Rapport INRIA 1994 — Programme 2

Atelier d'outils logicels pour le langage naturel

ACTION ATOLL

3 mai 1995

ACTION ATOLL

Atelier d'outils logicels pour le langage naturel

Localisation : *Rocquencourt*

Mots-clés : analyse lexicale (1), analyse syntaxique (1), automate logique à piles (6), base de données déductives (1), déduction automatique (1), documentation (1), grammaire attribuée (1), intelligence artificielle (1), interface homme-machine (1), interprétation abstraite (1), Lambda-Prolog (1), langage naturel (1), linguistique (1), modèle markovien (1), programmation dynamique (1, 6), programmation en logique (6), programmation logique (1), programmation par contraintes (1), tabulation (6), traitement d'erreur (1).

1 Composition de l'équipe

Responsable scientifique

Bernard Lang, directeur de recherche, INRIA

Responsable permanent

Pierre Boullier, directeur de recherche, INRIA

Secrétariat

Josy Baron

Personnel INRIA

Anne-Marie Vercoustre, directeur de recherche
Eric Villemonte de la Clergerie, chargé de recherche

Conseiller Scientifique

Véronique Donzeau-Gouge, professeur des universités, CNAM

Collaborateurs Extérieurs

François Barthélemy, maître de conférence, CNAM
Marcel Esclozas, ingénieur, détaché pour 6 mois de Rhône-Poulenc

Chercheurs doctorants

Alain Hui Bon Hoa, AMN CNAM
Frédéric Tendeau, boursier INRIA, université d'Orléans

2 Présentation du projet

Dans le contexte de la scission du projet ChLoE, les activités liées à l'axe *environnement* de ce projet ont évolué vers de nouveaux domaines, comme le laissait d'ailleurs présager l'évolution récente de nos centres d'intérêt scientifiques.

Cette évolution coïncide avec l'arrêt du projet Eurêka ESF (Eureka Software Factory), dans le cadre duquel nous avons travaillé pendant près de 6 ans. Pendant cette période, les thèmes principaux que nous avons développés concernaient d'une part les techniques de programmation déclarative, principalement dans les contextes complémentaires de la programmation en logique, des bases de données déductives et de l'analyse syntaxique, et d'autre part le traitement des documents électroniques hypertextuels.

L'une des applications majeures de ce travail concernait le développement d'un environnement de réutilisation automatique de composants logiciels fondé sur le système ALCOOL-90 de François Rouaix. Cette activité est passée cette année dans le projet Cristal, compte tenu de la réorganisation en cours du projet ChLoE, et du changement prévu d'orientation de son axe environnement.

Les travaux présentés ici sont donc à considérer dans le contexte de la préparation d'un nouveau projet de recherche orienté vers les outils logiciels pour le traitement syntaxique des documents, en particulier l'analyse syntaxique et les structures d'hypertexte.

Nous étudions les problèmes de syntaxe sous différents angles. Le développement des techniques d'analyse des langages formels, sans contexte ou faiblement contextuels en particulier, fournit à la fois une architecture

de base et un outil d'expression des principales structures linguistiques. L'adjonction de diverses décorations (structures de traits, probabilités, ...) permet d'exprimer les phénomènes plus fins. Les travaux sur la tabulation en programmation logique ont une étroite parenté avec les techniques d'analyse syntaxique, dont ils sont l'extension naturelle et nécessaire dans le cadre de l'analyse du langage naturel, à la fois pour la syntaxe et pour la sémantique. Ils représentent aussi le contexte le plus simple et le plus général pour étudier le traitement algorithmique des décorations des formalismes syntaxiques.

Notons que cette réorientation vers le traitement syntaxique du langage naturel complète les autres activités de l'INRIA dans ce domaine, qui sont plus axées vers le traitement sémantique ou la reconnaissance de la parole, et se situent géographiquement dans les autres Unités de Recherche.

3 Actions de recherche

3.1 Analyse syntaxique probabiliste

Participants : Frédéric TendeauPierre Boullier, Bernard Lang,

Un des objectifs majeurs du projet, développé dans les autres sections de ce rapport, concerne le développement d'un cadre uniforme d'étude et de réalisation d'analyseurs syntaxiques généraux, utilisant des stratégies variées, pour divers langages formels utilisés en linguistique.

