give the programmer the option of applying very strict checking of problematic constructs in the source code.

Other features of this release include:

- Suggestion of possible typos in case of "unbound identifier" error.
- New infix application operators in the standard library.
- Options to reduce the verbosity (and enhance the readability) of error messages.
- Many internal improvements, especially in compiler performance.

In parallel, we designed and experimented with several new features that are candidate for inclusion in the next major release of OCaml in 2014:

- Module aliases: a more efficient way of typechecking and compiling module declarations of the form `module M = ModuleName`, providing a lighter, more practical alternative to packed modules and reducing the need for name spaces.
- Extension points and preprocessing by rewriting abstract syntax trees: this approach provides an alternative to Camlp4 for macro processing and automatic code generation.
- A native code generator for the new ARM 64 bit instruction set (also known as AArch64).
- Several ongoing experiments to improve the performance of OCaml-compiled code: more aggressive function inlining and constant propagation; more unboxing of numbers; and a pass of common subexpression elimination.

6.4.2. Run-time types for the OCaml language

Participants: Grégoire Henry, Jacques Garrigue [University of Nagoya], Fabrice Le Fessant.

With the addition of GADTs to OCaml in version 4.00, it is now possible to provide a clean implementation of run-time types in the language, thus allowing the definition of polytypic function, a.k.a. generic function defined by case analysis on the structure of its argument's type. However, when integrating this mechanism into the language, its interaction with other parts of the type-system proved delicate, the main difficulty being the semantic of abstract types.

In collaboration with Jacques Garrigue during a 3 month stay in Japan, Grégoire Henry worked on different semantics for the runtime representation of abstract types. They tried to design a mechanism that preserves abstraction by default, and still allows to propagate type information when requested by the programmer.

6.4.3. Multi-runtime OCaml

Participants: Luca Saiu, Fabrice Le Fessant.

Multicore architectures are now broadly available, and developers expect their programs to be able to benefit from them. In OCaml, there is no portable way to use such architectures, as only one OCaml thread can run at any time.

As part of the ANR project "BWare", Luca Saiu and Fabrice Le Fessant developed a multi-runtime version of OCaml that takes advantage of multicore architectures. In this version, a program can start several runtimes that can run on different cores. As a consequence, OCaml threads running on different runtimes can run concurrently. This implementation required a lot of rewriting of the OCaml runtime system (written in C), to make all global variables context-dependent and all functions reentrant. The compiler was also modified to generate reentrant code and context-dependent variables. The sources of the prototype were released in September 2013, to be tested by users.

Luca Saiu then developed a library based on skeletons to facilitate the development of parallel applications that take advantage of the multi-runtime architecture.

6.4.4. Evaluation strategies and standardization

Participants: Thibaut Balabonski, Flávio de Moura [Universidade de Brasília].

During the past years, Thibaut Balabonski studied evaluation strategies, laziness and optimality for functional programming languages, in particular in relation to pattern matching. These investigations continued this year, with two highlights:

- Publication in the ICFP conference [14] of a theoretical result relating fully lazy evaluation (as can be found in some Haskell compilers) to optimal reduction in the weak λ-calculus.
- Collaboration with Flávio de Moura (Universidade de Brasília) on so-called "standard" evaluation strategies for a calculus with rich pattern matching mechanisms (the *Pure Pattern Calculus* of Jay and Kesner [42]). The challenge here lies in that the calculus does not satisfies the usual stability properties. As a consequences, standard strategies are not unique anymore, and new approaches are needed. A paper is in preparation.

6.5. Software specification and verification

6.5.1. Tools for TLA+

Participants: Damien Doligez, Jael Kriener, Leslie Lamport [Microsoft Research], Stephan Merz [EPI VeriDis], Tomer Libal [Microsoft Research-Inria Joint Centre], Hernán Vanzetto [Microsoft Research-Inria Joint Centre].

Damien Doligez is head of the "Tools for Proofs" team in the Microsoft-Inria Joint Centre. The aim of this team is to extend the TLA+ language with a formal language for hierarchical proofs, formalizing the ideas in [43], and to build tools for writing TLA+ specifications and mechanically checking the corresponding formal proofs.

This year, the TLA+ tools were released as open-source (MIT license), and in September we released a new version of the TLA+ Proof System (TLAPS), an environment for writing and checking TLA+ proofs. This environment is described in [38].

We have implemented a (not yet released) extension of TLAPS to deal with proofs of temporal formulas, using the propositional temporal logic prover LS4 as a back-end. Until now, TLAPS could only be used to prove safety properties (invariants). With this new version, our users will be able to prove liveness properties (absence of deadlock), refinement relations between specifications, etc.

Jael Kriener started a 2-year post-doc contract in December. She is working on theoretical and implementation aspects of TLA+ and TLAPS.

Web sites:
http://research.microsoft.com/users/lamport/tla/tla.html
http://tla.msr-inria.inria.fr/tlaps

6.5.2. The Zenon automatic theorem prover

Participants: Damien Doligez, David Delahaye [CNAM], Pierre Halmagrand [CNAM], Olivier Hermant [Mines ParisTech], Mélanie Jacquel [CNAM].

Damien Doligez continued the development of Zenon, a tableau-based prover for first-order logic with equality and theory-specific extensions.

David Delahaye and Mélanie Jacquel designed and implemented (with some help from Damien Doligez) an extension of Zenon called SuperZenon, based on the Superdeduction framework of Brauner, Houtmann, and Kirchner [37]. Mélanie Jacquel defended her thesis on this subject in April.

Pierre Halmagrand did an internship and started a thesis on integrating Deduction Modulo in Zenon; some results of this work are described in two papers published at LPAR [19] and IWIL [18].

6.5.3. Implementing hash-consed structures in Coq

Participants: Thomas Braibant, Jacques-Henri Jourdan, David Monniaux [CNRS, VERIMAG].

Hash-consing is a programming technique used to implement maximal sharing of immutable values in memory, keeping a single copy of semantically equivalent objects. Hash-consed data-structures give a unique identifier to each object, allowing fast hashing and comparisons of objects. This may lead to major improvements in execution time by itself, but it also make it possible to do efficient memoization of computations.

Hash-consing and memoization are examples of imperative techniques that are of prime importance for performance, but are not easy to implement and prove correct using the purely functional language of a proof assistant such as Coq. In a joint article at ITP 2013 [17], we described three different implementation techniques for hash-consed data-structures in Coq through the running example of Binary Decision Diagrams (BDDs). BDDs are representations of Boolean functions, and are often used in software and hardware verification tools (e.g., model checkers).

We substantially improved the work described in this ITP 2013 article afterwards. First, we came up with a fourth implementation technique for hash-consed data-structures in Coq. Then, we performed an in-depth comparative study of how our "design patterns" for certified hash-consing fare on two real-scale examples: BDDs and lambda-terms. This work is currently under revision for publication in a journal.

6.5.4. Working with names and binders

Participant: François Pottier.

François Pottier released **dblib**, a Coq library that helps work with de Bruijn indices in a generic and lightweight manner. This library is used in the formalization of Mezzo (see section 6.2.1). It is available at http://gallium.inria.fr/~fpottier/.

6.6. Technology transfer

6.6.1. Analysis of the Scilab Language

Participants: Fabrice Le Fessant, Michael Laporte.

The Scilab language is a scripting language providing easy access to efficient implementations of mathematical operations (on matrices, for example). It suffers from the lack of verifications of an untyped language, together with the performance problems of an interpreted language. As part of the FUI Richelieu project, Fabrice Le Fessant and Michael Laporte have been investigating solutions to these issues.

The first part of the work was to clarify the semantics of the Scilab language. For that, an interpreter was implemented in OCaml, based on the C++ AST provided by the forthcoming version 6 of Scilab. This work exhibited a number of bugs in the new implementation, and proved to be more performant than the C++ implementation, thanks to a better algorithm to manage the dynamic scopes of Scilab.

The second part of the work was to understand how users write Scilab code. For that, a style-checking application, called *Scilint*, has been developed. It implements static checking of some properties of Scilab programs, to be able to detect runtime errors before running the program. Warnings are displayed for suspicious cases. Using Scilint on large sets of Scilab code (from the Scilab forge or the Atom repository) showed that the most erroneous features of Scilab are commonly used and that, to achieve the ultimate goal of partial typing of the language, a subset of the language must be specified that the user should conform to, in order for the code to benefit from the next part of the work, i.e. just-in-time compilation.

MARELLE Project-Team

6. New Results

6.1. Bourbaki, Sets and Ordinals

Participant: José Grimm [correspondant].

In previous years, we developped a formal library describing the part of the Bourbaki books on set theory, cardinals and ordinals, [18]. Here are ome additions to the library.

Since addition of ordinals is non-commutative, the sum of n ordinals x_1 to x_n depends on their ordering; the maximum number $f(n)$ is a priori bounded by $n!$, and we have shown that it satisfies a recurrence relation (R), Bourbaki asks, in an exercise, to show that $f(n) = 81f(n-5)$ for $n \geq 20$. This is an easy consequence of an explicit formula (F) for f. That (R) implies (F) can be expressed in pure Coq (with binary integers), but we have no idea how to prove it.

We proved some facts of the theory of models: the set V_ω of hereditarily finite sets satisfies ZF (but not the axiom of infinity); the von Neumann universe satisfies ZF and AF, there is a subset of the universe satisfiying ZF containing no inaccessible cardinal. We have also studied the set of formulas and show the theorem of Lövenheim-Skolem.

The main contribution this year is the study of some families of ordinals. If the family is internally closed and too big to be a set, then it is the image of a normal (continuous and strictly increasing) function, called the enumeration function of the family. The family of fix-points of a normal function satisfies this property, and the enumeration of this family is called the first derived function. There is a derivation at every order. For instance, the first derivation of $x \mapsto 1 + x$ is $x \mapsto \omega x$, and the derivation of order n is $x \mapsto \phi(n, x)$. The least x such that $x = \omega^x$ is known as ϵ_0; the least x such that $x = \phi(x, 0)$ is known as Γ_0.

We have shown that the inductive type T defined by zero and a constructor of type $T \to N \to T \to T$, without the terms that are not in "normal form" , is isomorphic to the set of ordinals less than ϵ_0; in the case of $T \to T \to N \to T \to T$, we get all ordinals less than Γ_0; we have also studied the case with one more T (the first two types were first implemented by Castéran, the last was suggested by Ackermann) [19]

6.2. Homotopy Type Theory

Participants: Yves Bertot [correspondant], Florent Bréhard.

Homotopy Type Theory is a domain born out of the conjuction of type theory, which serves as foundations for proof systems like Coq or Agda, and homotopy theory, and domain of mathematics which is concerned with equivalence classes of objects modulo continuous deformation. In particular, Homotopy Type Theory concentrates on paths (continuous substrate between various objects) and paths between paths: paths between points can be understood as lines, paths between lines can be understood as surfaces.

In particular, paths can be thought has having the same properties as the notion of equality that is usually defined inductively in type theory systems and homotopy type theory goes against the trend started in the 1990s where specialists thought an axiom should be added to express that all paths between paths should be equal. On the contrary, if all paths between paths are not equal, type theory can be used to model homotopy theory and that domain of mathematics because a new area of applications for type theory-based theorem provers.

V. Voevodsky organized a special year at Institute for Advanced Study in Princeton on this topic, and Yves Bertot participated to this special year, during which many experiments were performed, extensions to proof systems were designed, and a book was produced. In particular, Yves Bertot devised an extension of the Coq system with *private types* which makes it possible to simulate a new concept known as *higher inductive types*. On top of this extension, the members of the special year produced a collection of higher inductive types, describing circles, spheres, truncations.

During his internship in the Marelle project, Florent Bréhard studied the equivalence between several presentations of higher-dimension spheres using higher inductive types.

Work on higher inductive types was pursued more precisely by Bruno Barras from Saclay. We expect that the result of this work will supersede the experiments made possible by Yves Bertot's implementation of private types, but the concept of private type may retain applications in other domains.

6.3. Isolation of polynomial roots

Participants: Yves Bertot [correspondant], Julianna Zsidó.

Together with techniques to produce square-free polynomials (polynomials whose roots are all simple), Bernstein polynomials provide a way to decide whether a polynomial has roots in a given interval. Together with a dichotomy procedure, this makes it possible to isolate all the roots of a polynomial, or to show that no root of a given polynomial occur in a given interval. At the end of 2012, Julianna Zsidó started to study this procedure: she showed the properties of the procedure to obtain square-free polynomials and she then formalized a proof for a theorem known as the *theorem of three circles* which plays a rôle in proving that dichotomy will terminate. This work has been published as an article in the *Journal of Automated Reasoning*.

We expect to wrap up all this work by producing easy-to-use tactics to prove properties of polynomial formulas and generalizing it to polynomials in several variables.

During a summer internship, Konstantinos Lentzos worked on the representation of algebraic numbers (which can always be represented as roots of polynomials in a given interval) and the question of finding polynomials for algebraic numbers obtained through simple operations (like addition, multiplication, opposite, and inversion). However, this work was made extremely difficult by the problem of finding morphisms between various fields definable on top of a polynomial ring.

6.4. Properties of the π number

Participants: Yves Bertot [correspondant], Laurence Rideau, Laurent Théry.

As a testbed for the progress of formalized libraries in the domain of calculus, we studied an algorithm to compute π (the circle ratio) using arithmetic-geometric means. This study brought us to extend the libraries with improper integrals, studies of *arcsinh*, variable change in integrals, and error propagation proofs.

We also studied a formal proof of the spigot algorithm designed by Bailey, Borwein, and Plouffe, which is used to compute far digits in the hexadecimal representation of π as a fractional number. This relies on floating point computations and error control, for which we provided a formal proof.

6.5. Formal study of cryptography

Participants: Gilles Barthe [IMDEA Software Institute], François Dupressoir [IMDEA Software Institute], Benjamin Grégoire [correspondant], César Kunz [IMDEA Software Institute], Yassine Lakhnech [Univ. Grenoble 1], Benedikt Schmid [IMDEA Software Institute], Pierre-Yves Strub [IMDEA Software Institute], Santiago Zanella Béguelin [MSR].

The goal of this work is to provide a friendly tool easily usable by cryptographers without knowledge of formal proof assistants. The idea is to use the techniques formally proved in Certycrypt and to call SMT-provers. We provide two differents tools:

- Easycrypt (see http://www.easycrypt.info/) is a toolset for reasoning about relational properties of probabilistic computations with adversarial code. Its main application is the construction and verification of game-based cryptographic proofs. This year, Easycrypt has been fully reimplemented, allowing more modularity in proofs and an interactive prover has been integrated.
- ZooCrypt (see http://www.easycrypt.info/zoocrypt/) is an automated tool for analyzing the security of padding-based public-key encryption schemes (i.e. schemes built from trapdoor permutations and hash functions). ZooCrypt includes an experimental mechanism to generate EasyCrypt proofs of security of analyzed schemes.

This year we published papers concerning formal proofs for properties of elliptic curves, differential privacy, padding-based encryption, and probabilistic relational verification.

6.6. Approximation of Mathematical functions

Participants: Guillaume Hanrot, Érik Martin-Dorel, Micaela Mayero [Université de Paris 13], Ioana Paşca [Université de Nimes], Laurence Rideau, Laurent Théry [correspondant].

In a collaboration supported by ANR project Tamadi, we study the approximation of mathematical functions (exponential and trigonometric functions) using polynomial functions.

This year, we completed the formal verification of our library that computes Taylor Models for the usual mathematical functions of one variable within Coq. A presentation of this work has been done at SYNASC'2013.

The SLZ algorithm checks that there is no hard-to-round floating numbers for a given range in a given floating-point format. It usually consists of a very long computation returning a yes/no answer. Formally proving the implementation of the algorithm is current outside reach since it requires very sophisticated numerical libraries that are currently impossible to verify formally. We have defined a notion of certificate for these computations based on Hensel's lemma and derived an executable checker within Coq that is capable to verify such computations. A publication has been submitted.

6.7. Formal verification in Geometry

Participants: Laurent Fuchs, Laurent Théry [correspondant].

Grassmann-Cayley Algebras are a convenient algebraic way of talking about geometrical concepts. We have further improved our certified Grassmann-Cayley Algebra library to accommodate unbalanced binary trees. A publication has been accepted and will be published in 2014.

6.8. SMT automation for Ssreflect

Participants: Antoine Grospellier, Laurent Théry [correspondant].

The proof of the Feit-Thompson theorem (also known as the odd-order theorem) has been carried on with little use of automation. We have customised the existing connection between Coq and SMT solvers using Why to accomodate Ssreflect specificities. The preliminary results are encouraging.

MEXICO Project-Team

6. New Results

6.1. Diagnosis

- For non-diagnosable discrete event systems, *active* diagnosis aims at synthesizing a partial-observabion based control for the system in order to make it diagnosable. While some solutions had already been proposed for the active diagnosis problem, their complexity remained to be improved. In [40], we solved both the active diagnosability decision problem and the active diagnoser synthesis problem, proving that (1) our procedures are optimal w.r.t. to computational complexity, and (2) the memory required for the active diagnoser produced by the synthesis is minimal. Furthermore, focusing on the minimal delay before detection, we establish that the memory required for any active diagnoser achieving this delay may be highly greater than the previous one. So we refine our construction to build with the same complexity and memory requirement an active diagnoser that realizes a delay bounded by twice the minimal delay. An extension to *probabilistic* systems has been accepted to *FoSSaCS 2014*.
- In [41], we present a methodology for fault diagnosis in concurrent, partially observable systems with additional fairness constraints. In this *weak* diagnosis, one asks whether a concurrent chronicle of observed events allows to determine that a non-observable fault will inevitably occur, sooner or later, on any maximal system run compatible with the observation. The approach builds on strengths and techniques of unfoldings of safe Petri nets, striving to compute a compact prefix of the unfolding that carries sufficient information for the diagnosis algorithm. Our work extends and generalizes the unfolding-based diagnosis approaches by Benveniste et al. as well as Esparza and Kern. Both of these focused mostly on the use of sequential observations, in particular did not exploit the capacity of unfoldings to reveal inevitable occurrences of concurrent or future events studied by Balaguer et al. [19]. Our diagnosis method captures such indirect, revealed dependencies. We develop theoretical foundations and an algorithmic solution to the diagnosis problem, and present a SAT solving method for practical diagnosis with our approach. The algorithms to check diagnosability of concurrent systems are usually performed by local diagnoses of twin plant communicating with each other, directly or through a co- ordinator, and by that means pooling together the observations. Parallel analysis of diagnosability [43] takes advantage of the distribution of the system allowing to decide the diagnosability of the whole system in terms of the diagnosability of smaller systems.

6.2. Testing for Concurrent Systems

6.2.1. Model Based Testing with Labeled Event Structures

In [52], we have developped a complete testing framework for concurrent systems, which included the notions of test suites and test cases. We studied what kind of systems are testable in such a framework, and we have proposed sufficient conditions for obtaining a complete test suite as well as an algorithm to construct a test suite with such properties. However complete test suites are usually infinite in practice. In [44] (and a submitted journal version), we have proposed several testing criteria based on dedicated notions of complete prefixes that selects a manageable test suite together with a coverable criterion that allows to compare them.

6.3. Petri Nets

6.3.1. A Modular Approach for Reusing Formalisms in Verification Tools of Concurrent Systems

Over the past two decades, numerous verification tools have been successfully used for verifying complex concurrent systems, modelled using various formalisms. However, it is still hard to coordinate these tools since they rely on such a large number of formalisms. Having a proper syntactical mechanism to interrelate them through variability would increase the capability of effective integrated formal methods. In [28], we propose a modular approach for defining new formalisms by reusing existing ones and adding new features and/or constraints. Our approach relies on standard XML technologies; their use provides the capability of rapidly and automatically obtaining tools for representing and validating models. It thus enables fast iterations in developing and testing complex formalisms. As a case study, we applied our modular definition approach on families of Petri nets and timed automata.

