Projet PROTHEO

Contraintes, Déduction Automatique et Preuve de Propriétés de Logiciels

Localisation : *Nancy*[1]

Mots-clés : concurrence, contrainte, déduction automatique, démonstration automatique, réécriture, spécification formelle.

1 Composition de l'équipe

Responsable scientifique

Claude Kirchner, DR Inria

Responsable permanente

Hélène Kirchner, DR CNRS

Secrétaire

Christelle Bergeret, TR Inria

Personnel Inria

Adel Bouhoula, CR
Isabelle Gnaedig-Antoine, CR
Christophe Ringeissen, CR
Michaël Rusinowitch, DR

Personnel CNRS

Eric Domenjoud, CR

Personnel Université H. Poincaré

Denis Lugiez, maître de conférence

[1] Projet commun à l'INRIA et au CRIN, URA 262 du CNRS et des universités Henri Poincaré Nancy 1, Nancy 2 et INPL.

Chercheurs doctorants

Farid Ajili, boursier Cnous et Inria
Iliès Alouini, boursier du gouvernement Tunisien et Inria (jusqu'au 30/9/96) puis ATER (Université Henri-Poincaré)
Ali Amaniss, ATER (Université de Metz, jusqu'au 31/8/96)
Narjès Berregeb, boursière Cnous et Inria (jusqu'au 30/9/96) puis ATER (Université de Nancy II)
Peter Borovanský, boursier Cnous et Inria
Carlos Castro, boursier de l'Université Technique Federico Santa Maria, Chili et Inria
Thomas Genet, allocataire MESR
Eric Monfroy, ATER (Université de Nancy II, jusqu'au 31/8/96)
Pierre-Etienne Moreau, allocataire MESR
Christelle Scharff, allocataire MESR
Sorin Stratulat, boursier Inria (à compter du 1/10/96)
Pauline Strogova, boursière MAE et Inria

Chercheurs invités

Rakesh Verma, Houston University (à compter du 1/9/96)

Chercheurs post-doctorants

Régis Curien, enseignant-chercheur (Etablissement Saint-Joseph, Epinal)
Bernhard Gramlich, post-doc Inria (à compter du 1/2/96)
Christopher Lynch, post-doc Inria (jusqu'au 31/7/96)
Laurent Vigneron, ATER
Marian Vittek, post-doc Inria/Industrie (ILOG) (jusqu'au 31/3/96)

2 Présentation du projet

L'objectif du projet PROTHEO est de contribuer à la conception et à la réalisation d'outils permettant de faire des preuves de propriétés de logiciels. Le projet comporte trois volets principaux dont l'intégration est un objectif majeur : la résolution de contraintes, la mécanisation de la déduction et les preuves de propriétés de logiciels.

Nous étudions la résolution de contraintes dans différentes structures algébriques particulièrement intéressantes pour la déduction, que ce soit des domaines de termes ou des domaines numériques comme les entiers ou les réels.

Nos résultats sur la mécanisation de la déduction vont dans plusieurs directions complémentaires. D'une part la conception et le développement d'un cadre logique permettant de décrire les logiques d'intérêt du projet. D'autre part l'étude des preuves avec contraintes et par récurrence, leurs applications étant de première importance pour la preuve de propriétés de logiciels. Enfin, nous étudions la parallélisation de procédures de déduction et l'implantation de la réécriture parallèle pour des machines à mémoire distribuée.

Les résultats de ces recherches s'enrichissent et permettent de traiter des applications s'adaptant bien à la modélisation par les contraintes et à l'utilisation de la déduction, principalement automatique, pour valider des propriétés de logiciels.

Parmi les points marquants de cette année 1996, mentionnons :

– La conception et la réalisation d'un compilateur pour la réécriture non-déterministe avec stratégies pour ELAN, cadre logique basé sur la logique de réécriture et adapté à la spécification et à l'exécution de procédures de déduction, de résolution et de manipulation de contraintes.

– La combinaison des techniques de preuve par induction et du raisonnement associatif-commutatif au sein du système *SPIKE*.

– L'introduction des graphes PATCH, dans la lignée des graphes SOUR, pour traiter les problèmes de complétion dans les groupes.

– La confrontation de nos techniques et résultats à des problèmes de détection d'interaction de service sur les réseaux téléphoniques et à la modélisation et résolution de problèmes de planification de tournées.

3 Actions de recherche

3.1 Contraintes

Nos travaux sur la résolution et la satisfaisabilité de contraintes couvrent un large éventail de problèmes engendrés lors d'un processus de déduction (avec contraintes). Cela va des contraintes réelles non-linéaires à des contraintes d'ordre sur les termes, en passant par des contraintes entières linéaires, des contraintes équationnelles sur les termes du premier ordre ou d'ordre supérieur et des combinaisons de contraintes sur plusieurs domaines différents ou de différentes natures.