Ces langages formels ont l'avantage de permettre la construction de techniques d'analyse spécialisées pour les structures qu'ils expriment, mais gardent toujours un pouvoir expressif trop faible pour exprimer l'ensemble des structures fines des phénomènes linguistiques. L'un de nos soucis, également considéré dans les travaux de F. Barthélemy, concerne l'extension des algorithmes d'analyse pour leur permettre de traiter des structures décorées par des informations additionnelles. Nous nous intéressons ici à l'addition d'informations probabilistes permettant de guider le processus d'analyse ou le choix des structures syntaxiques en cas d'ambiguïté.

A partir d'une grammaire non-contextuelle, on sait construire un automate à pile décrivant une stratégie d'analyse syntaxique, et l'interprétation en programmation dynamique correspondante. Nous nous

proposons de décorer les règles de grammaire par des attributs (traits, probabilités), et de porter la décoration aux analyseurs.

Nous montrons la validité de ces machines abstraites en exhibant le lien entre les calculs de l'automate et certains parcours dans les arbres d'analyse produits. Les décorations portant sur des règles se retrouvent dans les arbres de la forêt d'analyse. Le lien entre stratégie et parcours des arbres d'analyse nous permet de mieux cerner la définition, la signification et les preuves de correction des fonctions de calcul d'attributs, et en particulier des diverses probabilités utilisées.

Guidés par des travaux récents (algorithme de Stolcke), nous avons mené ces travaux sur l'automate Earley. Il correspond à un schéma mixte (mi-prédictif, mi-synthétique) dont les calculs sont totalement dynamiques. Nous utilisons des stratégies fondées sur l'analyse statique des calculs : compilation d'un automate en ayant pré-calculé le plus de prédictions possible. Ainsi, nous avons commencé par décorer l'automate LC (*left corner*). Notre formalisme et nos preuves permettent de mieux comprendre certains choix et de mieux expliciter les motivations de certains calculs.

Nous avons montré que la généralisation directe de cette approche à l'analyse LR(k) ne peut être réalisée, car elle conduit à un nombre d'états infini. Nous envisageons de résoudre ce problème par une approche mixte statique et dynamique du traitement des probabilités.

3.2 Une architecture générique pour l'analyse syntaxique

Participant : François Barthélemy

Nous avons développé un cadre générique pour l'étude formelle et la réalisation d'analyseurs syntaxiques pour les grammaires avec contraintes. Ce cadre se caractérise par une architecture originale. L'activité d'analyse est séparée en trois tâches distinctes et indépendantes : la construction des squelettes d'arbres d'analyse, la gestion du non-déterminisme quand plusieurs alternatives permettent de prolonger une analyse partielle, la résolution des contraintes. Cette architecture permet de définir des méthodes d'analyse paramétrées par la nature des contraintes, ou plus exactement par le résolveur chargé de les résoudre.

Notre modèle comporte trois niveaux successifs permettant d'aborder des problèmes différents. Le premier est celui de la grammaire, description purement déclarative des aspects syntaxiques d'un langage. Le second niveau comprend une partie des aspects opérationnels : il s'agit d'une machine abstraite, un automate à pile étendu pour prendre en compte les contraintes. Cette machine est non-déterministe, c'est-à-dire qu'elle ne dispose pas d'un mécanisme gérant automatiquement les choix. Certaines propriétés des analyseurs syntaxiques peuvent être utilement étudiées à ce niveau intermédiaire d'abstraction. C'est, par exemple, le cas des propriétés de clôture et de dépendance des variables de la grammaire. Le troisième niveau est celui de l'analyseur syntaxique proprement dit, constitué d'un automate à pile étendu et d'un gestionnaire de non-déterminisme. C'est le bon niveau d'analyse pour certaines propriétés opérationnelles telles que la terminaison et l'efficacité.

Nous avons conforté notre modèle théorique par une implémentation expérimentale. Celle-ci a été réalisée en Alcool–90, un dialecte de la famille de langages de programmation fonctionnels ML disposant d'un système de modules perfectionné fondé sur la notion de surcharge dynamique (initialement développé dans le projet ChLoE par François Rouaix).

L'utilisation de ce langage a permis de réaliser un système modulaire conforme à l'architecture de notre modèle théorique et en particulier de conserver le degré d'abstraction propre à chacun de ses composants [1]. Il en résulte un partage de code entre différents algorithmes. Nous avons pour l'instant écrit une cinquantaine de modules pouvant être combinés de différentes façons, offrant un choix de 66 algorithmes différents pour trois classes de grammaires avec contraintes [2].