6.3.2. Computation of summaries using net unfoldings

In [38], we study the following summarization problem: given a parallel composition A = A1 || ... || An of labelled transition systems communicating with the environment through a distinguished component Ai, efficiently compute a summary Si such that E || A and E || Si are trace-equivalent for every environment E. While Si can be computed using elementary automata theory, the resulting algorithm suffers from the state-explosion problem. We present a new, simple but subtle algorithm based on net unfoldings, a partial-order semantics, give experimental results. Our algorithm can also handle divergences and compute weighted summaries with minor modifications.

6.3.3. Complexity Analysis of Continuous Petri Nets

At the end of the eighties, continuous Petri nets were introduced for: (1) alleviating the combinatory explosion triggered by discrete Petri nets and, (2) modelling the behaviour of physical systems whose state is composed of continuous variables. Since then several works have established that the computational complexity of deciding some standard behavioural properties of Petri nets is reduced in this framework. In [39], we first establish the decidability of additional properties like boundedness and reachability set inclusion. We also design new decision procedures for the reachability and lim-reachability problems with a better computational complexity. Finally we provide lower bounds characterising the exact complexity class of the boundedness, the reachability, the deadlock freeness and the liveness problems.

6.3.4. Contextual Merged Processes

In [45], we integrate two compact data structures for representing state spaces of Petri nets: merged processes and contextual prefixes. The resulting data structure, called contextual merged processes (CMP), combines the advantages of the original ones and copes with several important sources of state space explosion: concurrency, sequences of choices, and concurrent read accesses to shared resources. In particular, we demonstrate on a number of benchmarks that CMPs are more compact than either of the original data structures. Moreover, we sketch a polynomial (in the CMP size) encoding into SAT of the model-checking problem for reachability properties.

6.3.5. A Canonical Contraction for Safe Petri Nets

Under maximal semantics, the occurrence of an event a in a concurrent run of an occurrence net may imply the occurrence of other events, not causally related to a, in the same run. In recent works, we have formalized this phenomenon as the *reveals* relation, and used it to obtain a contraction of sets of events called *facets* in the context of occurrence nets. In [36], we extend this idea to propose a canonical contraction of general safe Petri nets into pieces of partial-order behaviour which can be seen as "macro-transitions" since all their events must occur together in maximal semantics. On occurrence nets, our construction coincides with the facets abstraction. Our contraction preserves the maximal semantics in the sense that the maximal processes of the contracted net are in bijection with those of the original net.

6.4. Composition

6.4.1. Specification of Asynchronous Component Systems with Modal I/O-Petri Nets

In collaboration with Professor Rolf Hennicker from LMU and M.H. Møller, a PhD student from Aalborg University, we have studied the asynchronous composition of systems where the internal channels remain observable. In [42], we have modelled such systems by Petri nets enlarged with communication channels, we have defined several channel properties and shown these properties are compositional, and proved their decidability. In TGC 2013 (not yet in HAL), we have extended the previous models with modalities *must* and *may* "à la Larsen" and generalized most of the results in this framework.

6.4.2. Bounding models families for performance evaluation in composite Web services

One challenge of composite Web service architectures is the guarantee of the Quality of Service (QoS). Performance evaluation of these architectures is essential but complex due to synchronizations inside the orchestration of services. In (ADD WHEN IN HAL), we propose methods to automatically derive from the original model a family of bounding models for the composite Web response time. These models allow to find the appropriate trade-off between accuracy of the bounds and the computational complexity. The numerical results show the interest of our approach w.r.t. complexity and accuracy of the response time bounds.

6.5. Stochastic Systems

6.5.1. Simulation-based Verification of HASL (Hybrid Automata Stochastic Logic) Formulas for Stochastic Symmetric Nets

The Hybrid Automata Stochastic Logic (HASL) has been recently defined as a flexible way to express classical performance measures as well as more complex, path-based ones (generically called "HASL formulas"). The considered paths are executions of Generalized Stochastic Petri Nets (GSPN), which are an extension of the basic Petri net formalism to define discrete event stochastic processes. The computation of the HASL formulas for a GSPN model is demanded to the COSMOS tool, that applies simulation techniques to the formula computation. Stochastic Symmetric Nets (SSN) are an high level Petri net formalism, of the colored type, in which tokens can have an identity, and it is well known that colored Petri nets allow one to describe systems in a more compact and parametric form than basic (uncolored) Petri nets. In [27], we propose to extend HASL and COSMOS to support colors, so that performance formulas for SSN can be easily defined and evaluated. This requires a new definition of the logic, to ensure that colors are taken into account in a correct and useful manner, and a significant extension of the COSMOS tool.

6.5.2. Steady-state control problem for Markov decision processes

We address in (ADD CITATION WHEN IN HAL) a control problem for probabilistic models in the setting of Markov decision processes (MDP). We are interested in the steady-state control problem which asks, given an ergodic MDP M and a distribution δ, whether there exists a (history-dependent randomized) policy π ensuring that the steady-state distribution of M under $\imath$ is exactly δ. We first show that stationary randomized policies suffice to achieve a given steady-state distribution. Then we infer that the steady-state control problem is decidable for MDP, and can be represented as a linear program which is solvable in PTIME. This decidability result extends to labeled MDP (LMDP) where the objective is a steady-state distribution on labels carried by the states, and we provide a PSPACE algorithm. We also show that a related steady-state language inclusion problem is decidable in EXPTIME for LMDP. Finally, we prove that if we consider MDP under partial observation (POMDP), the steady-state control problem becomes undecidable.

6.6. Timed Systems

6.6.1. Back in Time Petri Nets

The time progress assumption is at the core of the semantics of real-time formalisms. It is also the major obstacle to the development of partial-order techniques for real-time distributed systems since the events are ordered both by causality and by their occurrence in time. Anyway, extended free choice safe time Petri nets (TPNs) were already identified as a class where partial order semantics behaves well. In [37], we show that, for this class, the time progress assumption can even be dropped (time may go back in case of concurrency), which establishes a nice relation between partial-order semantics and time progress assumption.

6.6.2. Expressiveness of Timed Models

In coopération with Nantes and UPMC, an in-depth study of the expressiveness of time Petri nets was completed [20]. With roughly the same partners, we have extended th ITA (Interrupt Timed Automata) by parametrizing both guards and clock rates while preserving the decidability results (RP 2013, not yet in HAL).

6.7. Weighted Systems

6.7.1. Specification and Verification of Quantitative Properties via Expressions, Logics, and Automata

Alongside boolean properties, automatic verification of *quantitative* properties such as lifespan of an equipment, energy consumption of an application or reliability of a program is gaining importance rapidly. In the thesis [14] and the articles [32], [14] , several weight-enabled formalisms for specification of such properties were examined, including denotational ones such as regular expressions, first-order logic with transitive closure, or temporal logics, as well as more operational ones such as navigating automata, possibly extended with pebbles. A unified framework of graph structures allows to compare these formalisms with respect to expressiveness, using efficient translations from denotational to operational formalisms. Several decidability and complexity results for the algorithmic questions that arise were obtained, depending on the underlying semiring from which weights are chosen, and on the structures (words, trees, ...) considered.

6.8. Dynamic Communicating Systems

6.8.1. Specification and Verification of Dynamic Message-Passing Systems

In [31], we study dynamic communicating automata (DCA), an extension of classical communicating finite-state machines that allows for dynamic creation of processes. The behavior of a DCA can be described as a set of message sequence charts (MSCs). While DCA serve as a model of an implementation, we propose branching high-level MSCs (bHMSCs) on the specification side. Our focus is on the implementability problem: given a bHMSC, can one construct an equivalent DCA? As this problem is undecidable, we introduce the notion of executability, a decidable necessary criterion for implementability. We show that executability of bHMSCs is EXPTIME-complete. We then identify a class of bHMSCs for which executability effectively implies implementability.

6.9. Concurrent Recursive Programs

6.9.1. The Complexity of Model Checking Concurrent Recursive Programs

In [34], we consider the linear-time model checking problem for boolean concurrent programs with recursive procedure calls. While sequential recursive programs are usually modeled as pushdown automata, concurrent recursive programs involve several processes and can be naturally abstracted as pushdown automata with multiple stacks. Their behavior can be understood as words with multiple nesting relations, each relation connecting a procedure call with its corresponding return. To reason about multiply nested words, we consider the class of all temporal logics as defined in the book by Gabbay, Hodkinson, and Reynolds (1994). The unifying feature of these temporal logics is that their modalities are defined in monadic second-order (MSO) logic. In particular, this captures numerous temporal logics over concurrent and/or recursive programs that have been defined so far. Since the general model checking problem is undecidable, we restrict attention to phase bounded executions as proposed by La Torre, Madhusudan, and Parlato (LICS 2007). While the MSO model checking problem in this case is non-elementary, our main result states that the model checking (and satisfiability) problem for all MSO-definable temporal logics is decidable in elementary time. More precisely, it is solvable in (n + 2)-EXPTIME where n is the maximal level of the MSO modalities in the monadic quantifier alternation hierarchy. We complement this result and provide, for each level n, a temporal logic whose model checking problem is n-EXPSPACE-hard.

6.9.2. Model Checking Concurrent Recursive and Communicating Programs via Split-Width

The work described in the following was done by Aiswarya Cyriac in collaboration with Paul Gastin and K. Narayan Kumar, and it is part of Aiswarya Cyriac's PhD thesis, which has recently been defended. It is a generalisation of our CONCUR'12 paper where split-width is introduced to address the decidability of MSO specifications for multi-pushdown systems.

We consider generic systems which incorporate shared-variable communication and communication via channels. We are considering physically distributed machines which communicate via (possibly several) reliable first-in-first-out queues. Each of these machines are capable of running potentially recursive multi-threaded programs. These programs within a machine use shared variable for communication. Such a machine consisting of a set of threads communicating by shared memory can be formally modelled as a multi-pushdown system. Thus we have a network of multi-pushdown systems communicating via FIFO queues. Moreover, these programs may use stacks and queues as data-structures to aid their local computation. We call such a system a system of concurrent processes with data-structures (CPDS).

We introduce and study a new technique called split-width for the under-approximate verification of CPDS. This parameter is based on simple shuffle and merge operations and gives us a divide-conquer-way to prove the bound of languages. When parametrised by a bound on split-width, we obtain decidability for various verification problems. We provide a uniform decision procedure for various verification problems with optimal complexities.

We expose the power of split-width in several ways. We show that our simple algebra is powerful enough to capture any class of CPDS which admits decidability for MSO model checking, and yardstick graph metrics such as tree-width and clique-width. We also show that various restrictions well-studied in the literature for obtaining decidability of reachability for the particular cases of multi-pushdown systems and message passing systems admit a bound on split-width. In fact, we propose generic controllers which subsume many of these cases.

Distributed controller design amounts to designing a controller (which is another CPDS) which, when run sychronously with a system ensures bounded split-width. These controllers are distributed in nature and are independent of the system it is controlling. Thus such a controller respects the privacy of the system (by not reading their states, for instance). Moreover, thanks to split-width such a controlled system offers efficient (in most cases optimal) decision procedures for the verification of the controlled system. We propose a generic approach to define controllable classes of CPDS in terms of quotient graphs, which admit a "suitable" acyclicity restriction. We also give a generic controller for several of the classes definable in this framework. The controllers we propose are sound and complete for the respective class, meaning that they allow all and only the behaviours of this class. Moreover, our technique for proving the bound on split-width of the controlled systems is also generic and systematic, hence may easily extend to generalisations and other classes as well.

The decidability results for the controllable classes proposed in the thesis are new while they capture, as special cases, several restrictions studied in the literature like bounded phase, bounded scope, poly-forest topology etc.

PARSIFAL Project-Team

6. New Results

6.1. Substitution as Proof Compression

Participants: Lutz Straßburger, Novak Novakovic.

In previous work [58] we have shown how the calculus of structures can accommodate Tseitin extension without relying on the cut (or modus ponens). Thus, cut and extension can be studied independently as proof compression mechanisms. Another such proof compression mechanism is substitution. It has been shown by Cook, Reckhow, Krajíček and Pudlák that in the presence of cut, extension and substitution are equally powerful with respect to proof complexity. This year we succeeded in showing that this is also the case in the absence of cut. I.e., we have shown that the cut-free system with extension and the cut-free system with substitution p-simulate each other. This result is presented in [34].

6.2. Herbrand Confluence

Participants: Lutz Straßburger, Stefan Hetzl.

In the result on Herbrand confluence from last year [46], the endsequent of a proof had to be an existential sentence in prenex form. This year we were able to relax this restriction and to extend our result to arbitrary endsequent. This work has been published in [15].

6.3. Nested Sequents for Intuitionistic Modal Logics

Participant: Lutz Straßburger.

We present cut-free deductive systems without labels for the intuitionistic variants of the modal logics obtained by extending IK with a subset of the axioms d, t, b, 4, and 5. For this, we use the formalism of nested sequents, which allows us to give a uniform cut elimination argument for all 15 logic in the intuitionistic S5 cube. This work (published in [25]), is an improvement of the result on intuitionistic modal logic from 2011: the deductive systems the cut elimination proof are much simpler now.

6.4. First efforts at designing proof certificates

Participants: Hichem Chihani, Quentin Heath, Dale Miller, Fabien Renaud.

Work on the ERC Advance Grant ProofCert has progressed along two lines.

Given earlier work within the team [6], [7], there now exists a flexible and well understood concept of focused proof for classical and intuitionistic first-order logics. Chihani, Miller, and Renaud have been working to use that notion of proof as a means of providing flexible definition of *proof evidence* for those two logics. Initial results along those directions have been reported in the [19] and [20]. In those papers, several examples definitions of the semantics of *proof certificates* (formal documents providing the details of some proof evidence) are provided in such a way that a single, simple proof checker can formally elaborate that evidence into a focused sequent calculus. Such an elaboration thus guarantees the soundness of that proof. These papers also describe a "reference proof checker" that has been built with the expectation that its formal correctness can be established. That checker is also able to do bounded *proof reconstruction* as well as allow both deterministic and non-deterministic computation to be mixed with deduction.

Our understanding of focused proofs in the presence of both induction and co-induction (inference rules found in model checkers and most theorem provers) is less well developed. As a result, Miller and Tiu have been studied a simple approach of proof certificate in the setting of model checking in the hope of identifying the relevant proof theory designs that need to be developed. In [33], they showed how tabled deduction in model checking can be used to provide a formal proof certificate for a range of co-inductively defined predicates.

6.5. Combinations of classical and intuitionistic logic

Participant: Dale Miller.

Chuck Liang and Miller have been studying the question of how one can mix intuitionistic and classical logic into a single logic. The initial motivation for considering this problem arose from the concerns raised by the ProofCert project of how best to deal with both classical and intuitionistic logic and their associated proof evidence. Will there need to be two different kinds of checkers and two different kinds of libraries for these two different kinds of logics? Will we be able to mix theorems and proofs in one logic with those in the second logic in rich and useful ways?

One way we have considered answering this question is to actually consider a third logic that combines these other two logics. Our work on such combinations is reported in [16], where a thorough analysis of the semantics and proof theory of such a combination is provides, and in [24], where significant examples of the computational aspects of proofs are explored in detail.

6.6. Formal meta theory of sequent calculus

Participant: Dale Miller.

Keeping with the ProofCert theme of finding global, eternal, and formal mechanisms representing proof evidence, Miller and Pimentel describe in [17] a way in which linear logic can be used to formally specify inference rules for a wide range of proof system in several logics. They were able to show that adequacy of their encodings and to provide sufficient conditions for both cut-elimination and initial-elimination to hold for the resulting proof systems. The fact that these elimination results hold or not is an important characteristic for judging a proof system. Using this work, these important questions can be resulted automatically for a wide range of such proof systems.

6.7. The correctness of program using finite precision

Participants: Ivan Gazeau, Dale Miller.

Programs dealing with real number quantities must live with the fact that such numbers are represented using only finite precision. As such, programs that might be considered correct over the abstract field of infinite precision arithmetic can display chaotic and incorrect behaviors when run on actual computer hardware.

One such problem with finite precision is that programs can “leak” information about values that are intended to be hidden or at least obfuscated as happens in the area differential privacy. In [22], Gazeau, Miller, and Palamidessi illustrated just how such attacks on information hiding can be made and how it is possible to add noise to reported data values in such a way that only appropriate amounts of information leakage occurs.

In his PhD thesis, *Safe Programming in finite precision: Controlling the errors and information leaks* (École Polytechnique, 2013 [11]), Gazeau develops that theme further as well as shows how techniques from rewriting theory can be applied to show that, in some situations, the chaotic behavior of finite precision programs can be expected to converge in acceptable time to acceptable answers.

6.8. Sequent Calculus with Calls to a Decision Procedure

Participants: Mahfuza Farooque, Stéphane Graham-Lengrand.

In the PSI project, a version of the focused sequent calculus (for first-order classical logic) has been designed, which can call external decision procedures. Several results were achieved in 2013 since the last Activity Report:

Firstly, a bug was discovered in the proof of cut-elimination, which was used to prove the logical completeness of the calculus. Fixing the bug required minor changes in the definition of the system, but incurred a major re-development of the meta-theory. Out of this technical work, one idea emerged: in presence of a non-trivial theory, changing the polarity of literals may change the provability of formulas. This was quite unexpected, but it led to interesting issues, such as finding sufficient conditions on polarities to guarantee cut-elimination and logical completeness. An substantial achievement in this research topic was to successfully address such issues, which gave rise to a new version of the report [30].

Secondly, more techniques from automated reasoning were captured as proof-search in this sequent calculus (the incremental construction of proof-trees): besides the SMT-solving algorithm DPLL(T) treated successfully in 2012 (which was written down and published this year in [21]), the techniques of *clause tableaux* and *connection tableaux* were captured this year. This includes in particular a notion of *clause tableaux modulo theories* that C. Tinelli introduced in 2007 [60]. This new range of captured techniques is interesting as clause tableaux are designed to handle quantifiers, which DPLL(T) does not. This gives a new hope to combine the efficiency of SAT-solvers for propositional reasoning with the handling of quantifiers.

6.9. Path Functors in the Category of Small Categories

Participant: François Lamarche.

In [31] François Lamarche gives a detailed description of two path functors in the category of small categories, which he calls $\mathbf{Pe}$ and $\mathbf{P}$, and proves some of their important properties. The second of these is the functor which is used to model the Martin-Löf identity type in [47]; it associates to every small category X an internal category structure whose object of objects is X; one important theorem which is proved in [31] is that the category of internal (co- or contravariant) presheaves on $\mathbf{P}X$ coincides with the category of Grothendieck bifibrations over the base X. Thus, through a trivial use of monadic abstract nonsense, we can say that $\mathbf{P}X$ is the free bifribration over X. The category $\mathbf{P}X$ is obtained by taking the bigger $\mathbf{Pe}X$, which is a little more than just a category, being poset-enriched, and getting rid of the order enrichment by quotienting. $\mathbf{Pe}X$ is a more general kind of bifibration than an ordinary Grothendieck bifibration, and the enrichment is necessary to describe its properties, thus taking us outside of the theory 1-categores.

6.10. Subformula Linking as an Interaction Method

Participant: Kaustuv Chaudhuri.

We showed how to generalize the *calculus of structures*, a *deep inference* formalism, for classical linear logic to a *calculus of linking* [18]. This generalization simplifies the calculus by eliminating most of its inference rules. In its place we add a notion of annotation with *links* and a *link resolution* procedure. We show that this is sound and complete with respect to the usual calculus of structures. The linking calculus is the foundational basis of the *Profound* tool described in 5.1 .

6.11. Recovering Proof Structures in the Sequent Calculus

Participants: Kaustuv Chaudhuri, Stefan Hetzl, Dale Miller.

The *sequent calculus* is often criticized as a proof syntax because it contains a lot of noise. It records the precise minute sequence of operations that was used to construct a proof, even when the order of some proof steps in the sequence is irrelevant and when some of the steps are unnecessary or involve detours. These features lead to several technical problems: for example, cut-elimination in the classical sequent calculus LK, as originally developed by Gentzen, is not confluent, and hence proof composition in LK is not associative. Many people choose to discard the sequent calculus when attempting to design a better proof syntax with the desired properties.