3.1.1 Combinaison de solveurs

Participants : Eric Domenjoud, Hélène Kirchner, Christophe Ringeissen

Notre problématique est de combiner des solveurs élémentaires afin d'en construire de nouveaux capables de résoudre des contraintes qui ne pouvaient l'être par un seul solveur. Par le passé, l'intérêt avait d'abord porté sur la combinaison de solveurs de contraintes équationnelles (égalité, filtrage [228], unification) pour le mélange de théories équationnelles disjointes. Nous nous étions ensuite intéressés aux mélanges de théories équationnelles non-disjointes et nos premiers résultats dans ce domaine ont été complétés cette année par un résultat d'indécidabilité qui prouve que l'égalité dans un mélange de théories équationnelles peut être indécidable [241] lorsque l'égalité dans les théories élémentaires est décidable et même si les symboles de fonctions partagés vérifient une propriété adéquate de constructeurs.

Nous avons également poursuivi l'étude du problème de la satisfaisabilité dans une union de théories non nécessairement équationnelles. Ce problème consiste à déterminer si, étant données une contrainte et une théorie, il existe un modèle de la théorie dans lequel la contrainte puisse être satisfaite. Le cas de théories formées sur des signatures disjointes avait été résolu au début des années 80 par Nelson et Oppen. Nous avons apporté une première solution dans le cas où les théories partagent des symboles de fonctions [257].

Parallèlement à ces travaux de nature théorique, nous nous sommes intéressés à l'implantation de différents algorithmes de combinaison pour l'unification. Il s'agit d'algorithmes de combinaison qui s'expriment par des règles de transformations auxquelles viennent s'adjoindre des appels à des solveurs externes. Il était donc tout naturel d'entreprendre le prototypage en ELAN (cf. section 3.2.1). Ce fut l'occasion par ailleurs de valider les possibilités qu'offre ELAN pour l'appel de solveurs externes. Nous avons réalisé un algorithme déterministe pour la combinaison des algorithmes d'unification dans des théories équationnelles dites régulières et non-effondrantes puis plus généralement un algorithme non-déterministe pour résoudre l'unification dans un mélange de théories équationnelles arbitraires.

Dans le futur, cette réalisation logicielle devrait servir de plate-forme à l'intégration d'autres algorithmes de combinaison conçus ou à concevoir dans le projet.

3.1.2 Collaboration de solveurs

Participants : Eric Monfroy, Christophe Ringeissen, Michaël Rusinowitch

Eric Monfroy a soutenu sa thèse [221] sur la collaboration de solveurs pour la programmation logique à contraintes. Le solveur de contraintes est un composant essentiel des systèmes CLP qui influence beaucoup la déclarativité et détermine l'efficacité du système. Néanmoins, il n'est pas toujours possible d'obtenir un solveur suffisamment efficace pour un domaine donné. La coopération et la combinaison de solveurs sont des approches visant à pallier ce problème. Le principe de la coopération réside dans l'utilisation de plusieurs solveurs, et l'échange d'information entre ces solveurs qui travaillent tous sur le même domaine de calcul (les réels par exemple), mais avec des classes de contraintes admissibles différentes (l'un admet des égalités, un autre des inéquations linéaires, ...).

Nous avons réalisé un tel système CLP pour les contraintes polynômiales non-linéaires [255].

Mais jusqu'à présent, les travaux basés sur les concepts de coopération et combinaison sont dédiés à des domaines précis et ne sont pas toujours implantables, ni flexibles, ni adaptables. De cette constatation nous est venue l'idée d'un environnement pour la construction de collaborations de solveurs (la collaboration étant une approche unifiant la coopération et la combinaison). Les objectifs de cet environnement [256, 266] nommé **BALI** sont multiples. Il fournit d'abord un cadre formel et un langage pour manipuler et faire collaborer les solveurs. Cette phase s'appuie sur des primitives de collaborations autorisant plusieurs paradigmes (tels que séquentialité, concurrence et parallélisme), et des primitives de contrôle (telles que répétition, point-fixe, garde, conditionnelle) pour composer les collaborations. La deuxième spécificité de cet environnement est de créer automatiquement des serveurs ayant une architecture distribuée et réalisant les collaborations désirées.

Nous proposons également un langage d'accueil, de type CLP(X), qui offre plusieurs stratégies de résolution de contraintes et dont l'implantation est un client particulier des serveurs. Dès qu'il est connecté à une collaboration, le paramètre X est instancié par le domaine de contraintes de cette dernière. **BALI** peut donc être considéré comme un environnement pour réaliser de nouveaux systèmes CLP fondés sur des collaborations de solveurs. Une maquette a montré la faisabilité de notre approche ainsi que son intérêt en pratique.

Après avoir étudié les améliorations possibles de la partie solveur grâce à la collaboration, nous avons étendu la déclarativité de **BALI**. L'intérêt de notre approche est d'accroître la déclarativité indépendamment du système de contraintes. Nous proposons un schéma [254, 265], nommé **SoleX**, qui étend un solveur en lui permettant de traiter de nouvelles classes de contraintes, c'est-à-dire en autorisant la manipulation de nouveaux symboles de fonctions. **SoleX** est composé de règles de transformations permettant le traitement des nouveaux symboles en amont et en aval du solveur à étendre. Chaque règle de transformation peut être vue comme un solveur composant, et **SoleX** comme une collaboration de solveurs de **BALI**. Ceci apporte une originalité supplémentaire à ce système : l'extension de la déclarativité de **BALI** peut être décrite en **BALI**.