Ce système est évolutif. L'ajout de nouveaux modules permet de multiplier les alternatives offertes.

3.3 Les grammaires dynamiques

Participant : Pierre Boullier

Il est bien connu que la *sémantique statique* des langages de programmation n'est rien d'autre que de la syntaxe contextuelle qui est hors d'atteinte des méthodes traditionnelles (déterministes) d'analyse syntaxique utilisées en compilation. Elle ne peut même pas se décrire par des grammaires non contextuelles générales (Chomsky type 2). L'exemple

type en est le langage $\{wcw \mid w \in \{a,b\}^*\}$ qui abstrait la notion de déclaration d'identificateur (le premier w) suivie d'une utilisation de cet identificateur (le deuxième w) et qui ne peut pas se décrire par une grammaire hors contexte.

Nous avons défini la notion de *grammaires dynamiques* qui permet, dans un même formalisme, de décrire à la fois la syntaxe (au sens usuel du terme) et les dépendances contextuelles. Une grammaire dynamique est définie comme un dispositif qui peut engendrer un nombre non borné de grammaires indépendantes du contexte, chacune de ces grammaires étant produite, au cours de l'analyse d'un texte source, par la reconnaissance de constructions particulières. L'analyse d'un texte est faite à l'aide d'une grammaire initiale qui est modifiée (spécialisée) au fur et à mesure de la reconnaissance de ce texte et qui est utilisée pour en poursuivre l'analyse. Par exemple la reconnaissance de la déclaration d'une variable entière `foo` va produire une règle de grammaire qui va stipuler que désormais (jusqu'en sortie de bloc), le symbole terminal `foo` pourra se réduire vers la notion non-terminale de `variable-entière`.

Nous avons montré d'une part que les grammaires dynamiques ont la puissance formelle des machines de Turing et que, d'autre part, cette puissance pouvait s'utiliser pratiquement. Pour valider cette notion, un système expérimental, qui implante un *analyseur dynamique* non ambigu, a été réalisé et ce système a été utilisé pour résoudre quelques problèmes d'analyse sémantique. Certains des exemples traités sont non triviaux (résolution de surcharge, types dérivés, polymorphisme, ...). Signalons en particulier que la résolution de la surcharge dans un langage de type ADA se fait sans aucun algorithme spécifique à une telle résolution : l'analyseur dynamique choisit *naturellement* la bonne interprétation (si elle existe et si elle est unique).

Le lecteur intéressé trouvera les détails de cette recherche en [7].

3.4 DyALog

Participants : Eric Villemonte de la Clergerie, Bernard Lang

Eric de la Clergerie travaille sur le développement d'un interprète en programmation dynamique de programmes logiques appelé DyALog. Cet interprète permet l'exécution avec partage de calculs (par tabulation) de toutes sortes de stratégies de résolution définissables à l'aide d'auto-

mates à piles. Cet évaluateur assure en prime la complétude des réponses et une meilleure détection des boucles.

Les travaux actuels portent d'une part sur les fondements théoriques qui sous-tendent DyALog (notion d'automate à pile logique et ses extensions [SPDA], techniques de programmation dynamique et stratégies de résolution[11]), et d'autre-part sur les problèmes d'implémentation (architecture modulaire de DyALog, partage de structures, indexation, ...). Plus spécifiquement, les travaux les plus récents ont porté sur la mise au point de stratégies générales et fines permettant de combiner arbitrairement évaluation ascendante et descendante en minimisant la quantité d'information traitée au cours des calculs.

Cependant le champ d'application glisse progressivement vers le domaine de la linguistique où l'émergence de grammaires «logiques» (avec unification), intrinsèquement non-déterministes, ainsi que le besoin de stratégies d'analyse plus riches qu'en programmation en logique, justifie de développer plus avant nos techniques.

En conséquence, Eric de la Clergerie a effectué un séjour post-doctoral de novembre 93 à juillet 94 dans le Département Linguistique de AT&T Bell Laboratories (Murray Hill, NJ), sous la direction de Fernando Pereira. Cette visite a été mise à profit pour s'initier au champ de la linguistique calculatoire.