In recent years, there has been a project at Parsifal to recover some of these alternative proof syntaxes by imposing a certain abstraction over sequent proofs. Our technique, pioneered at Parsifal, involves the use of *maximal multi-focusing* which gives a syntactic characterization of those sequent proofs that: (1) have a "don't care" ordering of proof steps where the order does not matter, and (2) groups larger logical steps, called *actions*, into a maximally parallel form where only important orderings of actions are recorded. The earliest example of this technique was in [40], where we showed a class of sequent proofs that were isomorphic to proof nets for multiplicative linear logic. In 2012, we were able to obtain a similar result for first-order classical logic, wherein we defined a class of sequent proofs that are isomorphic to expansion proofs, a generalization of Herbrand disjunctions that is in some sense a minimalistic notion of proof for classical logic. This result was published in a preliminary form at the CSL 2012 conference [39].

In 2013 we published an extended paper on this result in the Journal of Logic and Computation [14]. The major contribution here was a detailed proof of the result that gives a precise account of the proof identifications made by expansion proofs.

PI.R2 Project-Team

5. New Results

5.1. Proof-theoretical and effectful investigations

Participants: Pierre Boutillier, Guillaume Claret, Pierre-Louis Curien, Yann-Régis Gianas, Hugo Herbelin, Guillaume Munch-Maccagnoni, Ludovic Patey, Pierre-Marie Pédrot, Alexis Saurin.

5.1.1. Sequent calculus and computational duality

Categorical semantics.

During his collaboration with Marcelo Fiore and Pierre-Louis Curien, Guillaume Munch-Maccagnoni characterised the polarised evaluation order through a categorical structure where the hypothesis that composition is associative is relaxed. Duploid is the name of the structure, as a reference to Jean-Louis Loday's duplicial algebras. The main result, in the lineage of Führmann's [38] direct-style characterisation of monadic models, is a reflection Adj→Dupl where Dupl is a category of duploids and duploid functors, and Adj is the category of adjunctions and pseudo maps of adjunctions. The result suggests that the various biases in denotational semantics: indirect, call-by-value, call-by-name... are a way of hiding the fact that composition is not always associative. This work was accepted for publication in FoSSaCs 2014 [53].

Pierre-Louis Curien, in connection with his increasing interests in operads and algebraic structures of various kinds, found out that the core syntax of system L (underlying the duality of computation) could be used with profit to describe the wiring structures underlying operads, dioperads, cyclic operads, and more generally Lamarche's structads [48]. He also showed a syntactic equivalence between Munch-Maccagnoni's (pre)duploids and system L syntax. These results were presented in his invited talks at the Loday's Mathematical Legacy workshop in Strasbourg and at the workshop Algebra and Computation in Lyon, in January 2014.

Duality of construction.

Paul Downen and Zena Ariola developed a generalized theory of the sequent calculus for understanding the concepts of evaluation strategy and of data (for example, pairs in ML) and co-data (for example, functions) in programming languages. This theory provides a single framework for user-defined data and co-data types as well as a generalized treatment of evaluation strategies, including call-by-value, call-by-name, and call-by-need, that are given as parameters to the theory. In the end, the framework encompasses the previously known duality of call-by-name and call-by-value in the sequent calculus, both by Curien and Herbelin [3] and Wadler [59] , while also including call-by-need and its dual. Additionally, the framework reveals connections with approaches by Zeilberger [60], Munch-Maccagnoni [6], and Curien and Munch-Maccagnoni [33], for using polarization and focalization to provide deterministic strategies for classical computation with structures and pattern matching. This work will be presented at ESOP 2014 [15].

Luke Maurer and Zena Ariola in collaboration with Daniele Varacca studied the connections between π-calculus encodings of the λ-calculus and similar continuation-passing style (CPS) transformations, extending the connections for call-by-value and call-by-name encodings to include the call-by-need π-calculus encoding as well. This development revealed a better understanding of the computational effect needed in the λ-calculus to model call-by-need evaluation, which better reflects the way that memoization for call-by-need is implemented. The work is going to be submitted to RTA-TLCA.

Constructive interpretation of an involutive negation.

Guillaume Munch-Maccagnoni developped a syntax of delimited control operators that exposes a formulae-as-types correspondence between an involutive negation in classical natural deduction, and the idea that captured contexts, unlike continuations, can be inspected. This decomposes technical artefacts found in call-by-name classical realisability, and simplifies witness extraction from proofs of Σ formulae. This work has been submitted and appears in his PhD thesis [11].

5.1.2. Dependent monads

Guillaume Claret and Yann Régis-Gianas are developping a monadic translation from functional programs with effects to Coq that uses a dependent monad. The aim of this work is to allow to reason about effectful programs directly in Coq.

5.1.3. Linear dependent types

Pierre-Marie Pédrot developped a dependent version of the Dialectica translation, that gives interesting insights into the possibility to design linear dependent types. Indeed, Dialectica can be decomposed as a translation acting on linear types instead of intuitionistic ones.

5.1.4. Delimited continuations, polarity and computational effects

Guillaume Munch-Maccagnoni's polarised decomposition of delimited control calculi appeared in his PhD thesis [11].

5.1.5. Reverse mathematics

Ludovic Patey studied with Laurent Bienvenu and Paul Shafer the deep connections between algorithmic randomness and reverse mathematics by defining formally the ability of computing a solution to a problem by probabilistic means within the framework of reverse mathematics, the No Randomized Algorithm property (NRA). They provided a classification of the whole revese mathematics zoo created by Damir Dzhafarov in terms of having the NRA property or not, answering to some open separation questions.

Ludovic Patey stated two dichotomy theorems about satisfiability problems within reverse mathematics and proved them using clone theory. The corresponding paper is submitted to Computability in Europe 2014. He studied also ramseyan theorems related to the Rainbow Ramsey theorem and provided characterizations in terms of diagonally non-computable functions, algorithmic randomness, and related it to the Erdös Moser theorem and Thin set theorem.

5.1.6. Gödel's functional interpretation

Pierre-Marie Pédrot showed that the Dialectica translation could be explained in terms of the Krivine abstract machine, in a way similar to the usual presentation of classical realizability. This opens the door to a better understanding of related translations, as well as adding semi-classical effect into PTS.

5.1.7. Logical foundations of call-by-need evaluation

Alexis Saurin and Pierre-Marie Pédrot developed a structured reconstruction of call-by-need based on linear head reduction which arose in the context of linear logic. This opens new directions both to extend call-by-need to control and to apply linear logic proof-theory (and particularly proof-nets) to call-by-need evaluation.

5.1.8. Streams and classical logic

Alexis Saurin and Fanny He have been working on transfinite term rewriting in order to model stream calculi and their connections with lambda-calculi for classical logic.

Jaime Gaspar identified the eight simplest variants (some already known) of the Kuroda negative translation that translate classical logic into minimal logic.

5.2. Type theory and the foundations of Coq

Participants: Pierre Boutillier, Pierre-Louis Curien, Hugo Herbelin, Pierre-Marie Pédrot, Yann Régis-Gianas, Alexis Saurin, Matthieu Sozeau.

5.2.1. Substitutions and isomorphisms

Pierre-Louis Curien completed his joint work with Richard Garner and Martin Hofmann on relating syntax unstrictification through coercions with model strictification (cf. πr^2 report 2012), adding a careful treatment of identity types. The corresponding paper was accepted for publication in the TCS special issue for Glynn Winskel's anniversary.

5.2.2. Homotopy type theory

Hugo Herbelin, Matthieu Sozeau and Pierre-Louis Curien participated to the univalent foundations program. A collaborative book [18] on the results of this program has been published.

5.2.3. Models of type theory

Simplicial sets and their extensions as Kan complexes can serve as models of homotopy type theory. Hugo Herbelin developed a concrete type-theoretic formalisation of semi-simplicial sets following ideas from Steve Awodey, Peter LeFanu Lumsdaine and other researchers both at Carnegie-Mellon University and at the Institute of Advanced Study. This has been accepted for publication in a special issue of MSCS on homotopy type theory [22].

The technique he used generalises to provide type-theoretic constructions for arbitrary presheaves on Reedy categories, thus including simplicial sets. In particular, this provides with a formulation of simplicial sets where degeneracies are decidable, which is not the case with the definition as a presheaf.

Hugo Herbelin also investigated hybrid constructive definitions of simplicial sets where face maps are axiomatised but degeneracies are built. Again, this provides with a formulation of simplicial sets where it is decidable whether a given simplex is degenerate or not.

5.2.4. Internalizing the setoid model of type theory

As an example use of the new polymorphic universe extension of Coq, Matthieu Sozeau developed together with Nicolas Tabareau (Inria Ascola team, École des mines Nantes) a complete groupoid model of type theory, following the seminal work of Hofmann and Streicher. A preliminary paper presenting a partial generalization of this model to 2-goupoids was written and will be resubmitted [23].

A completed version of this model has since been formalized and will be submitted shortly. This model showcases the use of the polymorphic universes: in the course of its formalization we uncovered hidden assumptions in the interpretation of substitution and sigma types in the original presentation thanks to the universe system.

5.2.5. Proof irrelevance, eta-rules

Matthieu Sozeau finished his implementation of a proof-irrelevant system but did not publish it. Indeed, the homotopy type theory interpretation suggests new ways to introduce proof-irrelevance using bracket types that seem to significantly depart from the syntactic treatment developped by Werner and himself. An investigation of the relationship between the presentation of the calculus of inductive constructions given by Hugo Herbelin and Arnaud Spiwack in [44] which includes the bracket construction and the aforementioned syntactic version will be part of a master's internship supervised by Matthieu Sozeau in 2014.

5.3. Homotopy of rewriting systems

Participants: Cyrille Chenavier, Pierre-Louis Curien, Yves Guiraud, Maxime Lucas, Philippe Malbos.

5.3.1. The homotopical completion-reduction procedure

In [39], Stéphane Gaussent (Institut Camille Jordan), Yves Guiraud and Philippe Malbos have introduced the homotopical completion-reduction procedure as a higher-dimensional rewriting method to compute coherent presentations of monoids. The results of this procedure on Artin monoids of spherical type have been implemented by Yves Guiraud in a Python library, available on his webpage. The procedure is currently improved towards the explicit computation of full polygraphic resolutions of Artin monoids to provide a purely algebraic and constructive account of well-known geometric objects, such as Caylay graphs and Salvetti complexes.

In [16], Yves Guiraud, Philippe Malbos and Samuel Mimram (CEA Saclay) have further investigated the homotopical completion-reduction procedure, extended with the adjunction/elimination of redundant generators, with successful application to two new classes of monoids: the plactic and the Chinese monoids. This work has been implemented by Samuel Mimram and Yves Guiraud into a prototype, that can be tested at http://www.pps.univ-paris-diderot.fr/~smimram/rewr, and has been presented to RTA 2013 by Philippe Malbos, where it has received the best paper award.

5.3.2. New methods for the computation of coherent presentations

During his M2 internship, Maxime Lucas, supervised by Yves Guiraud, has improved the rewriting method used in [43] for the computation of homotopy bases of monoids and categories. This allows a more effective computation in several cases, based on the notion of Anick chain [25] instead of the broader notion of critical branching. Maxime Lucas has now started a PhD thesis, supervised by Yves Guiraud and Pierre-Louis Curien, and currently investigates the use of Garside-like structures [35] to further improve the computation of coherent presentations for higher-dimensional categories.

5.3.3. Higher-dimensional linear rewriting

Cyrille Chenavier, Pierre-Louis Curien, Yves Guiraud and Philippe Malbos investigate with Eric Hoffbeck (LAGA, Université Paris 13) and Samuel Mimram (CEA Saclay) the links between set-theoretic rewriting theory and the computational methods known in symbolic algebra, such as Gröbner bases [28]. This interaction is supported by the Focal project of the IDEX Sorbonne Paris Cité. Yves Guiraud, Eric Hoffbeck and Philippe Malbos are currently working on an improvement, based on the homotopical completion-reduction procedure, of the methods known in algebra to compute homological invariants of algebras and operads. Cyrille Chenavier has started a PhD thesis, supervised by Yves Guiraud and Philippe Malbos, to use Berger's theory of reduction operators [27] to design new methods for the study of rewriting systems.

5.3.4. Homotopical and homological finiteness conditions

Yves Guiraud and Philippe Malbos have written a comprehensive introduction [21] on the links between higher-dimensional rewriting, the homotopical finiteness condition "finite derivation type" and the homological finiteness condition "FP_3", from the point of view of higher categories and polygraphs. The purpose of this work is to provide an introduction to the field, formulated in a contemporary language, and with new, more formal proofs of classical results.

In [19], Yves Guiraud and Philippe Malbos have introduced a notion of identities among relations for higher categories presented by polygraphs. This notion is well-known in combinatorial group theory, where it is linked to the explicit computation of homological invariants and of formal representations of groups as crossed complexes. The main result of [19] is a procedure based on higher rewriting to compute generators of the identities among relations. They have related the facts that the natural system of identities among relations is finitely generated and that the higher category has finite derivation type (a homotopical finiteness condition introduced in [43] for higher categories after Squier's work for monoids [57]).

5.4. Coq as a functional programming language

Participants: Pierre Boutillier, Guillaume Claret, Lourdes Del Carmen González Huesca, Hugo Herbelin, Pierre Letouzey, Matthias Puech, Yann Régis-Gianas, Matthieu Sozeau.

5.4.1. Type classes and libraries

Type Classes are heavily used in the HoTT/Coq library (http://github.com/HoTT/coq) developed by the Univalent Foundations program at the IAS, to which Matthieu Sozeau participated.

5.4.2. Dependent pattern-matching

How to encode structurally dependent pattern matching into case analysis by hand has been described by Jean François Monin in [52]. Pierre Boutillier, with the help of Thomas Braibant (GALLIUM team), has mechanized this process and exhibited a missing part to make it scale. These are the main results presented in Pierre Boutillier's forthcoming thesis.

5.4.3. Incrementality in proof languages

Lourdes González and Yann Régis-Gianas studied incremental computing and self-adjusting computation [24] as a starting point to develop an applicative notion of change over data structures, to be applied to lambda-terms. They formalized in Coq a notion of derivative of an inductive function to define how to compute a new result from an input that has changed, this is done by using the derivative of the function and the difference on inputs and old outputs. They are working out a technique that allows a specification of functions using derivatives and old inputs and outputs including a cost analysis of the benefits of reusing previous computations.

5.4.4. Lightweight proof-by-reflection

In collaboration with Beta Ziliani (MPI), In the context of the ANR project Paral-ITP, Lourdes del Carmen González Huesca, Guillaume Claret and Yann Régis-Gianas developed a new technique for proof-by-reflection based on a notion of *a posteriori* simulation of effectful computations in Coq. This work has been presented at ITP 2013 ([14]).

SUMO Team

6. New Results

6.1. Model expressivity and quantitative verification

6.1.1. Diagnosis from scenarios

Participants: Loïc Hélouët, Blaise Genest, Hervé Marchand.

Diagnosis of a system consists in providing explanations to a supervisor from a partial observation of the system and a model of possible executions. This year, we have extended results on diagnosis algorithm from scenarios. Systems are modeled using High-level Message Sequence Charts (HMSCs), and the diagnosis is given as a new HMSC, which behaviors are all explanations of the partial observation. The results published this year are first an offline centralized diagnosis algorithm (a single process in a network collects an observation, and emits a diagnosis) that has then been extended to a decentralized version of this algorithm. This allows us to give a complete diagnosis framework for infinite state systems, with a strong emphasis on concurrency and causal ordering in behaviors. HMSC-based diagnosis showed nice properties w.r.t. compositionality. We have also considered solutions for online diagnosis from scenarios, but came to the conclusion that online solutions are memory consuming, and need too many restrictions to run with finite memory.

The last contribution of this work is an application of diagnosis techniques to anomaly detection, that is a comparison of observation of the system with a model of usual behaviors to detect security attacks. This work is already available online in [25], and will soon be published.

6.1.2. Probabilistic model checking

Participants: Nathalie Bertrand, Blaise Genest, Paulin Fournier.

In [20], 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 symbolics, we proved that the language of trajectories of distribution (one for each initial distribution) is not regular in general, even with 3 states. We then proposed a parametrized algorithm which approximate what happens to infinity, such that each symbolic block in the approximate language is at most ϵ away from the concrete distribution.

With the objective of model checking infinite state probabilistic systems, we proved a general finite-time convergence theorem for fixpoint expressions over a well-quasi-ordered set [22]. This has immediate applications for the verification of well-structured systems, where a main issue is the computability of fixpoint expressions, and in particular for game-theoretical properties and probabilistic systems where nesting and alternation of least and greatest fixpoints are common [35].

Parameterized verification aims at validating a system's model irrespective of the value of a parameter. In [34] we introduced a model for networks of an arbitrary number of probabilistic timed processes, communicating by broadcasting. This model is suitable for distributed protocols, and can be applied to wireless sensor networks or peer-to-peer applications. The number of processes is unknown and either is constant (static case), or evolves over time through random disappearances and creations (dynamic case). On the one hand, most parameterized verification problems turn out to be undecidable in the static case (even for untimed processes). On the other hand, we prove their decidability in the dynamic case.

6.1.3. Distributed timed systems

Participants: Nathalie Bertrand, Amélie Stainer.

We study the reachability problem for communicating timed processes, both in discrete and dense time. Our model comprises automata with local timing constraints communicating over unbounded FIFO channels. Each automaton can only access its set of local clocks; all clocks evolve at the same rate. Our main contribution is a complete characterization of decidable and undecidable communication topologies, for both discrete and dense time. We also obtain complexity results, by showing that communicating timed processes are at least as hard as Petri nets; in the discrete time, we also show equivalence with Petri nets. Our results follow from mutual topology-preserving reductions between timed automata and (untimed) counter automata. To account for urgency of receptions, we also investigate the case where processes can test emptiness of channels. This resut is published in [39] and is a part of Amélie Stainer's PhD manuscript [18]. It also constitutes a contribution to ANR VACSIM.

We also studied a model for distributed systems composed of stochastic and timed processes that interact via broadcasting. For these networks of stochastic timed automata (NSTA), we provided a precise performance evaluation algorithm, without resorting to simulation techniques. The idea is to characterize the general state space Markov chain through transient stochastic state classes that represent the system's state after each action. This yields an algorithmic approach to the transient analysis of NSTA models, with fairly general termination conditions [32].

6.2. Management of large distributed systems

6.2.1. Test generation from Recursive Tile Systems

Participants: Sébastien Chédor, Thierry Jéron, Christophe Morvan.

We explore the generation of conformance test cases for Recursive Tile Systems (RTSs) in the framework of the classical ioco testing theory. The RTS model allows the description of reactive systems with recursion, and is very similar to other models like Pushdown Automata, Hyperedge Replacement Grammars or Recursive State Machines. Test generation for this kind of models is seldom explored in the literature. We first propose an off-line test generation algorithm for Weighted RTSs, a determinizable sub-class of RTSs, and second, an on-line test generation algorithm for the full RTS model. Both algorithms use test purposes to guide test selection through targeted behaviours. Additionally, essential properties relating verdicts produced by generated test cases on an implementation with both the conformance with respect to its specification, and the precision with respect to a test purpose, are proved. This work is published in [51], and a journal version will appear in 2014. It is also a part of Sébastien Chédor's PhD manuscript.

6.2.2. Distributed control

Participants: Blaise Genest, Hervé Marchand.

We focused this year on the control of distributed systems modeled as *asynchronous automata*, that is asynchronous network of automata communicating through peer to peer synchronizations. First, we considered the case where all events are controllable, and the objective is to accept exactly a given language. Here, a famous result is the Zielonka theorem [62], stating that every regular language closed under commutation can be turned into an asynchronous automaton. However, the construction is plagued with deadends and final state of the network are decided by a global controller monitoring every process at the same time and perfectly, which is unrealistic and defeat the distribution idea. This year, we characterized the languages which can be controlled realistically (no deadends, local final states and local decision on each process), and give algorithms to obtain the associated distributed machines in [30]. The case where some events are uncontrollable is reputed very difficult. We made a progress this year in [42], showing that we can decide whether a reachability objective can be ensured, granted that the communication between the processes follow a tree: siblings can not communicate directly together, they need to go through their common parent.