3.5 Tabulation pour λProlog

Participant : Alain Hui Bon Hoa

Alain Hui-Bon-Hoa a achevé en Avril 1994 son service national en tant que VSN à l'université de Pennsylvanie aux Etats-Unis. La période de 16 mois qu'il y a passée lui a permis d'acquérir une plus grande expérience du langage λProlog développé à l'origine dans cette université, et de bénéficier des conseils de son créateur, le professeur Dale Miller. Le travail de thèse qu'il poursuit consiste justement en l'étude de méthodes d'évaluation pour ce langage. En effet, λProlog est un langage de programmation logique étendu qui comprend l'utilisation de λ-termes, ainsi que divers outils logiques permettant de les manipuler aisément. Un tel langage permet de nouvelles et intéressantes applications de la programmation logique à la linguistique, telles que l'analyse syntaxique de propositions relatives par *gap-threading* ou encore l'analyse

sémantique à partir de la grammaire de Montague. Mais ces applications se heurtent, comme dans les clauses de Horn, à l'insuffisance des méthodes par chaînage arrière, manquant de complétude et de partage. Des méthodes d'évaluation reposant sur un chaînage avant sont donc souhaitées. Malheureusement, si la conception et la mise en œuvre de telles méthodes sont bien avancées au niveau des clauses de Horn (voir les travaux d'Eric de la Clergerie), il n'en est pas de même pour la logique des formules héréditaires de Harrop, qui sous-tendent le langage λProlog : en effet, contrairement au cas des clauses de Horn, l'évaluation dans cette logique est contextuelle, dépendant à la fois du programme et de la signature (ensemble de constantes disponibles). La conception de méthodes d'évaluation par chaînage avant pour les formules héréditaires de Harrop demande donc le développement de mécanismes nouveaux, adaptés à la logique intuitionniste utilisée.

Après avoir résolu séparément les problèmes dans des extensions des clauses de Horn avec quantification [4] ou implication dans les buts [3], Alain Hui-Bon-Hoa a réussi à donner une présentation unifiée de ces résultats, qui permet d'aborder l'ensemble de la théorie des formules héréditaires de Harrop [5]. Sa méthode de résolution repose sur des règles de transformation de clauses. Ces règles peuvent être vues comme une généralisation originale du méta-interprète existant pour les formules héréditaires de Harrop. Ce système de résolution autorise une notion de calcul par chaînage avant, ainsi qu'un libre mélange de chaînage avant et chaînage arrière. Ce mélange devrait permettre d'appliquer des méthodes de prédiction afin de limiter la recherche, dans un schéma proche de celui proposé par Eric de la Clergerie.

3.6 Documents électroniques et accès à l'information

Participant : Anne-Marie Vercoustre

Les documents électroniques sont couramment produits en utilisant des éditeurs de textes. Parmi ceux-ci les éditeurs structurés, et en particulier les éditeurs SGML, permettent de produire des documents obéissant à des standards d'organisation et de présentation. Ces éditeurs ont été conçus principalement pour la production de documents sur papier et ne fournissent pas le support approprié pour accéder ou lire ces documents sous forme électronique. Le récent succès des réseaux électronique comme WWW a montré que l'ordinateur pouvait être une aide

non seulement pour produire les documents mais aussi pour accéder à l'information qu'ils contiennent.

L'utilisation de HTML comme langage de description pour échanger de l'information sur le réseau à travers des interfaces comme Mosaic a confirmé l'importance des standards de format. Cependant HTML, et même HTML+, le langage utilisé pour décrire l'information sur WWW, est un modèle de documents trop pauvre pour supporter le processus de lecture, le filtrage de l'information, sa réutilisation, ainsi que le développement d'outils de plus haut niveau.

Nous avons développé un modèle de documents suffisamment riche pour supporter l'édition et la lecture de documents ayant une forte organisation logique. Notre modèle, décrit en SGML, est fondé sur le modèle *article*, avec une extension hypertexte de liens multi-cibles et la possibilité de définir des contextes de lecture attachés à certaines parties du document. Ce modèle est compatible avec HTML et HTML+, ainsi qu'avec le modèle utilisé par IntelliText, un outil de lecture de documents électroniques développé par le CSIRO (Australie).

Notre modèle inclut également une structure de *chemin* (*path*) qui peut être utilisé pour implanter plusieurs outils de navigation, comme une visite guidée, un historique et des *signets* (*hotlist*) structurés [9]. Les modèles de document et de chemin ont été compilés pour l'éditeur SGML Grif, et les fonctions de navigation ont été implantées en développant une application avec la boîte à outils de Grif (GATE). L'ensemble de ces modèles et fonctions peut être vu comme un premier pas vers Grif WWW.