In [27], we consider an alternative model for the control of distributed systems; the aim is to build local controllers that restrict the behavior of a distributed system in order to satisfy a global state avoidance property. We model distributed systems as communicating finite state machines with reliable unbounded FIFO queues between subsystems. Local controllers can only observe the behavior of their proper subsystem and do not see the queue contents. To refine their control policy, controllers can use the FIFO queues to communicate by piggy-backing extra information (some timestamps and their state estimates) to the messages sent by the subsystems. We provide an algorithm that computes, for each local subsystem (and thus for each controller), during the execution of the system, an estimate of the current global state of the distributed system. We then define a synthesis algorithm to compute local controllers. Our method relies on the computation of (co-)reachable states. Since the reachability problem is undecidable in our model, we use abstract interpretation techniques to obtain overapproximations of (co-)reachable states. Similarly, in [46], we have been interested in the control of distributed systems with synchronous communications (called decentralized Discrete Event Systems). We introduced a novel architecture that extends the class of problems that can be solved in decentralized DES control in the absence of communication. In this architecture, unlike previous architectures that use either conjunction or disjunction to fuse local control decisions, the fusion rule is exclusive or. We characterized the new architecture, where controllers take a single decision, with respect to the recently-proposed multi-decision framework of Chakib and Khoumsi. Unlike previous architectures, parity-based controllers cannot predetermine their local control decision based solely on their local observations. Instead, the local control decisions are calculated a priori.

6.2.3. *Enforcement of timed and security properties*

Participants: Thierry Jéron, Hervé Marchand, Srinivas Pinisetty.

Runtime enforcement is a verification/validation technique aiming at correcting (possibly incorrect) executions of a system of interest. This year, we first consider enforcement monitoring for systems with timing specifications (modeled as timed automata). We consider runtime enforcement of any regular timed property specified by a timed automaton [45]. To ease their design and their correctness-proof, enforcement mechanisms are described at several levels: enforcement functions that specify the input-output behavior, constraints that should be satisfied by such functions, enforcement monitors that implement an enforcement function as a transition system, and enforcement algorithms that describe the implementation of enforcement monitors. The feasibility of enforcement monitoring for timed properties is validated by prototyping the synthesis of enforcement monitors. This work is also a contribution to ANR Vacsim. In [41], we studied an alternative enforcement problem of security properties, namely, the enforcement of K-step opacity at runtime. In K-step opacity, the knowledge of the secret is of interest to the attacker within K steps after the secret occurs and becomes obsolete afterwards. We introduce the mechanism of runtime enforcer that is placed between the output of the system and the attacker and enforces opacity using delays. If an output event from the system violates K-step opacity, the enforcer stores the event in the memory, for the minimal number of system steps until the secret is no longer interesting to the attacker (or, K-step opacity holds again)

6.2.4. *Discrete control of computing systems administration*

Participants: Hervé Marchand, Nicolas Berthier.

We address the problem of using Discrete Controller Synthesis for the administration of Computing Systems, following an approach supported by a programming language [24]. We present a mixed imperative/declarative programming language, where declarative contracts are enforced upon imperatively described behaviors. Its compilation is based on the notion of supervisory control of discrete event systems. More precisely, our language can serve programming closed-loop adaptation controllers, enabling flexible execution of functionalities w.r.t. changing resource and environment conditions. DCS is integrated into a1 programming language compiler, which facilitates its use by users and programmers, performing executable code generation. The tool is concretely built upon the basis of a reactive programming language compiler, where the nodes describe behaviors that can be modeled in terms of transition systems. Our compiler integrates this with a DCS tool, making it a new environment for formal methods. We apply our method to the problem of coordinating several administration loops in a data center (number of servers, repair, and local processor frequencies) [40].

We formulate this problem as an invariance controller synthesis problem. We are currently working on an extension of the controller synthesis tool so that it can handle the use of numerical variables in order to model both the system and the properties to be ensured by control.

6.2.5. Distributed 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 consistent local plans, one per component, such that their combination forms a global plan of minimal cost.

Our previous solutions to this problem modeled components as weighted automata. In collaboration with Loïg Jezequel (TU Munich) and Victor Khomenko (Univ. of Newcastle), we have extended this approach to the case of components modeled as safe Petri nets [44]. 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.

6.2.6. Diagnosis based on self-modeling

Participants: Éric Fabre, Carole Hounkonnou.

Model-based approaches have been proved to provide the best results for fault diagnosis in telecommunication networks, with various kinds of models. They suffer however from several difficulties: one has to build a model adequate to the supervised network (and possibly adapt it as the network evolves), one has to find the correct abstraction level for this model, and one has to deal with size issues of such models. In Carole Hounkonnou's thesis [15], we have proposed an approach that addresses these three limitations, under the generic name of self-modeling. It consists modeling a network in a generic manner, through its building rules. The actual instance one has to manage is then discovered on the fly, when some malfunction explanation request is triggered. Starting from the identified malfunction, the network model instance is discovered/revealed progressively, as requested by the needs of the diagnosis procedure. The latter progressively extends a Bayesian network model of the network, in order to collect more information and identify the malfunction rootcause. The model extension is guided by an information theory criterion: it seeks access to the new observations that are be the most informative (on the average) given previous observations taken into account. This approach allows to deal with potentially large models, as the supervised system needs not be entirely modeled before the diagnosis starts. We are currently working on the extension of this setting to model refinement, and to a framework of dynamic systems rather than static systems.

6.2.7. Graceful restart methods for link state routing protocols

Participants: Éric Fabre, Carole Hounkonnou.

Link state routing protocols are ubiquitous in the internet. OSPF (Open Shortest Path First) is one of them within an Autonomous System. In collaboration with Alcatel-Lucent, we have proposed an extension of graceful restart procedures, that allow to shut down the control plane of routers while maintaining the data plane active, and thus the packet forwarding activity. A drawback of existing procedures was that frozen routers had to be removed from the network as soon as topology evolved. We have shown that this pessimistic precaution could be damageable to the network and was not necessary [43]. Frozen routers may still be useful, even if they do not forward packets in an optimal manner. And even if they create routing loops, the latter can be easily detected, and optimally patched, which is often more efficient than declaring these routers as dead. Experiments on classical topologies of the topology zoo, as well as on random topologies, have confirmed these results.

6.3. Data driven systems

6.3.1. Web services

Participants: Blaise Genest, Loïc Hélouët.

This year, we considered transactional properties (ACID) for web services. In particular, we focused on the atomicity (A of ACID) property, obtained in case of a failure inside an atomic block through compensation of the executed actions of the block. To do so, logs need to be kept. We were interested in maintaining the maximal amount of privacy. We proposed modular algorithms [23] which maintain privacy between modules, with minimal information shared among modules, both in the logging and the compensation phases. Furthermore, each module logs a small number of information, such that the sum of all actions logged is guaranteed minimal. Last, modularity allows fast algorithms, as they need to consider only what happens in the module itself, and not the exact structure of its parent module nor of its sub-modules.

We also have extended the *session system* model originally proposed in [55]. We have deisgned a mode for Web-based systems that allows to describe systems running an arbitrary number of transactions over an arbitrary number of agents. For these systems, syntactic restrictions allow to decide coverability properties, and then more elaborated business rules, such as conflict of interest (the fact that a participant to a system can be involved in two exclusive services), or the Chinese Wall Property (that prevents users of a system to use benefits or information right they may have obtained from a privileged role at later instant of any execution of the system. These results were obtained with M. Mukund and S. Akshay within the context of the DISTOL associated team, and should lead to a publication next year.

6.3.2. Implementation of scenarios

Participants: Loïc Hélouët, Rouwaida Abdallah.

We have revisited the problem of program synthesis from specifications described by High-level Message Sequence Charts. The main objective is to obtain a distributed implementation (for instance described with communicating automata) from a global specification given as High-level MSCS. In the general case, synthesis by a simple projection on each component of the system allows more behaviors in the implementation than in the specification. The differences arise from loss of ordering among messages, but we have shown that for a subclass of HMSCs (the *local HMSCs*) behaviors can be preserved by addition of communication controllers, that intercept messages to add stamping information before resending them, and deliver messages to processes in the order described by the specification. This work was published in [19].

The second aspect of our work on scenarios implementability has considered implementation of requirements expressed as non-local HMSCs. We have proposed a new technique to transform an arbitrary HMSC specification into a local HMSC, hence allowing implementation. This transformation can be automated as a constraint optimization problem, and the impact of modifications brought to the original specification minimized w.r.t. a cost function. The approach was evaluated on a large number of randomly generated HMSCs, and the results show an average runtime of a few seconds, which demonstrates applicability of the technique. These results were published in [28]. Both results mentionned in this sections are part of the PhD thesis of Rouwaida Abdallah, defended this year [14].

6.3.3. Attribute grammars

Participant: Éric Badouel.

Evaluation of attributes w.r.t. an attribute grammar can be obtained by inductively computing a function expressing the dependencies of the synthesized attributes on inherited attributes. This higher-order functional approach to attribute evaluation can straightforwardly be implemented in a higher-order lazy functional language like Haskell. The resulting evaluation functions are, however, not easily amenable to optimization when we want to compose two attribute grammars. In [21], we present an alternative first-order functional interpretation of attribute grammars where the input tree is replaced by an extended cyclic tree each node of which is aware of its context viewed as an additional child tree. These cyclic representations of zippers (trees with their context) are natural generalizations of doubly-linked lists to trees over an arbitrary signature.

Then we show that, up to that representation, descriptional composition of attribute grammars reduces to the composition of tree transducers.

TOCCATA Team

6. New Results

6.1. Deductive Verification

- F. Bobot, J.-C. Filliâtre, C. Marché, G. Melquiond, and A. Paskevich have presented the proof session mechanism of *Why3* at VSTTE 2013 [23]. It is a technique to maintain a proof session against modification of verification conditions. It was successfully used in developing more than a hundred verified programs and in keeping them up to date along the evolution of *Why3* and its standard library. It also helps out with changes in the environment, *e.g.* prover upgrades.
- M. Clochard, C. Marché, and A. Paskevich developed a general setting for developing programs involving binders, using Why3. This approach was successfully validated on two case studies: a verified implementation of untyped lambda-calculus and a verified tableaux-based theorem prover. This work will be presented at the PLPV conference in January 2014 [29]
- M. Clochard published at the POPL conference a paper presenting a work done during an internship at Rice University (Houston, TX, USA) with S. Chaudhuri and A. Solar-Lezama [28]. It is a new technique for parameter synthesis under boolean and quantitative objectives. The input to the technique is a "sketch" — a program with missing numerical parameters — and a probabilistic assumption about the program's inputs. The goal is to automatically synthesize values for the parameters such that the resulting program satisfies: (1) a boolean specification, which states that the program must meet certain assertions, and (2) a quantitative specification, which assigns a real valued rating to every program and which the synthesizer is expected to optimize.
- J.-C. Filliâtre, L. Gondelman, and A. Paskevich have formalized the notion of ghost code implemented in *Why3*, in a paper *The Spirit of Ghost Code* [49] to be submitted. This is an outcome of L. Gondelman's M2 internship (spring/summer 2013).
- In 2013, two public releases of *Why3* were launched, version 0.81 in March and version 0.82 in December [42]. A first important evolution relies on significant efficiency improvements both in terms of execution speed and of memory usage. The second major evolution is the support for many new provers, including interactive provers PVS 6 (used at NASA) and Isabelle2013-2 (planned to be used in the context of Ada program via Spark), and automated ones: CVC4, Mathematica, Metitarski, Metis, Beagle, Princess, and Yices2. The design of the programming language of *Why3*(WhyML) was presented during a tool demonstration at the ESOP conference [33].

6.2. Floating-Point and Numerical Programs

- S. Boldo, F. Clément, J.-C. Filliâtre, M. Mayero, G. Melquiond, and P. Weis, finished the formal proof of a numerical analysis program: the second order centered finite difference scheme for the one-dimensional acoustic wave [15].
- S. Boldo developed a formal proof of an algorithm for computing the area of a triangle, an improvement of its error bound and new investigations in case of underflow [25].
- S. Boldo, J.-H. Jourdan, X. Leroy, and G. Melquiond, extended CompCert to get the first formally verified compiler that provably preserves the semantics of floating-point programs [26].
- S. Boldo and G. Melquiond wrote a chapter of the book [38] that describes the current state of the Mathematics/Computer science research in France.
- C. Lelay worked on formalizing power series for the Coq proof assistant [35].

- Most 18-year old French students pass an exam called Baccalaureate which ends the high school and is required for attending the university. The idea was to try our Coq library Coquelicot on the 2013 mathematics test of the scientific Baccalaureate. C. Lelay went to the "Parc de Vilgénis" high school in Massy, France and took the 2013 test at the same time as the students, but had to formally prove the answers. There was therefore no possible cheating: the Coq library was already developed and it was tested as is during the four hours of the test. This experiment shows that Coquelicot is able to cope with basic real analysis: it has the necessary definitions and lemmas, and its usability and efficiency have been demonstrated in a test with a limited time [45] (see also https://www.lri.fr/~lelay/).
- D. Ishii and G. Melquiond applied methods of deductive program verification to ensure the safety of hybrid automata [34].
- É. Martin-Dorel, G. Hanrot, M. Mayero, L. Théry, showed how to generate and formally check certificates in the Coq proof assistant to solve myriads of instances of the Integer Small Value Problem (ISValP). This problem is directly related to solving the Table Maker's Dilemma with hardest-to-round computations [50]. A new version of the formalized library has been released (http://tamadi.gforge.inria.fr/CoqHensel/).
- É. Martin-Dorel, G. Melquiond, and J.-M. Muller, studied issues related to double rounding in the implementation of error-free transformations [16].

6.3. Automated Reasoning

- C. Dross, S. Conchon, J. Kanig, and A. Paskevich have proposed a new approach for handling quantified formulas in SMT solvers. Their framework is based on the notion of instantiation patterns, also known as triggers, that suggest instances which are more likely to be useful in proof search. This framework has been implemented in the Alt-Ergo SMT solver [48].
- S. Conchon, A. Goel, S. Krstic, A. Mebsout, and F. Zaïdi have designed a new model checking algorithm that is able to infer invariants strong enough to prove complex parameterized cache-coherence protocols [30].
- S. Conchon, A. Mebsout, and F. Zaïdi have presented a new SMT library called Alt-Ergo-Zero. This library is tightly integrated to the backward reachability algorithm of the Cubicle model checker [31].
- S. Conchon, M. Iguernelala, and A. Mebsout have designed a collaborative framework for reasoning modulo simple properties of non-linear arithmetic. This framework has been implemented in the Alt-Ergo SMT solver [47].
- J. C. Blanchette and A. Paskevich designed an extension to the TPTP TFF (Typed First-order Form) format of theorem proving problems to support rank-1 polymorphic types (also known as ML-style parametric polymorphism). This extension, named TFF1, was incorporated in the TPTP standard and was presented at the CADE-24 conference [22].

6.4. Certification of Languages, Tools and Systems

- A. Tafat and C. Marché developed a certified VC generator using Why3. The challenge was to formalize the operational semantics of an imperative language, and a corresponding weakest precondition calculus, without the possibility to use *Coq* advanced features such as dependent types nor higher-order functions. The classical issues with local bindings, names and substitutions were solved by identifying appropriate lemmas. It was shown that Why3 can offer a very significantly higher amount of proof automation compared to *Coq* [36]
- A. Charguéraud, together with the other members of the *JsCert* team have developed this year the first complete formalization of the semantics of the JavaScript programming language. This project is joint work with Philippa Gardner, Sergio Maffeis, Gareth Smith, Daniele Filaretti and Daiva Naudziuniene from Imperial College, and Alan Schmitt and Martin Bodin from Inria Rennes – Bretagne Atlantique (see http://jscert.org).

The formalization consists of a set of inductive rules translating the prose from the *ECMAScript Language Specification, version 5*. These rules can be used to formally reason about program behaviors or to establish the correctness of program transformations. In addition to the inductive rules, a reference interpreter has been proved correct. This interpreter may be used to run actual JavaScript program following the rules of the formal semantics. It has been used in particular to validate the formal semantics against official JavaScript test suites.

The formalization of JavaScript has been published at POPL 2014 [24]. A key ingredient in this formalization is the use of the *pretty-big-step semantics*. This technique allows for representing evaluation rules in big-step style without suffering from a duplication of several premises across different rules. The pretty-big-step technique is described in a paper published by A. Charguéraud at ESOP 2013 [27].

- É. Contejean, together with V. Benzaken and their PhD student S. Dumbrava, have proposed a *Coq* formalization of the relational data model which underlies relational database systems [21]. Proposing such a formalization is the first, *essential* step, that will allow to *prove* that existing systems conform to their specifications and to *verify* both production implementations of database systems and database-backed applications. More precisely, they present and formalize the data definition part of the model including integrity constraints, attributes, tuples, relations, schemas and integrity constraints (including the so-called Armstrong's system and the chase). They model two different query language formalisms: relational algebra and conjunctive queries. The former is the basis of the SQL commercial query language and the latter underlies graphical languages, such as Microsoft Access or Query By Example (QBE). They also present logical query optimization and prove the main "database theorems": algebraic equivalences, the homomorphism theorem and conjunctive query minimization.

6.5. Miscellaneous

- R. El Sibaie and J.-C. Filliâtre have developed *Combine*, an OCaml library for combinatorics. It provides two different solutions to the exact matrix cover problem: Knuth's dancing links and ZDDs, a variant of binary decision diagrams [32].

VERIDIS Project-Team

6. New Results

6.1. Automated and Interactive Theorem Proving

6.1.1. Using symmetries in SMT

Participants: David Déharbe, Pascal Fontaine, Stephan Merz.

Joint work with Carlos Areces, Raúl Fervari, Guillaume Hoffmann, and Ezequiel Orbe at Universidad Nacional de Córdoba (see also section 8.2).

Methods exploiting problem symmetries have been very successful in several areas including constraint programming and SAT solving. We proposed similar techniques for enhancing the performance of SMT-solvers by detecting symmetries in the input formulas and using them to prune the search space of the SMT algorithm. These techniques are based on the concept of (syntactic) invariance by permutation of symbols. In 2011, we presented a technique restricted to constants but which exhibited impressive results for some categories of formulas [4]; this technique was quickly implemented in major SMT solvers, including CVC4 and Z3.

In 2013, we proposed, together with our colleagues at the University of Córdoba, Argentina, a more general approach to detect symmetries in an SMT context. These techniques are based on graph isomorphisms, and the Schreier-Sims algorithm for improving the presentation of the symmetries. This work was published at the SMT workshop 2013 [21].

6.1.2. Computing minimal models (prime implicants)

Participants: David Déharbe, Pascal Fontaine.

Joint work with Daniel Le Berre and Bertrand Mazure from the CRIL laboratory in Lens, France.

Model checking and counter-example guided abstraction refinement are examples of applications of SAT solving that require the production of models for satisfiable formulas. Instead of giving a truth value to every variable, it is usually preferable to provide an implicant, i.e. a partial assignment of the variables such that every full extension is a model for the formula. An implicant is *prime* if every assignment is necessary. Since prime implicants contain no literal irrelevant for the satisfiability of the formula, they are considered as highly refined information.

In 2013, we proposed a novel algorithm that uses data structures found in modern CDCL SAT solvers for efficiently computing prime implicants starting from an existing model. The original aspects are (1) the algorithm is based on watched literals and a form of propagation of required literals, adapted to CDCL solvers, (2) the algorithm works not only on clauses, but also on generalized constraints, and (3) for clauses (and more generally, for cardinality constraints) the complexity of the algorithm is linear in the size of the constraints. We implemented and evaluated the algorithm with the Sat4j library. This work gave rise to a publication at the FMCAD 2013 international conference [13].

6.1.3. Encoding TLA+ proof obligations for SMT solvers

Participants: Stephan Merz, Hernán Vanzetto.