Pour faciliter la recherche d'information dans une base de documents, nous avons également interconnecté Grif avec Sigma, un outil de recherche textuelle à l'intérieur ou parmi des documents, développé par le CSIRO [10] et fondé sur une indexation statistique des documents. Les points clefs de cette intégration sont les suivants :

- L'indexation des documents se fait en appelant Sigma depuis Grif, ce qui permet d'utiliser la structure d'arbre pour déterminer les éléments à indexer (et plus tard de donner à l'utilisateur la possibilité de définir dynamiquement la granularité souhaitée).
- La position dans un document des éléments indexés est stockée par Sigma dans les tables d'index. Sigma retourne à Grif l'ensemble des éléments répondant à la requête et leur position. Le proto-

cole de communication utilise le codage des *selections* développé pour Centaur, c'est-à-dire le chemin depuis la racine de l'arbre jusqu'à l'élément sélectionné. Ceci permet de ne pas engendrer des étiquettes supplémentaires dans le source SGML.

- La requête ainsi que la liste des réponses sont échangées en utilisant une structure de *chemin*.

Cette fonction de recherche textuelle peut être considérée comme une extension de l'éditeur pour aider à la construction de liens statiques, ou comme une fonction de navigation à travers des liens calculés dynamiquement.

4 Actions industrielles

4.1 Consultation Thématique du CNET : projet VADA

Participants : Pierre Boullier, Bernard Lang, Frédéric Tendeau

La convention concernant le projet intitulé « VADA : Analyse Syntaxique avec Valuation d'Attributs dans un demi-anneau », accepté dans le cadre de la Consultation Thématique du CNET de 1993 (thème 6 : « Traitement Automatique de la Langue Naturelle ») a été signée.

Le travail est effectué dans le cadre de la thèse de F. Tendeau.

Sur le plan scientifique, notre objectif est de déterminer un cadre théorique unifié et simple pour la prise en compte uniforme de divers systèmes d'attributs dans les formalismes syntaxiques et dans les outils qui en dérivent, dans le cas assez courant où ces attributs sont valués dans un domaine possédant une structure de demi-anneau. L'un des objectifs pratiques est d'utiliser les résultats obtenus pour déterminer une architecture d'analyseur modulaire, permettant de prendre en compte efficacement divers systèmes d'attributs. La réalisation d'un prototype est prévue.

4.2 Coopération avec Rhône-Poulenc

Participants : Marcel Esclozas, Bernard Lang, François Rouaix

Dans le cadre d'une coopération avec la société Rhône-Poulenc, M. Marcel Esclozas a été détaché pour 6 mois dans notre projet pour y étudier

les problèmes de documentation électronique, sa structure et ses supports, en vue d'une évolution éventuelle des outils utilisés par sa société. Cela a été l'occasion de diffuser auprès de cet acteur économique majeur notre expérience concernant l'utilisation des réseaux électroniques pour la diffusion et l'acquisition d'informations internes ou externes.

5 Actions nationales et internationales

A.M. Vercoustre était en détachement au CSIRO, à la Division of Information Technology, Sydney (Australie), de décembre 1993 à Août 1994, dans le cadre d'une collaboration INRIA-CSIRO.

Une coopération est en cours d'élaboration avec l'université de Linköping (Suède) sur le développement d'évaluateurs logiques tabulaires et leur application au traitement du langage naturel. Notre contribution inclurait notre technologie pour l'implantation effective d'évaluateurs tabulaires et l'analyse syntaxique. L'université de Linköping nous apporterait diverses extensions, en particulier concernant l'utilisation de la négation.

5.1 Actions nationales

Véronique Donzeau-Gouge est membre du comité de rédaction de la revue TSI. Elle était présidente du comité de programme des JFLA'95 (Journées Francophones des Langages Applicatifs), et a organisé matériellement la conférence LICS'94 (Logic In Computer Science) qui s'est tenue au CNAM en juillet 94. Elle a également organisé une présentation des logiciels CoQ et CTCoQ lors de deux journées animées par des chercheurs des projets CoQ et CRISTAL, au CNAM en juin et octobre 1994.

5.2 Actions internationales

Véronique Donzeau-Gouge est membre du comité de direction des conférences *European Software Engineering Conference*, membre des comités de programme des conférences ESEC'95, *Fifth European Software Engineering Conference* (25-28 septembre 95, Sitges, Espagne), et *TAPSOFT'95* (22-26 mai 95, Aarhus Danemark). Elle a aussi été invitée au workshop on *Software Engineering Education, ICSE94* en mai 94. (Sorrento, Italie).

Bernard Lang a été co-président de la conférence POPL'94 «ACM-SIGACT-SIGPLAN conference on Principles of Programming Languages» (Portland, Oregon, janvier 1994). Il a été membre du comité de Programme de la conférence ICSE94 (International Conf. on Software Engineering).

Bernard Lang est éditeur associé des revues «Letters On Programming Languages And Systems» (ACM LOPLAS) et «Computational Linguistics» (Assoc. for Computational Linguistics), et secrétaire élu de ACM-SIGPLAN, Special Interest Group on Programming Languages de l'ACM.

Anne-Marie Vercoustre a été membre du comité d'évaluation de ECHT'94 (Edimbourg, septembre 1994).

5.3 Invitations et Séminaire de recherche

- Nissim Francez, Technion-IIT, Israël, *A WAM-like abstract machine for unification-based grammar formalisms*, septembre.
- Donald A. Smith, University of Waikato, Nouvelle Zélande, *Why Multi-SLD Beats SLD (Even on a Uniprocessor)*, septembre.
- Jan Alexandersson, DFKI, Saarbruck, *Dialogue Processing in Verbmobil*, novembre.
- Ulf Nilsson et Lars Degerstedt, université de Linköping, Suède, *Tabulated Resolution: Search Forests For Normal Programs*, novembre.

6 Diffusion des résultats

6.1 Actions d'enseignement

6.1.1 Jurys de thèse

P. Boullier a présidé le jury de la thèse de B. Marmol (Orléans).

6.2 Participation aux manifestations

- François Barthélemy a présenté ses travaux à JFPL'94, Journées Françaises de la Programmation Logique (Bordeaux, mai-juin),

à la conférence ICLP'94, International Conference on Logic Programming (Gênes/Santa Marguerita, juin), et à la conférence COLING'94 (Tokyo, juillet-août).

- Eric Villemonte de la Clergerie a présenté ses travaux à la conférence ICLP'94, International Conference on Logic Programming (Gênes/Santa Marguerita, juin).
- Alain Hui Bon Hoa a présenté ses travaux à la conférence TACS'94 (Sendaï/Tokyo avril), à JFPL'94, Journées Françaises de la Programmation Logique (Bordeaux, mai-juin), et à ILPS'94, International Logic Programming Symposium (Ithaca-NY, novembre).
- Bernard Lang a participé à la conférence POPL'94 (ACM-SIGPLAN-SIGACT conference on Principles of Programming Languages, Portland) dont il était co-président, au congrès TALN-94, Traduction Automatique du Langage Naturel (Marseille, avril), et aux Journées de Génie Linguistique (Paris, juillet).
- Pierre Boullier, François Barthélemy et Bernard Lang ont participé au $3^{\text{ème}}$ colloque international sur les grammaires d'adjonction d'arbres (Paris, septembre).
- Anne-Marie Vercoustre a présenté ses travaux au Workshop ERCIM W4G, World Wide Web Working Group (Amsterdam, novembre).
- Frédéric Tendeau a participé en mars à l'Ecole des Jeunes Chercheurs en Programmation à Toulouse.

6.3 Diffusion des produits et du savoir-faire

La version 3.8g du système SYNTAX est actuellement en cours de distribution.

6.3.1 A l'INRIA, dans l'enseignement et dans la recherche

SYNTAX est diffusé dans de nombreux instituts de recherche, écoles et universités à fins d'enseignement et de recherche.

A l'INRIA, les plus récentes utilisations concernent les projets :

SPECTRE : SYNTAX est utilisé dans l'outil CÆSAR de compilation de programme LOTOS.

VERSO : SYNTAX construit les analyseurs de l'extension du SGML d'EUROCLID destinée à produire de l'O_2.

6.3.2 Dans l'industrie (contrats)

DOXA Informatique : a acquis la licence d'utilisation de la version 3.8e de SYNTAX sur IBM RS6000.