The TLA$^+$ proof system TLAPS (see section 5.2) is being developed within a project at the MSR-Inria Joint Centre to which we contribute. Typical proof obligations that arise during the verification of TLA$^+$ specifications mix reasoning about sets, functions, arithmetic, tuples, and records. In previous work [47], we have developed translations from TLA$^+$ set theory to SMT-Lib, the standard input language of SMT solvers. The main challenge has been to design a sound translation from untyped TLA$^+$ to the multi-sorted first-order logic that underlies SMT-Lib. Our solution is based on an incomplete type inference based on "typing hypotheses" present in TLA$^+$ proof obligations. When type inference fails, we fall back to an "untyped" encoding where interpreted sorts such as integers are injected into a designated sort of TLA$^+$ values, and proof obligations corresponding to well-sortedness conditions must be discharged during the proof.

In 2013, we have stabilized and extended the type inference, based on a more expressive type system that includes dependent types, predicate types, and subtyping. The new type system is able to solve many more typing conditions during the translation of proof obligations and thus improves both the scope and the efficiency of the SMT backend. It has been implemented as part of the SMT backend of TLAPS, and an article describing the type system has been submitted. A full description will appear in the PhD thesis of Hernán Vanzetto, expected to be defended in early 2014.

6.1.4. Formalization of stuttering invariance in temporal logic

Participant: Stephan Merz.

Extending our previous formalization in the interactive proof assistant Isabelle/HOL of the concept of stuttering invariance, we formally proved that a property expressible in propositional temporal logic is stuttering invariant if and only if it is equivalent to a formula using only the *until* temporal operator (and in particular not the *next-time* operator). The formalization follows the proof in the classical paper by Peled and Wilke [49]. It allowed us to uncover and correct an error in the proof that had previously not been known. The corresponding extended version of the Isabelle proof development has been accepted at the Archive of Formal Proofs.

6.1.5. Superposition modulo theories

Participants: Noran Azmy, Christoph Weidenbach.

We are currently in a transition phase moving SPASS from a first-order logic prover to a first-order logic prover over theories SPASS(T), in particular arithmetic. Our experience in combining SPASS with interactive verification systems such as TLAPS or Isabelle shows that this is a mandatory step in improving automation [46], [34]. Meanwhile we have built the theoretical foundations [41], [40], [43] for combining superposition with theories which we now turn into algorithmic solutions. This makes an overall reimplementation of SPASS necessary. As a first step we reimplemented and improved our clause normal form transformation [11].

In particular, we want to support integer theories and modulo reasoning [15], as it is often used in distributed algorithms [46]. We have built first implementations of arithmetic modules which we want to combine in 2014 to a first version of SPASS(T).

6.1.6. Presburger Arithmetic in Compiler Optimization

Participants: Marek Košta, Thomas Sturm.

One of our focuses in 2013 was the application of SMT-solvers in new and different problem areas. We started a fruitful cooperation with the Compiler Lab at the Saarland University, Germany on compilation of data-parallel languages.

Data-parallel languages like OpenCL and CUDA are an important means to exploit the parallel computational capabilities of today's computing devices. However, the historical development of data-parallel languages stemming from GPUs plays a crucial role when compiling them for a SIMD (Single Instruction Multiple Data) CPU: on the CPU, one has to emulate dynamic features that on GPU are implemented in the hardware. This difference gives rise to several problems that have to be dealt with during the compilation process.

Our work [15] considers compilation of OpenCL programs for CPUs with SIMD instruction sets. It turns out that SMT-solvers can be used to generate more efficient CPU code. The lack of some dynamic features on CPU implies that one wants to statically decide whether or not certain memory operations access consecutive addresses. Our approach formalizes the notion of consecutivity and algorithmically reduces the static decision to satisfiability problems in Presburger Arithmetic. This is where SMT-solvers come into play. To make an application of an off-the-shelf SMT solver feasible, a preprocessing technique on the SMT problems was introduced. Combining three different systems (computer algebra system REDLOG, SMT-solver Z3, and an OpenCL driver developed in the Compiler Lab), a proof-of-concept system based on our approach was developed. The system generated more efficient code than any other state-of-the-art OpenCL compiler.

Further development is needed to turn the proof-of-concept system mentioned above into one integrated software system. To achieve this, the redundant combination of three heterogeneous systems needs to be replaced by a coherent library offering the same functionality. The work [23] presents the development of such a novel library. The library provides functions to fully automatize the approach proposed in the previous work. It is capable of parallel computations by means of threads and processes and uses an SMT-solver library to carry out the needed computations. To create the final system, the integration of the library with the OpenCL driver needs to be done. This final step is left for future work.

6.1.7. Non-Linear SMT-Solving

Participants: Marek Košta, Thomas Sturm.

In [42] de Moura and Jovanović give a novel satisfiability procedure for the theory of the reals. The procedure uses DPLL-style techniques to search for a satisfying assignment. In case of a conflict, cylindrical algebraic decomposition (CAD) [38] is used to guide the search away from the conflicting state: on the basis of one conflicting point, the procedure learns to avoid in the future an entire CAD cell containing the point. The function realizing this learning is the crucial ingredient that makes the DPLL-style search possible at all. Unfortunately, it is the main computational bottleneck of the whole procedure.

The work of Brown [35] develops a more efficient learning function for the case when the cell to-be learned is full-dimensional. In collaboration with Prof. Brown (United States Naval Academy, USA), we extend this to the general case. While restricting to one cell is quite straightforward for the base and lifting phases of a CAD algorithm, our approach is able to optimize the projection phase as well. This requires a thorough analysis of available geometric infomation and properties of the involved projection operator. Our cell construction algorithm is able to produce bigger cells and it is faster than the approach used in [42]. Both of these are benefits, because a bigger cell means a better generalization of the conflicting assignment. Prototypical implementation of our cell construction algorithm gives very promising results on various kinds of problems. Its elaborate implementation and integration with an DPLL engine within the computer algebra system REDLOG is left for future work. A publication has been submitted to the Journal of Symbolic Computation.

6.1.8. Towards Tropical Decision for NLA

Participant: Thomas Sturm.

Inspired by problems related to stability analysis of chemical reaction networks we have developed an incomplete decision procedure for satisfiability in nonlinear real arithmetic. A first implemented version focuses on specific situations where all variables are known to be stricly positive, which naturally occurs in many scientific contexts. Furthermore, only one single equation is considered. The principal *tropical* approach is, after reducing the problem to finding a point with positive value for f in the considered equation $f = 0$, to consider instead of f only the exponent tuples of the contained summands as points in $\mathbb{Z}^n$. On that basis dominating summands can be identified using LP techniques.

In our particular application discussed in [14], we were able to solve problems, which are intractable even by numerical methods: Typical input equations had around 6000 summands and up to seven variables of degrees between 4 and 9. The methods failed in only 3 percent of the 496 considered input problems.

We are currently generalizing the approach to the general case where variables can have arbitrary values. Furthermore, as it is well known that every existential decision problems over the reals can be equi-satisfiably encoded into one equation, we are aiming at a corresponding general procedure as a long-term research goal.

6.1.9. Hierarchical superposition for arithmetic

Participant: Uwe Waldmann.

Many applications of automated deduction require reasoning in first-order logic modulo background theories, in particular some form of integer arithmetic. A major unsolved research challenge is to design theorem provers that are "reasonably complete" even in the presence of free function symbols ranging into a background theory sort. The hierarchic superposition calculus of Bachmair, Ganzinger, and Waldmann already supports such symbols, but not optimally. We have introduced a novel form of clause abstraction, a core component in the hierarchic superposition calculus for transforming clauses into a form needed for internal operation. We have also demonstrated that hierarchic superposition is refutationally complete for linear integer or rational arithmetic, even if one considers the standard model semantics rather than the first-order semantics, provided that all background-sorted terms in the input are either ground or variables (variables with integer offsets can be permitted in certain positions).

6.2. Proved development of algorithms and systems

6.2.1. Incremental development of distributed algorithms

Participants: Dominique Méry, Manamiary Andriamiarina.

Joint work with Mohammed Mosbah and Mohammed Tounsi from the LABRI laboratory in Bordeaux, France.

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.

Our research was initially supported by the ANR project RIMEL (see http://rimel.loria.fr). More concretely, we aim at an integration of the correct-by-construction refinement-based approach into the *local computation* programming model. The team of LABRI develops an environment called VISIDIA (http://visidia.labri.fr) that provides a toolset for developing distributed algorithms expressed as a set of rewriting rules of graph structures. The simulation of rewriting rules is based on synchronization algorithms, and we have developed these algorithms by refinement [20].

In particular, we show how state-based models can be developed for specific problems and how they can be simply reused by controlling the composition of state-based models through the refinement relationship. Traditionally, distributed algorithms are supposed to run on a fixed network, whereas we consider a network with a changing topology.

The contribution is related to the development of proof-based patterns providing effective help to the developer of formal models of applications [10]. Our patterns simplify the development of distributed systems using refinement and temporal logic. Moreover, we have especially evaluated the extension of the scope of Event B by proposing a technique for integrating fairness in the development of distributed algorithms [17].

6.2.2. Modeling Medical Devices

Participant: Dominique Méry.

Formal modelling 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.

In [9], we propose a refinement-based methodology for complex medical systems design, which possesses all the required key features. A refinement-based combined approach of formal verification, model validation using a model-checker and refinement chart is proposed in this methodology for designing a high-confidence medical device. Furthermore, we show the effectiveness of this methodology for the design of a cardiac pacemaker system.

Inappropriate mode transitions can be a common cause of mishaps in complex health-care systems. In [19], we present an approach for formalizing and reasoning about optimal mode transition in a health-care system that uses several operating modes in various operating states. Modes are formalized and their relation to a state-based formalism is established through a refinement approach. The efficiency of this approach is presented by formalizing an ideal operating mode transition of a cardiac pacemaker case study. An incremental approach is used to develop the system and its detailed design is verified through a series of refinements. In this way, we show how to improve system structuring, elicitation of system assumptions and expected functionality, as well as requirement traceability using modes in state-based modeling. Models are expressed in the Event B [25] modeling language, and they are validated by the model checker ProB.

Finally, in a joint work with colleagues of the CRAN laboratory in Nancy, we have completed a joint project with Airbus on the integration of physiological features in the development of systems like maintenance systems.

6.2.3. *Analysis of real-time Java programs*

Participants: Jingshu Chen, Marie Duflot-Kremer, Pascal Fontaine, Stephan Merz.

Joint work with Nadezhda Baklanova, Jan-Georg Smaus, Wilmer Ricciotti, and Martin Strecker at IRIT Toulouse, France, funded by EADS Foundation (see also section 7.1).

We investigate techniques for the formal verification of programs written in a dialect of Java that includes real-time annotations. Inspired by Safety-Critical Java [36], our partners in Toulouse developed a formal semantics for that dialect in Isabelle/HOL. In joint work, we have designed translations of programs to respectively timed automata and to SMT-Lib for analysis with the Uppaal model checker and with SMT solvers. We are evaluating the features and the scalability of the two approaches, and also plan to formally prove the soundness of the translations based on the semantics formalized in Isabelle.

6.2.4. *Fundamentals of Network Calculus in Isabelle/HOL*

Participant: Stephan Merz.

Joint work with Marc Boyer from ONERA (Toulouse, France) and Loïc Fejoz, Etienne Mabille and Nicolas Navet from RealTime at Work (RTaW, Nancy).

Network Calculus [45] is a well-established theory for the design and analysis of embedded networks. Based on the $(\min, +)$ dioid, it allows a network designer to compute upper bounds for delay and buffer sizes in networks. The theory is supported by several commercial and open-source tools and has been used in major industrial applications, such as the design and certification of the Airbus A380 AFDX backbone. Nevertheless, it is difficult for certification authorities to assess the correctness of the computations carried out by the tools supporting Network Calculus, and we propose the use of *result certification* techniques for increasing the confidence in the Network Calculus toolchain. We have formalized parts of the theory underlying Network Calculus in the proof assistant Isabelle/HOL. We have also developed a prototype analyzer that outputs traces of its computations so that they can be certified using Isabelle. Our work has been published at the conferences EUCASS and ITP [16], [24], and we have submitted a project proposal to ANR together with ONERA, RTaW, Kalray, Eurocopter, and Astrium. Unfortunately, the project was not granted, and future work on this promising subject is on hold.

6.2.5. Modeling and verifying the Pastry routing protocol

Participants: Tianxiang Lu, Stephan Merz, Christoph Weidenbach.

As a significant case study for the techniques that we are developing within VeriDis, we are modeling and verifying the routing protocol of the Pastry algorithm [37] for maintaining a distributed hash table in a peer-to-peer network. As part of his PhD work, Tianxiang Lu developed a TLA$^+$ model of the Pastry routing protocol, and has uncovered several problems in the existing presentations of the protocol in the literature that could lead to network partitioning.

He proposed a novel variant of the protocol and proved its correctness under the strong assumption that no nodes leave the network, using TLAPS (see section 5.2). He also demonstrated that the protocol could not work if arbitrary nodes are allowed to leave; it is not clear at this point under what reasonable assumptions the protocol can be made to work. The correctness proofs contain almost 15000 interactions and constitutes the largest case study carried out so far using TLAPS. Tianxiang Lu defended his thesis at the end of November 2013; a journal publication describing this work is in preparation.

6.2.6. Bounding message length in attacks against security protocols

Participant: Marie Duflot-Kremer.

Joint work with Myrto Arapinis from the University of Birmingham, UK.

Security protocols are short programs that describe communication between two or more parties in order to achieve security goals. Despite the apparent simplicity of such protocols, their verification is a difficult problem and has been shown to be undecidable in general. This undecidability comes from the fact that the set of executions to be considered is of infinite depth (an infinite number of protocol sessions can be run) and infinitely branching (the intruder can generate an unbounded number of distinct messages). Several attempts have been made to tackle each of these sources of undecidability. We have shown [30] that, under a syntactic and reasonable condition of "well-formedness" on the protocol, we can get rid of the infinitely branching part. Following this conference publication, we have submitted a journal version of this result extending the set of security properties to which the result is applicable, in particular including authentication properties.

6.2.7. Evaluating and verifying probabilistic systems

Participant: Marie Duflot-Kremer.

Joint work with colleagues at ENS Cachan and University Paris Est Créteil.

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 was fulfilling 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 cannot fall in the field of model checking. The aim is thus not to tell 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 written. The first one presents the approach in details with a few illustrative applications. The second one focuses on biological application, and more precisely the use of statistical model checking to detect and measure several indicators of oscillating biological systems.

CARTE Project-Team

6. New Results

6.1. Computation and Dynamical Systems

In [12], we analyzed the power of dynamical system that are robust to infinitesimal perturbations. While previous works on this question were limited to very specific kinds of systems such as piecewise constant derivative systems, we obtained results for a quite general class of systems: the main hypothesis being smoothness (which is already a prerequisite in systems that perform analog computation). We show that if a system is robust, then the language it recognizes is computable, and the converse: all computable languages can be recognized by a robust smooth system. Those results are true for discrete-time as well as continuous-time dynamical systems on bounded or unbounded domains.

We investigated in [23], [15], [33] the isomorphism (conjugacy) problem for dynamical systems. While the decidability in the one-dimensional case is a long-standing open problem, we characterize its exact complexity [23] in higher dimensions. Our result suggest that the isomorphism problem is easier than the factoring and embedding problem (decide if one dynamical system is a subsystem of another). A traditional approach to prove two dynamical systems are not isomorphic is to prove that they have different dynamical invariants. We characterised in terms of complexity and computability classes different well known dynamic invariants (periodic points, Turing degrees) in [23], [33].

While Turing machines are usually used for computing, it is an interesting model of dynamical systems, which looks very much like two-dimensional piecewise-affine maps. We investigated dynmacial invariants (entropy and Lyapunov exponents) for Turing machines, and proved quite surprisingly that they are computable. Essentially this means that Turing machines that do interesting computations must do it so slowly that this cannot be seen in their dynamics. This work will be presented in STACS 2014

6.2. Computability, Complexity and Topology

6.2.1. Complexity of real functionals

Computability and topology are closely related as computability assumptions impose topological restrictions: on a topological space, computable functions are continuous and continuous functions are computable relative to some oracle. In the same way, complexity assumptions as bounds on the computation time impose analytical restrictions, but in a way that is not understood yet. For functions from the real numbers to the real numbers, it is known that polynomial-time computable functions correspond to functions with a polynomial modulus of continuity. However for functions on other spaces no such correspondence is known. We investigate the particular case of norms on the space of continuous real functions defined on the unit interval. We introduce analytical characteristics of a norm, namely its dependency on points and the concept of *relevant points*, and use them to characterize the polynomial-time computable norms. This work was presented at LICS 2013 [19]. A full version including other results on non-deterministic complexity classes is currently submitted [28].

6.2.2. Higher-order complexity

While computability theory is well-developed and understood on large classes of topological spaces, complexity theory in analysis is still in its infancy. We argue that the usual way of representing mathematical objects by functions from finite strings to finite strings (order 1 functions) is not appropriate for general spaces. We show that as soon as the space becomes large in a topological sense, it cannot be represented by order 1 functions in a way that respects complexity notions, so we propose to represent objects using higher order functions over finite strings. However higher order complexity theory is not well-understood. The only known class to date is BFF, the class of Basic Feasible Functionals, which does not enjoy nice properties: some intuitively feasible functionals do not belong to the class. We develop a new way of carrying out complexity theory at higher order types, using an adaptation of game semantics. A preliminary version of this work was presented at CCA 2013 [26].

6.2.3. Irreversible computable functions

As mentioned before, computable functions must be continuous. It gives a simple way of proving that some operator is not computable by showing that it is discontinuous. We recall that a function f is computable if there is a *single* oracle Turing machine M that on each x given by an oracle, computes $f(x)$. The following weaker notion is also interesting: a function f*preserves computability* if for each computable x, $f(x)$ is computable. Preservation of computability no more implies continuity, so there is no topological argument to show that some operator does not preserve computability. We develop a strong notion of discontinuity and prove a general result stating that this notion of discontinuity prevents preservation of computability. We apply this result to solve an open problem about the non-computability of the ergodic decomposition. We show that many classical constructions in computability theory are instances of our result. Hence we exhibit deeper connections between computability and topology. The work has been accepted at STACS 2014 [22]. A partial result was published in [13].

6.3. Implicit Computational Complexity

In the setting of non-interference and implicit computational complexity, Emmanuel Hainry, Jean-Yves Marion, and Romain Péchoux presented a characterization of FPSPACE in a language with a fork/wait mechanism [20]. The language used in this work is a classical imperative language with while loops complemented with a mechanism to launch new processes through forks. The fork instruction is heavily inspired by C's fork/wait construction for Unix operating systems, which anchors this work in a down-to-earth setting. Using a type system that enforces a data-ramification on variables, they show that all programs that can be typed and are terminating compute an FPSPACE function, that with a natural evaluation strategy, they indeed use only polynomial space, and conversely that this type system is complete as all FPSPACE functions can be implemented in this language in a typable way.

Emmanuel Hainry and Romain Péchoux also used data-ramification combined with non-interference principles to effectively bound the memory used by object oriented languages in [21]. This work introduces a type system for an object oriented language (derived from java). This type system allows to compute polynomial bounds on the heap and stack used by a typable program, ensuring that if the program halts, it will only use memory under this explicit bound. As the typing procedure is doable in time polynomial in the size of the program, those bounds are easy to obtain, though not tight. Interesting features of this work include inheritance (with overloading and overriding) and, the ability to analyze programs with flow statements controled by objects, contrary to most other works in implicit computational complexity. In [24], Romain Péchoux has shown that the notion of (polynomial) interpretation over term rewrite systems can be adapated on a process language, a variant of the pi-calculus with process recursive definitions. This work shows that the order induced by simulation can be used wrt a given process semantics to infer time and space upper bounds on process resource usage (reduction length, size of sent values, ...).

6.4. Computer Virology

The study on behavioural malware detection has been continued. Guillaume Bonfante, Isabelle Gnaedig and Jean-Yves Marion have been developing an approach detecting suspicious schemes on an abstract representation of the behavior of a program, by abstracting program traces, rewriting given subtraces into abstract symbols representing their functionality. Considering abstract behaviors allows us to be implementation-independent and robust to variants and mutations of malware. Suspicious behaviors are then detected by comparing trace abstractions to reference malicious behaviors.

Model checking is a strong point of our approach: the predefined behavior patterns, used to abstract program traces, are defined by first order temporal logic formulas, as well as the reference suspicious behaviors, given in a signature. The infection problem can then be seen as the satisfaction problem of the formula of the signature by an abstracted trace of the program, which can be checked using existing model checking techniques

The previous work by the team involved abstracting trace automata by rewriting them with respect to a set of predefined behavior patterns defined as a regular language described by a string rewriting system [37], and then, by a term rewriting system [38], which allows to detect information leak.

This work has been finished this year by designing a probabilistic generalization of our approach. Introducing probabilities in our technique allows to express a pertinence degree of detection when analysis of the program results in an incomplete or uncertain program dataflow, or when abstraction cannot be performed reliably. Proposing malware detection with a probabilistic rate is finer and more realistic in practice than giving the binary answer of whether a program is infected or not.

Using a tropical semiring over the reals, they have presented a formalism relying on a weighted term rewriting mechanism, where a weight w, naturally associated to a probability p by the formula: $w = -log(p)$, represents the probability that the realized abstraction be right.

Detection of an abstract behavior has then be defined with respect to a threshold, and a program P exhibits an abstract behavior M if and only if one of its traces admits an abstract form realizing M with a weight not exceeding this threshold.

The weighted abstraction formalism has the advantage of providing a detection algorithm with the same complexity as in the unweighted case, that is linear in the size of the trace automaton [27].

6.5. Graph rewriting

Guillaume Bonfante and Bruno Guillaume provide a new graph rewriting framework adapted to Natural Language Processing. It involves a new form of edge transformation. A new termination technique is also described. The extended paper [17] is accepted for publication in Mathematical Structure in Computer Science.

CASSIS Project-Team

6. New Results

6.1. 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.

6.1.1. Building and verifying decision procedures

Participants: Alain Giorgetti, Olga Kouchnarenko, Christophe Ringeissen, Elena Tushkanova.

We have developed a methodology to build decision procedures by using superposition calculi which are at the core of equational theorem provers. In [14], we have developed automated deduction techniques to prove properties about these superposition-based decision procedures. To this aim, we have further investigated the use of schematic superposition, to check the termination and the combinability of superposition-based procedures. We have worked on the development of a framework for specifying and verifying superposition-based procedures. We have designed an implementation in Maude of the schematic superposition calculus. Thanks to this implementation we automatically derive termination of superposition for a couple of theories of interest in verification.

Until now, schematic superposition was only studied for standard superposition. In [53], [55], we introduce a schematic superposition calculus modulo a fragment of arithmetics, namely the theory of Integer Offsets. This new schematic calculus is used to prove the decidability of the satisfiability problem for some theories extending Integer Offsets. We illustrate our theoretical contribution on theories representing extensions of classical data structures, e.g., lists and records. Our Maude-based implementation has been extended to incorporate this new schematic superposition calculus modulo Integer Offsets. It enables automatic decidability proofs for theories of practical use.

6.1.2. Hierarchical combination of unification procedures

Participant: Christophe Ringeissen.

In [45], [54], a novel approach is described for the combination of unification algorithms for two equational theories which share function symbols. We are able to identify a set of restrictions and a combination method such that if the restrictions are satisfied the method produces a unification algorithm for the union of non-disjoint equational theories. Furthermore, we identify a class of theories satisfying the restrictions. The critical characteristics of the class is the hierarchical organization and the shared symbols being restricted to "inner constructors". Our approach can be applied to theories used for the analysis of protocols. The property of having an inner constructor in one side of an equality is common in the use of exponentiation in Diffie-Hellman inspired key agreement protocols. We are working on considering additional hierarchical theories. A possible candidate theory is a partial theory of Cipher Block Chaining.

6.1.3. Unification modulo equational theories of cryptographic primitives

Participant: Michaël Rusinowitch.

Some attacks exploit in a clever way the interaction between protocol rules and algebraic properties of cryptographic operators. In [74], we provide a list of such properties and attacks as well as existing formal approaches for analyzing cryptographic protocols under algebraic properties.

We have further investigated unification problems related to the Cipher Block Chaining (CBC) mode of encryption. We first model chaining in terms of a simple, convergent, rewrite system over a signature with two disjoint sorts: list and element. The 2-sorted convergent rewrite system is then extended into one that captures a block chaining encryption-decryption mode at an abstract level, (using no AC-symbols); unification modulo this extended system is shown to be decidable [15].

6.2. 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 [70]. 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 6.4.3 we consider derived testing techniques for verifying protocol implementations.

6.2.1. Voting protocols

Participants: Véronique Cortier, David Galindo-Chacon, Stéphane Glondu, Malika Izabachene, Steve Kremer, Cyrille Wiedling.

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. We have studied several protocols that are currently in use:

- Helios is 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 authorities that provides credentials that the ballot box can verify but not forge. This new version has been implemented by Stéphane Glondu and has been tested in a mock election in the teams Cassis and Caramel.
  We have proved computational security for both ballot secrecy and full verifiability (due to our credentials). Helios, as well as Belenios, makes use of threshold decryption, to ensure that decryption keys are distributed among several authorities, yet allowing decryption even some of the authorities are missing. We have provided a fully distributed (with no dealer) threshold cryptosystem suitable for the Helios voting system (in particular, suitable to partial decryption), and prove it secure under the Decisional Diffie-Hellman assumption [40]. Ballot privacy of Belenios then follows from ballot privacy of Helios. For full verifiability, we had first to adapt existing definitions of verifiability in the case of a corrupted ballot box and then prove verifiability of Helios [60].
- The Section 07 of CNRS (now split into Section 06 and Section 07) has proposed a voting protocol for Face-to-Face meetings to enhanced the verifiability of an election run through electronic devices. We have formally modeled this protocol and proved both ballot secrecy and verifiability [32].

Security based on cryptography relies on the fact that certain operations (such as decrypting) are computationally infeasible. However, e-voting protocols should also guarantee privacy in the future, when computers will have an increased computational power and will be able e.g. to break nowadays keys. Such privacy in the future is called *everlasting privacy* and we have proposed a definition of *practical everlasting privacy* [31]. As an illustration, we show that several variants of Helios (including Helios with Pedersen commitments) and a protocol by Moran and Naor achieve practical everlasting privacy, using the ProVerif and the AKiSs tools, which we had to adapt to cope with everlasting privacy.

We have written a popularization science paper on e-voting in Interstices[4].

[4] https://interstices.info/jcms/int_68258/vote-par-internet

6.2.2. Other families of protocols

Participants: Véronique Cortier, Steve Kremer, Robert Künnemann, Cyrille Wiedling.

Securing routing Protocols. The goal of routing protocols is to construct valid routes between distant nodes in the network. If no security is used, it is possible for an attacker to disorganize the network by maliciously interacting with the routing protocols, yielding invalid routes to be built. We have proposed a new model and an associated decision procedure to check whether a routing protocol can ensure that honest nodes only accept valid routes, even if one of the nodes of the network is compromised. This result has been obtained for a bounded number of sessions, adapting constraint solving techniques to node topologies as well as some families of recursive tests, used in routing protocols [16].

Security APIs. In some systems, it is not possible to trust the host machine on which sensitive codes are executed. In that case, security-critical fragments of a program should be executed on some tamper resistant device (TRD), such as a smartcard, USB security token or hardware security module (HSM). The exchanges between the trusted and the untrusted infrastructures are ensured by special kind of API (Application Programming Interface), that are called *security APIs*. We have designed a generic API for key-management based on key hierarchy [23], that can self-recover from corruption of arbitrary keys, provided the number of corrupted, active keys is smaller than some threshold. In [50], we propose a universally composable key management functionality and show how to achieve a secure, distributed implementation on TRDs. We are currently also working on automated verification of security APIs (and more generally protocols that require global mutable state). A tool implementation using the tamarin prover as a backend is currently in progress.

6.2.3. Automated verification of indistinguishability properties.

Participants: Rémy Chrétien, Véronique Cortier, Stéphane Glondu, 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.

Static case. The YAPA tool [17] can check static equivalence for convergent equational theories. It is proved to terminate for a wide class of equational theories that includes subterm convergent theories (e.g. encryption, signatures, pairing and hash) and layered convergent theories (e.g. blind signatures). The procedure is generic in the sense that it remains sound and complete (but may not terminate) for any convergent theory. It has been implemented in the YAPA tool[5].

Active case. We have shown that, for arbitrary equational theories, verifying indistinguishability properties such as trace equivalence in security protocols amounts to deciding the equivalence of constraint systems, i.e., checking whether they have the same set of solutions [20]. 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 [73]. We have extended this decision procedure to the case where the attacker can observe the *length* of messages [37]. This yields the discovery of a new attack on the biometric passport. This attack has been implemented and successfully tested on a small set of passports. This attack is explained in details in a webpage[6] and has obtained some press coverage.

Active case, unbounded number of sessions. Rémy Chrétien has started a PhD on deciding trace equivalence for an unbounded number of sessions. He has shown that for some classes of protocols, decidability of trace equivalence can be reduced to equivalence of deterministic pushdown automata [38]. Equivalence of deterministic pushdown automata is decidable [81] and the corresponding decision procedure is currently implemented by Géraud Senizergues. Based on his tool, we are developing a tool for automatically checking equivalence, for an unbounded number of sessions.

[5] http://www.lsv.ens-cachan.fr/~baudet/yapa/
[6] http://www.loria.fr/ glondu/epassport/attack-lengths.html

6.2.4. Securely Composing Protocols

Participants: Véronique Cortier, Steve Kremer, Éric Le Morvan.

Protocols are often built in a modular way. For example, authentication protocols may assume pre-distributed keys or may assume secure channel. 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. How the security of these protocols can be combined is an important issue that is studied in the PhD thesis recently started by Éric Le Morvan.

A related problem arises when several protocols use the same secrets, e.g. the same keys. While each protocol may be proved secure in isolation, the protocols may become insecure when executed in parallel. In [21] we study whether password protocols can be safely composed, even when a same password is reused. It seems indeed unrealistic to suppose that users do not re-use the same password for different applications. More precisely, we present a transformation which maps a password protocol that is secure for a single protocol session (a decidable problem) to a protocol that is secure for an unbounded number of sessions. Our result provides an effective strategy to design secure password protocols: (i) design a protocol intended to be secure for one protocol session; (ii) apply our transformation and obtain a protocol which is secure for an unbounded number of sessions. Our technique also applies to compose different password protocols allowing us to obtain both inter-protocol and inter-session composition.

6.2.5. Soundness of the Dolev-Yao Model

Participants: Véronique Cortier, Guillaume Scerri.

All the previous results rely on symbolic models of protocol executions in which cryptographic primitives are abstracted by symbolic expressions. This approach enables significantly simple and often automated proofs. However, the guarantees that it offers have been quite unclear compared to cryptographic models that consider issues of complexity and probability. A somewhat recent line of research consists in identifying cases where it is possible to obtain the best of both cryptographic and formal worlds: fully automated proofs and strong, clear security guarantees.

A first approach consists in proving that symbolic models (as the ones studied on the previous sections) are actually *sound* w.r.t. cryptographic models, provided the primitives satisfy some (strong) security properties. Soundness result are usually established for some set of cryptographic primitives and extending the result to encompass new primitives typically requires redoing most of the work. In [35], we propose a notion of computational soundness, amenable to modular extensions. Specifically, we prove that a deduction sound implementation of some arbitrary primitives can be extended to include all standard primitives (asymmetric ans symmetric encryption, public data-structures - e.g. pairings or list, signatures, MACs, and hashes) without repeating the original proof effort. Furthermore, our notion of soundness concerns cryptographic primitives in a way that is independent of any protocol specification language.

Such soundness results require however strong hypotheses on the implementation. For example, primitives must be tagged to avoid confusion between e.g. pairs and encryption. Gergei Bana and Hubert Comon have proposed a new framework [67] where the symbolic model now specifies what an attacker *cannot* do instead of specifying what it can do. Checking protocols security can then be reduced to checking inconsistency of some set of first order formula. During his PhD, Guillaume Scerri studies how to develop a (polynomial) decision procedure for deciding consistency of sets of formulas, for some class of formulas corresponding to security protocols [39].

6.3. Model-based Verification

We have investigated extensions of regular model-checking to new classes of rewrite relations on trees. We have studied specification and proof of modular imperative programs.

6.3.1. Verification of Linear Temporal Patterns over Finite and Infinite Traces

Participants: Pierre-Cyrille Héam, Vincent Hugot, 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 [13] we have investigated suitable semantics for LTL on maximal rewriting words and their influence on the feasibility of a translation, and we have proposed a general scheme providing exact results for a fragment of LTL corresponding mainly to safety formulæ, and approximations for a larger fragment.

6.3.2. Approximations Techniques for Regular Model-Checking

Participants: Aloïs Dreyfus, Pierre-Cyrille Héam, Olga Kouchnarenko.

We address the following general problem of regular model-checking: decide whether $R^*(L) \cap L_p = \varnothing$ where R^* is the reflexive and transitive closure of a successor relation R, and L and L_p are both regular tree languages. Considering a relation R on finite words and a regular language L encoding the initial configurations of a system, the set $R^*(L)$ of accessible words is not necessarily regular. Therefore, a way to verify safety properties is to over-approximate the set of reachable words by a regular language. In [42], we develop new efficient approximation techniques based on syntactic criteria. When these syntactic over-approximations are too coarse, we propose CEGAR-like techniques to refine them using counter-examples. The approach has been successfully applied to verify mutual exclusion protocols.

6.4. 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 [75], or symbolic execution of programs by means of dedicated constraints, SMT solvers, or model-checkers.

6.4.1. Automated Test Generation from Behavioral Models

Participants: Fabrice Bouquet, Kalou Cabrera, Jérome Cantenot, Frédéric Dadeau, Jean-Marie Gauthier, Julien Lorrain.

We have developed an original model-based testing approach that takes a behavioural 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 [30]. To allow the test generation from SysML models, in [47] we study the transformation into a low level language suitable for hardware specification.

In the context of software evolution, we have worked on exploiting the evolution of requirements in order to classify test sequences, and precisely target the parts of the system impacted by this evolution. We have proposed to define the life cycle of a test via three test classes: (i) Regression, used to validate that unimpacted parts of the system did not change, (ii) Evolution, used to validate that impacted parts of the system correctly evolved, and (iii) Stagnation, used to validate that impacted parts of the system did actually evolve. The associated algorithms are under implementation in a dedicated prototype already used in the SecureChange European project.

6.4.2. Scenario-Based Verification and Validation

Participants: Fabrice Bouquet, Kalou Cabrera, Frédéric Dadeau.

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 designed a scenario based testing language for UML/OCL that can be either connected to a model animation engine or to a symbolic animation engine, based on a set-theoretical constraint solver [75]. In the context of the ANR TASCCC project, we investigated the automation of test generation from Security Functional Requirements (SFR), as defined in the Common Criteria terminology. To achieve that, we worked on the definition of description patterns for security properties, to which a given set of SFRs can be related. These properties are used to automatically generate test scenarios that produce model based test cases. The traceability, ensured all along the testing process, makes it possible to provide evidences of the coverage of the SFR by the tests, required by the Common Criteria to reach the highest Evaluation Assurance Levels.

We have proposed a dedicated formalism to express test properties. translated into a finite state automaton which describes a monitor of its behaviors [36]. We have 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[7] [41].

In the context of the SecureChange project, we have also investigatde the evolution of test scenarios. As the system evolves, the model evolves, and the associated test scenarios may also evolve. We are currently extending the test generation and management of system evolutions to ensure the preservation of the security.

6.4.3. Mutation-based Testing of Security Protocols

Participants: Frédéric Dadeau, Pierre-Cyrille Héam, Ghazi Maatoug, Michaël Rusinowitch.

Verification of security protocols models is an important issue. Nevertheless, the verification reasons on a model of the protocol, and does not consider its concrete implementation. While representing a safe model, the protocol may be incorrectly implemented, leading to security flaws when it is deployed. We have proposed a model-based penetration testing approach for security protocols [9]. 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. secret. We have applied our technique on protocols designed in HLPSL, and implemented a protocol mutation tool that performs the mutations. The mutants are then analyzed by the CL-Atse [82] front-end of the AVISPA toolset [64]. We have experimented our approach on a set of protocols, and we have shown the relevance of the proposed mutation operators and the efficiency of the CL-Atse tool to conclude on the vulnerability of a protocol and produce an attack trace that can be used as a test case for implementations. We applied our approach on the Paypal Express protocol, and we were able to retrieve an existing attack trace on this protocol[8]. We are now investigating the transformation of an attack trace into executable tests scripts. To achieve that, we propose to automatically generate skeletons of Java test programs that the validation engineer only has to fill in order to concretize the steps of the test. A first experience in this direction has been described in [48].

6.4.4. 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 geometries combining thin and periodic structures with the possibility of multiple nested scales. We have designed a transformation language facilitating the design of MEMSALab [18]. It

[7] A video of the prototype is available at: http://vimeo.com/53210102
[8] http://www.nbs-system.com/blog/faille-securite-magento-paypal.html

is proposed as a Maple™ package for rule-based programming, rewriting strategies and their combination with standard Maple™ code. We illustrate the practical interest of this language by using it to encode two examples of multiscale derivations, namely the two-scale limit of the derivative operator and the two-scale model of the stationary heat equation. A more general framework for the derivation of the multi-scale models was established in [26].

6.4.5. Code-related Test Generation and Static Analysis

Participants: Fabrice Bouquet, Frédéric Dadeau, Ivan Enderlin, Alain Giorgetti.

We have designed a new annotation language for PHP, named PRASPEL (for *PHP Realistic Annotation SPEcification Language*). This language relies on *realistic domains* which serve two purposes. First, they assign to a data a domain that is supposed to be specific w.r.t. a context in which it is employed. Second, they provide two features that are used for test generation: (i) *samplability* makes it possible to automatically generate a value that belongs to the realistic domain so as to generate test data, (ii) *predicability* makes it possible to check if the value belongs to a realistic domain. This approach is tool-supported in a dedicated framework for PHP which makes it possible to produce unit test cases using random data generators, execute the test cases on an instrumented implementation, and decide the conformance of the code w.r.t. the annotations by runtime assertion checking. This principle has been extended to generate grammar-based textual data based on various strategies, namely uniform random generation, bounded exhaustive generation and rule-coverage-based test generation. In a recent work, we have proposed a dedicated constraint solver for PHP arrays [44] aiming to avoid rejection during the generation of array structures.

6.4.6. Random Testing

Participants: Aloïs Dreyfus, Pierre-Cyrille Héam, Olga Kouchnarenko.

The random testing paradigm represents a quite simple and tractable software assessment method for various testing approaches. When performing random testing, the random sampler is supposed to be independent of tester choices or convictions: a solution is to exploit uniform random generators.

In [78] a method is proposed for drawing paths in finite graphs uniformly, and it is explained how to use these techniques for testing C programs within a control flow graph based approach. Nevertheless, as finite graphs often provide strong abstractions of the systems under test, many abstract tests generated by the approach cannot be played on the implementation. In [79], we have proposed a new approach, extending [78], to manage stack-call during the random test generation while preserving uniformity. In [61], we go further by investigating a way to biase the random testing, in order to optimize the probability to fulfil a coverage criterion. The new approaches have been implemented in a prototype and experimented on several examples. A similar approach for grammar based testing is developped in [43]: we show how to hedge the random generation of execution trees to optimize the probability of covering either all rules or all non terminal symbols.

6.5. 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.

6.5.1. 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 propose a novel 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. This orchestration specification is expressed in ASLan language, a formal language designed for modeling Web Services tied with security policies that was developed in AVANTSSAR project. The AVANTSSAR Orchestrator (presented in [28]) generates an attack trace describing the execution of a the mediator and translates it into ASLan. Then we can check with automatic tools that this ASLan specification verifies required security properties such as secrecy and authentication. If no flaw is found, we can compile the ASLan specification into a Java servlet that can be used to execute the orchestration.