6.4 Autres

- A l'occasion de la conférence POPL'94, Bernard Lang a passé 2 jours à Bellcore pour travailler avec Kent Wittenburg sur l'analyse syntaxique multidimensionelle.
- Fin avril, Bernard Lang a visité les universités de Linköping et d'Uppsala où il a donné des séminaires sur le thème « Two Views of Tabulation in Parsing ». Bernard Lang a participé à un groupe de travail sur les objectifs de l'ESI, dans les locaux de l'ESI à Bilbao (juin). Il a également participé au Ministère de la Recherche à un groupe de travail sur la « littérature grise » (informelle) et les problèmes de protection de l'information scientifique.
- Anne-Marie Vercoustre a fait une présentation intitulée « Active Documents System » au CSIRO-Sydney (janvier) et à l'Université de Melbourne (mars). Elle a donné un séminaire sur Grif, avec démonstration, en juin dans le cadre du Séminaire *hyperlunch* de l'université de Technologie de Sydney et à la Société Ferntree de Canberra, et en juillet au CITRI à Melbourne.

7 Publications

Communications à des congrès, colloques, etc.

[1] F. BARTHÉLEMY, F. ROUAIX, « Abstract Data-Types and Operators: an Experiment in Constraint-Based Parsing », *in : Proceedings of the Worskhop on ML and its applications*, p. 34–40, 1994.

[2] F. BARTHÉLEMY, F. ROUAIX, « A Modular Architecture for Constraint-Based Parsing », *in : Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, p. 454–460, Kyoto, Japan, Août 1994.

[3] A. Hui-Bon-Hoa, «Intuitionistic implication and resolution», *in : Proceedings of ILPS'94, International Logic Programming Symposium*, 1994.

[4] A. Hui-Bon-Hoa, «Intuitionistic Resolution for a Logic Programming Language with Scoping Constructs», *in : Proceedings of the 1994 international symposium on Theoretical Aspects of Computer Software*, *Lecture Notes in Computer Science*, 789, Springer-Verlag, p. 121–140, 1994.

[5] A. Hui-Bon-Hoa, «Flexible search in the intuitionistic theory of hereditary harrop formulas», *in : ICLP'95, International Conference on Logic Programming*, 1995. en cours de soumission.

[6] E. Villemonte de la Clergerie, B. Lang, «LPDA: Another look at Tabulation in Logic Programming», *in : Proc. of the 11th International Conference on Logic Programming (ICLP'94)*, V. Hentenryck (réd.), MIT Press, p. 470–486, Juin 1994.

Rapports de recherche et publications internes

[7] P. Boullier, «Dynamic grammars and semantic analysis», *Rapport de Recherche n° 2322*, INRIA, Rocquencourt, Août 1994.

[8] F. Tendeau, «Reconnaissance de forêts stochastiques par un automate à pile», Rapport intermédiaire, Décembre 1994.

[9] A.-M. Vercoustre, J.-L. Bouchenez, C. A. Lindley, B. Jansen, «Towards an SGML-based Reading Tool», *Rapport de recherche*, CSIRO et INRIA, 1994, à paraître.

[10] A.-M. Vercoustre, C. A. Lindley, «Information Retrieval and links Authoring in an SGML-based Editor», *Rapport de recherche*, CSIRO et INRIA, 1994, à paraître.

[11] E. Villemonte de la Clergerie, «Modulated Call/Return evaluation strategies for logic programs», submitted to ICLP'95, Novembre 1994.

8 Abstract

The *environment* subgroup of Project ChLoE has redirected its activities towards parsing and application of logic programming techniques to the development of tools for natural language processing, primarily on the syntactic side. The research work on hypertexts and electronic documents is also maintained.

Our research on parsing follows two complementary lines. On the one hand we develop the theory and the efficient implementations of basic

parsing algorithms for a variety of formal grammatical systems, either context-free or weakly context-sensitive. In particular we attempt to obtain a uniform and modular parser architecture. On the other hand we study various techniques to decorate these basic syntactic structures with finer information, such as feature structures or probabilities, to have the means for modeling linguistic phenomena more closely.

The use of decorated grammars is based on a general approach for the development of efficient tabular evaluation techniques for logic program, which correspond to chart techniques in a parsing context. Higher order extensions of these techniques are being researched, and could be applied to semantic analysis.

Dynamic grammars, that evolve as the parse proceeds, have been analysed as an alternative tool for expressing syntactic dependencies.

Our work on hyperdocuments, conducted in cooperation with CSIRO (Australia), has concerned SGML browsing and editing tools, and the definition of stronger document structures compatible with WWW HTML.

Table des matières