In [34] we introduce an alternative approach based on *fresh-variable 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 prove several closure properties for this class of automata and study their decision problems. We show the applicability of our model to Web services handling data from an infinite domain. We introduce a notion of simulation that enables us to 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 now work on synthesizing composed services that satisfy required security policies.

6.5.2. Secure Querying and Updating of XML Data

Participants: Abdessamad Imine, Houari Mahfoud, Michaël Rusinowitch.

It is increasingly common to find XML views used to enforce access control as found in many applications and commercial database systems. To overcome the overhead of view materialization and maintenance, XML views are necessarily virtual. With this comes the need for answering XML queries posed over virtual views, by rewriting them into equivalent queries on the underlying documents. A major concern here is that query rewriting for recursive XML views is still an open problem, and proposed approaches deal only with non-recursive XML views. Moreover, a small number of works have studied the access rights for updates. In [51], we present SVMAX (Secure and Valid MAnipulation of XML), the first system that supports specification and enforcement of both read and update access policies over arbitrary XML views (recursive or non). SVMAX defines general and expressive models for controlling access to XML data using significant class of XPath queries and in the presence of the update primitives of W3C XQuery Update Facility. Furthermore, SVMAX features an additional module enabling efficient validation of XML documents after primitive updates of XQuery. The wide use of W3C standards makes of SVMAX a useful system that can be easily integrated within commercial database systems as we will show. We give extensive experimental results, based on real-life DTDs, that show the efficiency and scalability of our system.

We introduce in [49] an extension of hedge automata called bidimensional context-free hedge automata, proposing a new uniform representation of vertical and horizontal computation steps in unranked ordered trees. We also extend the parameterized rewriting rules used for modeling the W3C XQuery Update Facility in previous works, by the possibility to insert a new parent node above a given node. Since the rewrite closure of hedge automata languages with these extended rewriting systems is a computable context-free hedge language we can perform some static typechecking on these XML transformations.

6.5.3. On Adding Friends Problem in Social Networks

Participants: Bao Thien Hoang, Abdessamad Imine.

Online social networks are currently experiencing a peak and they resemble real platforms of social conversion and content delivery. Indeed, they are exploited in many ways: from conducting public opinion polls about any political issue to planning big social events for a large public. To securely perform these large-scale computations, current protocols use a simple secret sharing scheme which enables users to obfuscate their

inputs. However, these protocols require a minimum number of friends, i.e. the minimum degree of the social graph should be not smaller than a given threshold. Often this condition is not satisfied by all social graphs. Yet we can reuse these graphs after some structural modifications consisting in adding new friendship relations. In this paper, we provide the first definition and theoretical analysis of the "*adding friends*" problem. We formally describe this problem that, given a graph G and parameter c, asks for the graph satisfying the threshold c that results from G with the minimum of edge-addition operations. We present algorithms for solving this problem in centralized social networks [33]. An experimental evaluation on real-world social graphs demonstrates that our protocols are accurate and inside the theoretical bounds.

6.5.4. Access Control Models for Collaborative Applications

Participants: Fabrice Bouquet, Abdessamad Imine, Michaël Rusinowitch.

The importance of collaborative systems in real-world applications has grown significantly over the recent years. The most of new applications are designed in a distributed fashion to meet collaborative work requirements. Among these applications, we focus on Real-Time Collaborative Editors (RCE) that provide computer support for modifying simultaneously shared documents, such as articles, wiki pages and programming source code by dispersed users. Although such applications are more and more used into many fields, the lack of an adequate access control concept is still limiting their full potential. In fact, controlling access in a decentralized fashion in such systems is a challenging problem, as they need dynamic access changes and low latency access to shared documents. In [19], we propose a generic access control model based on replicating the shared document and its authorization policy at the local memory of each user. We consider the propagation of authorizations and their interactions. We propose an optimistic approach to enforce access control in existing collaborative editing solutions in the sense that the access control policy can be temporarily violated. To enforce the policy, we resort to the selective undo approach in order to eliminate the effect of illegal document updates. Since, the safe undo is an open issue in collaborative applications. We investigate a theoretical study of the undo problem and propose a generic solution for selectively undoing operations. Finally, we apply our framework on a collaboration prototype and measure its performance in the distributed grid GRID'5000 to highlight the scalability of our solution.

However, verifying whether the combination of access control and coordination protocols preserves the data consistency is a hard task since it requires examining a large number of situations. In [52], we specify this access control protocol in the first-order relational logic with Alloy, and we verify that it preserves the correctness of the system on which it is deployed, namely that the access control policy is enforced identically at all participating user sites and, accordingly, the data consistency remains still maintained.

COMETE Project-Team

6. New Results

6.1. 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.

6.1.1. Differential privacy with general metrics.

Differential privacy can be interpreted as a bound on the distinguishability of two generic databases, which is determined by their Hamming distance: the distance in the graph determined by the adjacency relation (two databases are adjacent if they differ for one individual).

In [21] we lifted the restriction relative to the Hamming graphs and we explored the implications of differential privacy when the indistinguishability requirement depends on an arbitrary notion of distance. We showed that we can express, in this way, (protection against) kinds of privacy threats that cannot be naturally represented with the standard notion. We gave an intuitive characterization of these threats in terms of Bayesian adversaries, which generalizes the characterization of (standard) differential privacy from the literature. Next, we revisited the well-known result on the non-existence of universally optimal mechanisms for any query other than counting queries. We showed that in our setting, for certain kinds of distances, there are many more queries for which universally optimal mechanisms exist: Notably sum, average, and percentile queries. Finally, we showed some applications in various domains: statistical databases where the units of protection are groups (rather than individuals), geolocation, and smart metering.

6.1.2. Privacy for location-based services.

The growing popularity of location-based services, allowing unknown/untrusted servers to easily collect and process huge amounts of users' information regarding their location, has recently started raising serious concerns about the privacy of this kind of sensitive information. In [19] we studied geo-indistinguishability, a formal notion of privacy for location-based services that protects the exact location of a user, while still allowing approximate information - typically needed to obtain a certain desired service - to be released.

Our privacy definition formalizes the intuitive notion of protecting the user's location within a radius r with a level of privacy that depends on r. We presented three equivalent characterizations of this notion, one of which corresponds to a generalized version [21] of the well-known concept of differential privacy. Furthermore, we presented a perturbation technique for achieving geo-indistinguishability by adding controlled random noise to the user's location, drawn from a planar Laplace distribution. We demonstrated the applicability of our technique through two case studies: First, we showed how to enhance applications for location-based services with privacy guarantees by implementing our technique on the client side of the application. Second, we showed how to apply our technique to sanitize location-based sensible information collected by the US Census Bureau.

6.1.3. Relation between differential privacy and quantitative information flow.

Differential privacy is a notion that has emerged in the community of statistical databases, as a response to the problem of protecting the privacy of the database's participants when performing statistical queries. The idea is that a randomized query satisfies differential privacy if the likelihood of obtaining a certain answer for a database x is not too different from the likelihood of obtaining the same answer on adjacent databases, i.e. databases which differ from x for only one individual.

In [13], we analyzed critically the notion of differential privacy in light of the conceptual framework provided by the Rényi min information theory. We proved that there is a close relation between differential privacy and leakage, due to the graph symmetries induced by the adjacency relation. Furthermore, we considered the utility of the randomized answer, which measures its expected degree of accuracy. We focused on certain kinds of utility functions called "binary", which have a close correspondence with the Rényi min mutual information. Again, it turns out that there can be a tight correspondence between differential privacy and utility, depending on the symmetries induced by the adjacency relation and by the query. Depending on these symmetries we can also build an optimal-utility randomization mechanism while preserving the required level of differential privacy. Our main contribution was a study of the kind of structures that can be induced by the adjacency relation and the query, and how to use them to derive bounds on the leakage and achieve the optimal utility.

6.1.4. A differentially private mechanism of optimal utility for a region of priors

Differential privacy (already introduced in the previous sections) is usually achieved by using mechanisms that add random noise to the query answer. Thus, privacy is obtained at the cost of reducing the accuracy, and therefore the utility, of the answer. Since the utility depends on the user's side information, commonly modeled as a prior distribution, a natural goal is to design mechanisms that are optimal for every prior. However, it has been shown in the literature that such mechanisms do not exist for any query other than counting queries.

Given the above negative result, in [22] we considered the problem of identifying a restricted class of priors for which an optimal mechanism does exist. Given an arbitrary query and a privacy parameter, we geometrically characterized a special region of priors as a convex polytope in the priors space. We then derived upper bounds for utility as well as for min-entropy leakage for the priors in this region. Finally we defined what we call the tight-constraints mechanism and we discussed the conditions for its existence. This mechanism has the property of reaching the bounds for all the priors of the region, and thus it is optimal on the whole region.

6.1.5. Compositional analysis of information hiding

Systems concerned with information hiding often use randomization to obfuscate the link between the observables and the information to be protected. The degree of protection provided by a system can be expressed in terms of the probability of error associated to the inference of the secret information. In [14] we considered a probabilistic process calculus to specify such systems, and we studied how the operators affect the probability of error. In particular, we characterized constructs that have the property of not decreasing the degree of protection, and that can therefore be considered safe in the modular construction of these systems. As a case study, we applied these techniques to the Dining Cryptographers, and we derived a generalization of Chaum's strong anonymity result.

In [26], a similar framework was proposed for reasoning about the degree of differential privacy provided by such systems. In particular, we investigated the preservation of the degree of privacy under composition via the various operators. We illustrated our idea by proving an anonymity-preservation property for a variant of the Crowds protocol for which the standard analyses from the literature are inapplicable. Finally, we made some preliminary steps towards automatically computing the degree of privacy of a system in a compositional way.

6.1.6. Preserving differential privacy under finite-precision semantics

The approximation introduced by finite-precision representation of continuous data can induce arbitrarily large information leaks even when the computation using exact semantics is secure. Such leakage can thus undermine design efforts aimed at protecting sensitive information. For instance, the standard approach to achieve differential privacy (introduced in previous sections) is the addition of noise to the true (private) value. To date, this approach has been proved correct only in the ideal case in which computations are made using an idealized, infinite-precision semantics. In [23], we analyzed the situation at the implementation level, where the semantics is necessarily finite-precision, i.e. the representation of real numbers and the operations on them are rounded according to some level of precision. We showed that in general there are violations of the differential privacy property, and we studied the conditions under which we can still guarantee a limited (but, arguably, totally acceptable) variant of the property, under only a minor degradation of the privacy

level. Finally, we illustrated our results on two cases of noise-generating distributions: the standard Laplacian mechanism commonly used in differential privacy, and a bivariate version of the Laplacian recently introduced in the setting of privacy-aware geolocation.

6.1.7. Metrics for differential privacy in concurrent systems

Many protocols for protecting confidential information have involved randomized mechanisms and a nondeterministic behavior (such as the Dining Cryptographers protocol or the Crowds protocol). In [28], we investigate techniques for proving differential privacy in the context of concurrent systems which contain both probabilistic and nondeterministic behaviors. Our motivation stems from the work of Tschantz et al., who proposed a verification method based on proving the existence of a stratified family of bijections between states, that can track the privacy leakage, ensuring that it does not exceed a given leakage budget. We improve this technique by investigating state properties which are more permissive and still imply differential privacy. We consider three pseudometrics on probabilistic automata: The first one is essentially a reformulation of the notion proposed by Tschantz et al. The second one is a more liberal variant, still based on the existence of a family of bijections, but relaxing the relation between them by integrating the notion of amortization, which results into a more parsimonious use of the privacy budget. The third one aims at relaxing the bijection requirement, and is inspired by the Kantorovich-based bisimulation metric proposed by Desharnais et al. We cannot adopt the latter notion directly because it does not imply differential privacy. Thus we propose a multiplicative variant of it, and prove that it is still an extension of weak bisimulation. We show that for all the pseudometrics the level of differential privacy is continuous on the distance between the starting states, which makes them suitable for verification. Moreover we formally compare these three pseudometrics, proving that the latter two metrics are indeed more permissive than the first one, but incomparable with each other, thus constituting two alternative techniques for the verification of differential privacy.

6.1.8. Unlinkability

Unlinkability is a privacy property of crucial importance for several systems (such as RFID or voting systems). Informally, unlinkability states that, given two events/items in a system, an attacker is not able to infer whether they are related to each other. However, in the literature we find several definitions for this notion, which are apparently unrelated and shows a potentially problematic lack of agreement. In [20] we shed new light on unlinkability by comparing different ways of defining it and showing that in many practical situations the various definitions coincide. It does so by (a) expressing in a unifying framework four definitions of unlinkability from the literature (b) demonstrating how these definitions are different yet related to each other and to their dual notion of "inseparability" and (c) by identifying conditions under which all these definitions become equivalent. We argued that the conditions are reasonable to expect in identification systems, and we prove that they hold for a generic class of protocols.

6.1.9. Trust in anonymity networks

Trust metrics are used in anonymity networks to support and enhance reliability in the absence of verifiable identities, and a variety of security attacks currently focus on degrading a user's trustworthiness in the eyes of the other users. In [16] we have presented an enhancement of the Crowds anonymity protocol via a notion of trust which allows crowd members to route their traffic according to their perceived degree of trustworthiness of each other member of the crowd. Such trust relations express a measure of an individual's belief that another user may become compromised by an attacker, either by a direct attempt to corrupt or by a denial-of-service attack. Our protocol variation has the potential of improving the overall trustworthiness of data exchanges in anonymity networks, which cannot normally be taken for granted in a context where users are actively trying to conceal their identities. Using such formalization, in the paper we have then analyzed quantitatively the privacy properties of the protocol under standard and adaptive attacks.

6.2. 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.

6.2.1. Models and Emerging Trends of Concurrent Constraint Programming

The *Concurrent constraint programming (ccp)* paradigm focuses on information access and therefore it is suited for this new era of concurrent systems. Ccp singles out the fundamental aspects of asynchronous systems whose agents (or processes) evolve by accessing information in a global medium, represented as constraints over the variables of the system. Agents communicate by posting and querying partial information in the medium. This covers a vast variety of systems as those arising in biological phenomena, reactive systems, net- centric computing and the advent of social networks and cloud computing. In [17] we surveyed the main applications, developments and current trends of ccp.

6.2.2. Efficient computation of program equivalence for confluent concurrent constraint programming

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 [25] we first showed that this verification algorithm has an exponential-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.

6.2.3. Abstract Interpretation of Temporal Concurrent Constraint Programs

Timed concurrent constraint programming (tcc) is a declarative model for concurrency offering a logic for specifying reactive systems, i.e. systems that continuously interact with the environment. The universal tcc formalism (utcc) is an extension of tcc with the ability to express mobility. Here mobility is understood as communication of private names as typically done for mobile systems and security protocols. In [15] we considered the denotational semantics for tcc, and we extended it to a "collecting" semantics for utcc based on closure operators over sequences of constraints. Relying on this semantics, we formalized a general framework for data flow analyses of tcc and utcc programs by abstract interpretation techniques. The concrete and abstract semantics we proposed are compositional, thus allowing us to reduce the complexity of data flow analyses. We showed that our method is sound and parametric with respect to the abstract domain. Thus, different analyses can be performed by instantiating the framework. We illustrated how it is possible to reuse abstract domains previously defined for logic programming to perform, for instance, a groundness analysis for tcc programs. We showed the applicability of this analysis in the context of reactive systems. Furthermore, we made also use of the abstract semantics to exhibit a secrecy flaw in a security protocol. We also showed how it is possible to make an analysis which may show that tcc programs are suspension free. This can be useful for several purposes, such as for optimizing compilation or for debugging.

6.2.4. Foundations of Probabilistic Concurrent Systems

In [24] we introduced a formal proof system for compositional verification of probabilistic concurrent processes. Properties are expressed using a probabilistic modal μ-calculus, and the proof system is formulated as a sequent calculus in which sequents are given a quantitative interpretation. A key feature is that the probabilistic scenario is handled by introducing the notion of Markov proof, by which each proof in the

system is interpreted as a Markov Decision Process, with the proof only considered valid in the case that the value of the MDP is zero.

DICE Team

6. New Results

6.1. Economy of the world data flows

We have attempted to measure data flows in the world to estimate the concentration of the data industry. It is well known that the main plateforms of the Web, Google, Facebook, Amazon, etc. are concentrated in a few countries, mostly in the USA. Some countries, mostly asian, such as China, Russia, Korea or Japan have succesfully developed their own Web 2.0 industry, while others, such as European countries, hava failed to do so. We have explored in [6], the strategy of China, which has the largest Web industry behind the US and has made a priority of keeping its data at home, with systems in all activity sectors developed in general only one or two years after their main americain counterparts. The innovation strategy of China aims in all fields to achieve technological independence, with at most 30% of foreign IP as we have shown in [2].

A tentative measure of the flows of personal data from different regions is proposed in [3], based on the traffic on the largest platform at the international level. We show in particular that personal data captured in Europe on Web platforms mostly go to the US industry. In [4], we investigate the invisible part of the Web constituted by the trackers that are hidden on Web pages and transfer data to third parties, and show that the domination of the US is even stronger on trackers than it is on the visible Web.

6.2. Flow systems

We are currently working with Bull SA, Manuel Selva (PhD) and Lionel Morel from the Socrates team to build a monitoring framework for dynamic data-flow system in many-core architectures. Data-flow computing models computation as a pipeline of computation units absorbing a continuous stream of data. This computing model suits application development for embedded devices such as MPEG-4 video encoders. The incoming data flow is sliced into small size token (e.g. video frames). Each time, all computational units take some tokens from their inputs and produce some tokens on their outputs. We focus [7], [8] on a management layer for handling dynamic dataflow programs in many-core architectures, where computation units may be relocated at runtime from one core to another. The questions raised by Twitter Storm, Google Millwheel or Yahoo S4, are in essence very similar. Can our current architectures hold the information dataflow produced by users in terms of computing power and memory usage? We are currently extending these embedded results to study dataflow architectures with ATOS on flow computing inside Web browsers.

François Goichon will defend his PhD on resource access equity into best-effort operating systems such as Linux. Linux is built over a layered architecture, where each layer owns a local policy that may lead to a global policy being far from best-effort. With Guillaume Salagnac from Socrates team, we show [5], [9] that we can develop malware user space applications exploiting embedded linux firmware and device drivers differential policy that can block other concurrent applications from accessing CPU time. When this kind of applications are installed in multi-tenant architectures as found in cloud shared space, it can slowdown the entire system. These results are interesting for Dice when considering access time in web browser. Current in-browser applications are developed in Javascript, which imposes a single threaded executed model to the developer, yet operated on a multi-core architecture. Best-effort operating systems are not the best approaches to handle flow based applications that become the norm, and we think that some small, low-level shifts, should be considered.

PRIVATICS Team

5. New Results

5.1. Online Social Networks Tracking

Participants: Mohamed Ali Kaafar, Abdelberi Chaabane.

Behavioural advertisement, profiling, adver-gaming and social advertisement illustrate how user personal information and social relations have been integrated to the market model. In other words, user information is *commodified*: the user identity becomes a commodity to be sold and bought. This radical change raises several privacy questions and leads to clamouring for better understanding, regulation and protection of user privacy. Within this context, there is both a long-term and a short-term dimension to our work. For the sort term, I showed that OSN can present a real threat to user privacy as the *full* control of user data – both access and dissemination – is hard to achieve. For the long term, our work calls for a better educational approach to privacy as well as a stricter regulation.

5.2. Behavioural advertisement

Participants: Mohamed Ali Kaafar, Abdelberi Chaabane.

Online Social Networks Tracking. I examined web user tracking capabilities of the three major global OSNs. I studied the mechanisms which enable these services to persistently and accurately follow users web activity, and evaluate to which extent this phenomena is spread across the web. Through a study of the top 10K websites, our findings indicate that OSN tracking is diffused among almost all website categories, independently from the content and the audience. I also evaluated the tracking capabilities in practice and demonstrated – by analysing a real traffic traces – that OSNs can reconstruct a significant portion of users web profile and browsing history. I finally provided insights into the relation between the browsing history characteristics and the OSN tracking potential, highlighting the high risk properties. This work shows that web tracking in combination with personal information from social networks represents a serious privacy violation that shifts the tracking from a virtual tracking (i.e. the user is virtual) to a real "physical" tracking (i.e. based on user personal information).

5.3. Selling Off Privacy at Auction

Participants: Claude Castelluccia, Lukasz Olejnik, Cédric Lauradoux, Minh-Dung Tran.

The first one is a privacy analysis of Real-Time Bidding (RTB) and Cookie Matching (CM). RTB is a technology that allows ad buyers (advertisers) and ad sellers (publishers) to buy and sell ad spaces at real-time auctions through ad exchanges. In RTB, when user visits a publisher page, the ad impression (i.e. one ad display in an ad space) and the user information are immediately broadcast by the ad exchange to a number of bidders (i.e. advertisers or their representatives) for them to bid for the chance to serve ads to this user. CM protocol allows the ad exchange and the bidder to synchronize their cookies of the same user, thus facilitating their exchange of user data.

In [41], we characterize and quantify the potential user web history leakage from ad exchanges to bidders in RTB as a result of exchanging user data. We also discuss and quantify the extent to which companies can potentially collude to increase their tracked user profiles using CM. In addition, we leverage a design characteristic of RTB to observe the winning price of each RTB auction. By analyzing these prices, we show how advertisers evaluate the value of user privacy. This work (titled Selling Off Privacy at Auction) will be presented in NDSS 2014, San Diego, USA in February, 2014.

5.4. Wi-Fi and privacy

Participants: Cédric Lauradoux, Mathieu Cunche, Levent Demir.

Active service discovery in Wi-Fi involves wireless stations broadcasting their Wi-Fi fingerprint, i.e. the SSIDs of their preferred wireless networks. The content of those Wi-Fi fingerprints can reveal different types of information about the owner. In [5], we focus on the relation between the fingerprints and the links between the owners. Our hypothesis is that social links between devices owners can be identified by exploiting the information contained in the fingerprint. More specifically we propose to consider the similarity between fingerprints as a metric, with the underlying idea: similar fingerprints are likely to be linked. We have studied the performances of several similarity metrics on a controlled dataset and then apply the designed classifier to a dataset collected in the wild. Our study is based on a dataset collected in Sydney, Australia, composed of fingerprints belonging to more than 8000 devices.

Extending this problem, we present a set of attacks that allow an attacker to link a Wi-Fi device to its owner identity. We present two methods that, given an individual of interest, allow identifying the MAC address of its Wi-Fi enabled portable device. Those methods do not require a physical access to the device and can be performed remotely, reducing the risks of being noticed. We present in [4], [35] scenarios in which the knowledge of an individual MAC address could be used for mischief.

5.5. Sensor security and privacy

Participants: Claude Castelluccia, Marine Minier, Cédric Lauradoux, Mathieu Cunche.

Wireless sensor networks (WSNs) are composed of a large number of low-cost, low-power, and multi-functional sensor nodes that communicate at short distance through wireless links. They are usually deployed in an open and uncontrolled environment where attackers may be present. Due to the use of low-cost materials, hardware components are not tamper-resistant and an adversary could access to a sensor's internal state.

In [7], we consider packet pollution attack. Packet pollution attack is considered as the most threatening attack model against network coding based sensor networks. A widely held belief says that, in a single source multi-destination dissemination scenario, the total number of polluted packets in the network will grow with the length of the transmission path, and the decoding failure (DF) rate at the further destination nodes are relatively lower. In this work, we first obtain an opposite result by analyzing the pollution attack in multicast scenarios, and find out a convergence trend of pollution attack by network coding system, and quantify the network resiliency against the pollution attacks which happen at any place along the source-destination paths. Then, the analysis result is proved by our simulations on two most widely deployed buffer strategies, Random-In Random-Out (RIRO) and First-in First-Out (FIFO). Finally, it is proved that RIRO has a much advanced security feature than FIFO in constraining the pollution attack gradually, and almost vanished in the end.

An adversary can easily capture even a single node and inserts duplicated nodes at any location in the network. If no specific detection mechanisms are established, the attacker could lead many insidious attacks such as subverting data aggregation protocols by injecting false data, revoking legitimate nodes and disconnecting the network if the replicated nodes are judiciously placed in the network. In [8], we first introduce the algorithm already published in PIMRC 2009 that describes a new hierarchical distributed algorithm for detecting node replication attacks using a Bloom filter mechanism and a cluster head selection. This mechanism could be efficiently used in a WSN as soon as the network is built with a clustering algorithm creating a three tiers hierarchy. We extend the results of our previous results by a theoretical discussion on the bounds of our algorithm. We also perform extensive simulations of our algorithm for random topologies and we compare those results with other proposals of the literature. Finally we show the effectiveness of our algorithm and its energy efficiency.

Finding entropy sources is a major issue to design non-deterministic random generators for headless devices. Our goal in [22] is to evaluate a collection of sensors (e.g. thermometer, accelerometer, magnetometer) as potential sources of entropy. A challenge in the analysis of these sources is the estimation of min-entropy. We have followed the NIST recommendations to obtain pessimistic estimations from the dataset collected during our campaign of experiments. The most interesting sensors of our study are: the accelerometer, the magnetometer, the vibration sensor and the internal clock. Contrary to previous results, we observe far less entropy than it was expected before. Other sensors which measures phenomena with high inertia such as the temperature or air pressure provide very little entropy.

In [12], we propose a key certification protocol for wireless sensor networks that allows nodes to autonomously exchange their public keys and verify their authenticity using one-way accumulators. We examine and compare different accumulator implementations for our pro- tocol on the Sun SPOT platform. We observe that our protocol performs best with accumulators based on Elliptic Curve Cryptography (ECC): ECC-based accumulators have roughly the same speed as Secure Bloom filters, but they have a smaller memory footprint.

5.6. Buidling blocks

Participant: Marine Minier.

In [17], we develop a complete library of lightweight block ciphers dedicated to security applications in wireless sensor networks (WSNs). Choosing best algorithms in terms of energy-efficiency and of small memory requirements is a real challenge because the sensor networks must be autonomous. We study on a dedicated platform of sensors most of the recent lightweight block ciphers as well as some conventional block ciphers. First, we describe the design of the chosen block ciphers with a security summary and we then present some implementation tests performed on our platform. The library is available online: http://bloc.project.citi-lab.fr/library.html.

In [23], we present two related key impossible differential attacks against 14 rounds of Piccolo-80 and 21 rounds of Piccolo-128 without the whitening layers. Piccolo is a new lightweight block cipher proposed by SONY at CHES 2011. The attack against Piccolo-80 has a time and data complexity of $2^{68.19}$ whereas the time/data complexity of the attack against Piccolo-128 is $2^{117.77}$.

While Generalized Feistel Networks have been widely studied in the literature as a building block of a block cipher, we propose in [13] a unified vision to easily represent them through a matrix representation. We then propose a new class of such schemes called Extended Generalized Feistel Networks well suited for cryptographic applications. We instantiate those proposals into two particular constructions and we finally analyze their security.

We also obtain, in [24] a result concerning an integral distinguisher on the SHA-3 finalist Grøstl-512 v3.

5.7. Formal and legal issues of privacy

Participants: Thibaud Antignac, Denis Butin, Daniel Le Métayer.

- **Privacy by design** The privacy by design approach is often praised by lawyers as well as computer scientists as an essential step towards a better privacy protection. The general philosophy of privacy by design is that privacy should not be treated as an afterthought but rather as a first-class requirement during the design of a system. The approach has been applied in different areas such as smart metering, electronic traffic pricing, ubiquitous computing or location based services. More generally, it is possible to identify a number of core principles that are widely accepted and can form a basis for privacy by design. For example, the Organization for Economic Co-operation and Development (OECD) has put forward principles such as the consent, limitation of use, data quality, security and accountability. One must admit however that the take-up of privacy by design in the industry is still rather limited. This situation is partly due to legal and economic reasons: as long as the law does not impose binding commitments, ICT providers and data collectors do not have sufficient incentives to invest into privacy by design. The situation on the legal side might change in Europe though because the regulation proposed by the European Commission in January 2012 (to replace the European Directive 95/46/EC), which is currently under discussion, includes binding commitments on privacy by design.

  But the reasons for the lack of adoption of privacy by design are not only legal and economic: even though computer scientists have devised a wide range of privacy enhancing tools, no general methodology is available to integrate them in a consistent way to meet a set of privacy requirements. The next challenge in this area is thus to go beyond individual cases and to establish sound foundations and methodologies for privacy by design. As a first step in this direction, we have focused on the data minimization principle which stipulates that the collection should be limited

to the pieces of data strictly necessary for the purpose, and we have proposed a framework to reason about the choices of architecture and their impact in terms of privacy. The first strategic choices are the allocation of the computation tasks to the nodes of the architecture and the types of communications between the nodes. For example, data can be encrypted or hashed, either to protect their confidentiality or to provide guarantees with respect to their correctness or origin. The main benefit of a centralized architecture for the "central" actor is that he can trust the result because he keeps full control over its computation. However, the loss of control by a single actor in decentralized architectures can be offset by extra requirements ensuring that errors (or frauds) can be detected *a posteriori*. In order to help the designer grasp the combination of possible options, our framework provides means to express the parameters to be taken into account (the service to be performed, the actors involved, their respective requirements, etc.) and an inference system to derive properties such as the possibility for an actor to detect potential errors (or frauds) in the computation of a variable. This inference system can be used in the design phase to check if an architecture meets the requirements of the parties or to point out conflicting requirements.

- **Accountability**

  The principle of accountability, which was introduced three decades ago in the OECD guidelines, has been enjoying growing popularity over the last few years as a solution to mitigate the loss of control by increasing transparency of data processing. At the European level, the Article 29 Working Group published an opinion dedicated to the matter two years ago and the principle is expected to be enshrined in the upcoming European data protection regulation. But the term "accountability" is used with different meanings by different actors and the principle itself has been questioned by some authors as providing deceptive protections and also possibly introducing new risks in terms of privacy. We have studied the different interpretations of the notion of accountability following a multidisciplinary approach and we have argued that *strong accountability* should be a cornerstone of future data protection regulations. By *strong accountability* we mean a principle of accountability which

  – applies not only to policies and procedures, but also to practices, thus providing means to oversee the effective processing of the personal data, not only the promises of the data controller and its organisational measures to meet them;
  – is supported by precise binding commitments enshrined in law;
  – involves audits by independent entities.

  Strong accountability should benefit all stakeholders: data subjects, data controllers, and even data protection authorities whose workload should be considerably streamlined.

  But accountability is a requirement to be taken into account from the initial design phase of a system because of its strong impact on the implementation of the log architecture. Using real-world scenarios, we have shown that decisions about log architectures are actually nontrivial. We have addressed the question of what information should be included in logs to make their a posteriori compliance analysis meaningful. We have shown how log content choices and accountability definitions mutually affect each other and incites service providers to rethink up to what extent they can be held responsible. These different aspects are synthesized into guidelines to avoid common pitfalls in accountable log design. This analysis is based on case studies performed on our implementation of the PPL policy language.

- **Verification of privacy properties** The increasing official use of security protocols for electronic voting deepens the need for their trustworthiness, hence for their formal verification. The impossibility of linking a voter to her vote, often called voter privacy or ballot secrecy, is the core property of many such protocols. Most existing work relies on equivalence statements in cryptographic extensions of process calculi. We have proposed the first theorem-proving based verification of voter privacy which overcomes some of the limitations inherent to process calculi-based analysis. Unlinkability between two pieces of information is specified as an extension to the Inductive Method for security protocol verification in Isabelle/HOL. New message operators for association extraction and

synthesis are defined. Proving voter privacy demanded substantial effort and provided novel insights into both electronic voting protocols themselves and the analysed security goals. The central proof elements have been shown to be reusable for different protocols with minimal interaction.

- **Privacy and discrimination**

  The interactions between personal data protection, privacy and protection against discriminations are increasingly numerous and complex. For example, there is no doubt that misuses of personal data can adversely affect privacy and self-development (for example, resulting in the unwanted disclosure of personal data to third parties, in identity theft, or harassment through email or phone calls), or lead to a loss of choices or opportunities (for example, enabling a recruiter to obtain information over the Internet about political opinions or religious beliefs of a candidate and to use this information against him). It could even be suggested that privacy breaches and discriminations based on data processing are probably the two most frequent and the most serious types of consequences of personal data breaches. We have studied these interactions from a multidisciplinary (legal and technical) perspective and argued that an extended application of the application of non-discrimination regulations could help strengthening data protection. We have have analysed and compared personal data protection, privacy and protection against discriminations considering both the types of data concerned and the *modus operandi* (*a priori* versus *a posteriori* controls, actors in charge of the control, etc.). From this comparison, we have drawn some conclusions with respect to their relative effectiveness and argued that *a posteriori* controls on the use of personal data should be strengthened and the victims of data misuse should get compensations which are significant enough to represent a deterrence for data controllers. We have also advocated the establishment of stronger connections between anti-discrimination and data protection laws, in particular to ensure that any data processing leading to unfair differences of treatments between individuals is prohibited and can be effectively punished.

PROSECCO Project-Team

6. New Results

6.1. Verification of Security Protocols with Lists in the Symbolic Model

Participants: Bruno Blanchet, Miriam Paiola.

The symbolic model of protocols, or Dolev-Yao model is an abstract model in which messages are represented by terms. Our protocol verifier PROVERIF relies on this model. This year, we have mainly worked on the verification of protocols with lists in this model.

We designed a novel automatic technique for proving secrecy and authentication properties for security protocols that manipulate lists of unbounded length, for an unbounded number of sessions. This result is achieved by extending the Horn clause approach of the automatic protocol verifier ProVerif. We extended the Horn clauses to be able to represent lists of unbounded length. We adapted the resolution algorithm to handle the new class of Horn clauses, and proved the soundness of this new algorithm. We have implemented our algorithm and successfully tested it on several protocol examples, including XML protocols coming from web services. This work has been published in [22] and our prototype is available at http://prosecco.inria.fr/personal/bblanche/publications/BlanchetPaiolaCCS13.html.

Last year, we published a conference paper that shows that, for a limited class of protocols, if a protocol is proven secure by ProVerif with lists of length one, then it is secure for lists of unbounded length. A journal version [50] of this paper has now been accepted.

6.2. Generation of Implementations Proved Secure in the Computational model

Participants: Bruno Blanchet, David Cadé.

The computational model of protocols considers messages as bitstrings, which is more realistic than the formal model, but also makes the proofs more difficult. Our verifier CRYPTOVERIF is sound in this model. This year, we have continued working on our compiler from CRYPTOVERIF speficications to OCaml. Using CryptoVerif and this compiler, we can prove security properties of specifications of protocols in the computational model and generate runnable implementations from such proved specifications. We have published a journal paper on our implementation of SSH generated using this compiler [13] and a proof that this compiler preserves security [23], and we have submitted a journal version of this proof. David Cadé also defended his PhD thesis on this topic [44].

6.3. Computationally Complete Symbolic Attacker and Key Exchange

Participants: Gergely Bana [correspondant], Koji Hasebe, Mitsuhiro Okada.

Around year 2000, various research groups started looking into the relevance of symbolic verification techniques to computational security. If a symbolic verification technique results computational guarantees, we say that computational soundness holds. One of the major concerns has been that the usual Dolev-Yao symbolic attacker that automated symbolic tools used exclusively (to search for attacks) at that time did not seem to allow satisfactory soundness results, only with serious limitations. One possible promising approach to overcome this problem is to derive security guarantees directly as CryptoVerif or F7 does. As an alternative approach, in 2012, Bana and Comon-Lundh introduced a notion they called computationally complete symbolic attacker. With this technique, elimination of the possibility of a computational attack would also mean that computational attack does not exist without the limitations that the Dolev-Yao technique required. Their symbolic attacker can do everything that is not forbidden by conditions derived from standard computational assumptions on the primitives. In this current work, based on predicates for “key compromise”,

we provided such conditions to handle secure encryption even keys are allowed to be sent. We examined both IND-CCA2 and KDM-CCA2 encryptions, both symmetric and asymmetric situations as well as INT-CTXT encryptions. We verified (by hand) a number of protocols as the symmetric Needham-Schroeder protocol, Otway-Rees protocol, Needham-Schroeder-Lowe protocol. Furthermore, we also made some improvements in the computational semantics, and have established a relationship between the computational semantics of Bana and Comon-Lundh and Fitting's embedding of classical logic into S4. This work was published at CCS'13 [19].

6.4. Formal Models and Concrete Attacks on Web Applications

Participants: Karthikeyan Bhargavan [correspondant], Sergio Maffeis, Chetan Bansal, Antoine Delignat-Lavaud, Michael May.

Modern web applications are built as a combination of mostly static servers that host user data and highly dynamic client-side applications that process and present the data to the user. These client-side applications may be hosted as JavaScript within a browser or within custom applications written, say, for smartphones. Hence, in addition to traditional server-side mechanisms, the security of these applications increasingly depends on the correct use of browser-based security mechanisms, client-side access control, and cryptography. These mechanisms are often new, ad hoc, and deserving of close analysis.

Our approach is to formally model various client- and server-side security mechanisms for web applications and rigorously analyze their real-world deployments. When our formal analyses find attacks, we test them against example web applications, report vulnerabilities to various vendors, design countermeasures, and use automated security protocol analysis tools formally verify that our countermeasure resists a large class of attacks. This year, we published three papers in this area. At ESSoS, we formally modeled the authorization policies of common Android apps, found new attacks, and proposed a verified authorization framework [27]. At POST, we formally modeled various cloud-based encrypted storage applications and found both cryptographic and web attacks on them, resulting in patches to these websites and novel countermeasures [20]. At Usenix Security, we proposed a new, safer language for security-critical web components [25]. Defensive JavaScript is a subset of JavaScript that guarantees isolation from other (potentially untrusted) scripts on the same page. This enables, for the first time, the design of cryptographic and single sign-on components that can be formally guaranteed to preserve its secrets even if the hosting website is subject to a cross-site scripting attack.

6.5. Attacks and Proofs for TLS Implementations

Participants: Alfredo Pironti [correspondant], Karthikeyan Bhargavan, Pierre-Yves Strub, Cedric Fournet, Markulf Kohlweiss, Antoine Delignat-Lavaud.

TLS is possibly the most used secure communications protocol, with a 18-year history of flaws and fixes, ranging from its protocol logic to its cryptographic design, and from the Internet standard to its diverse implementations. We have been engaged in a long-term project on verifying TLS implementations and this project is now coming to fruition, with a number of papers are now in the pipeline. We present the main published results below, other papers have been submitted for review.

We have developed a verified reference implementation of TLS 1.2, called miTLS. Our code fully supports its wire formats, ciphersuites, sessions and connections, re-handshakes and resumptions, alerts and errors, and data fragmen- tation, as prescribed in the RFCs; it interoperates with mainstream web browsers and servers. At the same time, our code is carefully structured to enable its modular, automated verification, from its main API down to computational assumptions on its cryptographic algorithms. Our implementation is written in F# and specified in F7. We present security specifications for its main components, such as authenticated stream encryption for the record layer and key establishment for the handshake. We describe their verification using the F7 refinement typechecker. To this end, we equip each cryptographic primitive and construction of TLS with a new typed interface that captures its security prop- erties, and we gradually replace concrete implementations with ideal functionalities. We finally typecheck the protocol state machine, and thus obtain precise security theorems for TLS, as it is implemented and deployed. We also revisit classic attacks and report a few new ones. This work was published at IEEE S&P 2013 [21].

In parallel with this long-term constructive project, we have been analyzing the use of TLS in existing web applications, and our analyses uncovered a number of attacks, leading to patched in popular browsers like Chrome, Internet Explorer, and Firefox, as well as websites like Google and Akamai.

One of these classes of attacks was published at WOOT'13 [29]. In this paper, we identify logical web application flaws which can be exploited by TLS truncation attacks to desynchronize the user- and server-perspective of an application's state. It follows immediately that servers may make false assumptions about users, hence, the flaw constitutes a security vulnerability. Moreover, in the context of authentication systems, we exploit the vulnerability to launch the following practical attacks: we exploit the Helios electronic voting system to cast votes on behalf of honest voters, take full control of Microsoft Live accounts, and gain temporary access to Google accounts.