3.1.3 Unification d'ordre supérieur

Participants : Régis Curien, Claude Kirchner, Christophe Ringeissen

Nous avons continué dans deux directions notre étude de la description de l'unification d'ordre supérieur par une théorie équationnelle du premier ordre. D'une part la spécialisation de nos résultats au cas des patterns et d'autre part l'extension permettant de travailler modulo des théories équationnelles du premier ordre comme l'associativité-commutativité. Nous avons également donné un algorithme efficace de filtrage d'ordre 2 modulo l'associativité-commutativité.

Plus précisement, à partir de la procédure d'unification d'ordre supérieur que nous avons élaboré l'an dernier et qui est basée sur l'utilisation du calcul de substitutions explicites $\lambda\sigma$, nous avons étudié avec Gilles Dowek du projet COQ, Thérèse Hardin du projet PARA et Frank Pfenning de CMU, le cas des patterns d'ordre supérieurs introduit par Miller. L'algorithme général se spécialise au cas des patterns

où il devient déterministe grâce à la propriété d'inversibilité à droite des substitutions explicites restreintes aux patterns [242]. Nous avons également précisé les éléments d'une implantation efficace de l'algorithme abstrait et de sa généralisation au cas de la simplification de système d'équations ne comportant pas que des patterns. Testée comme moteur dans ELF, un langage de programmation logique d'ordre supérieur avec contraintes développé par Frank Pfenning, cette implantation a donné de bons résultats expérimentaux.

Afin de permettre dans ce contexte l'introduction de symboles vérifiant des propriétés équationnelles du premier ordre, nous avons étudié la notion d'unification associée. La E-unification d'ordre supérieur est définie comme l'unification modulo la relation d'équivalence $=_{\beta\eta\cup E}$ engendrée par la $\beta\eta$-conversion et une théorie équationnelle du premier ordre E. Nous avons montré comment réduire la E-unification d'ordre supérieur à un problème d'unification du premier ordre dans un mélange formé de E et d'une théorie de substitutions explicites $\lambda\sigma(E)$ qui est une extension du $\lambda\sigma$-calcul intégrant en partie la théorie équationnelle E et ses symboles de fonctions. Il s'agit d'un mélange de théories équationnelles non-disjointes pour lequel les résultats connus sur la combinaison des procédures d'unification ne s'appliquent pas. C'est pourquoi nous avons conçu une nouvelle procédure complète de $\lambda\sigma E$-unification [251] qui s'appuie sur celle développée pour la $\lambda\sigma$-unification. Ceci conduit simplement à considérer des règles de transformation supplémentaires pour gérer l'interaction entre E et $\lambda\sigma$.

En se restreignant au cas du filtrage du second ordre et à l'associativité-commutativité comme théorie équationnelle, Régis Curien a, dans le cadre de sa thèse, élaboré un algorithme efficace en collaboration avec Z. Qian et H. Shi de l'université de Brême. Cet algorithme évite la règle d'imitation qui peut être coûteuse et la remplace par un calcul précis de partition des sous-termes à considérer pour un symbole associatif-commutatif. Ceci élimine de nombreuses solutions redondantes calculées par l'algorithme simple du filtrage d'ordre 2 modulo l'associativité-commutativité et qui rendent celui-ci impraticable, comme Régis Curien l'avait constaté dans sa première implantation. L'algorithme a été implanté par les chercheurs de l'université de Brême et l'ensemble du travail a été présenté à la conférence RTA [240].

3.1.4 Contraintes diophantiennes linéaires

Participants : Farid Ajili, Eric Domenjoud, Claude Kirchner

Dans le cadre du DEA de E. Petitjean [275], nous avons étudié une amélioration d'un algorithme de résolution d'équations linéaires sur les entiers positifs décrit par E. Domenjoud en 1990. Le principe de cet algorithme consiste à déterminer le cône des solutions rationnelles positives puis à y effectuer une recherche des points entiers. Les améliorations portent sur deux points. Les redondances sont éliminées par une triangulation préalable du cône et les triangles sont ensuites explorés en parallèle. Cette seconde partie est en cours de réalisation au moyen de la bibliothèque MPI.

Nous avons implanté la méthode de résolution complète de contraintes diophantiennes linéaires décrite dans [268]. L'analyse qualitative des performances de la méthode a montré que son efficacité augmente dès que le nombre d'inéquations est important. De plus, nous avons montré que la méthode peut servir au calcul d'une seule solution : elle retourne la solution ayant la plus petite longueur.

Nous avons également étudié la combinaison d'un langage de contraintes symboliques tels que l'unification, qui opère sur les termes avec un langage de contraintes prédéfinies qui opère sur un domaine usuel comme par exemple les entiers positifs [229]. Le cas où les deux langages interfèrent par des homomorphismes (par exemple la taille d'un terme) a été résolu de façon modulaire en réutilisant deux solveurs qui coopèrent par l'intermédiaire d'une interface de propagation. Le rôle de cette interface est de maintenir une consistance entre les calculs des deux solveurs. Ceci permet en particulier de traiter le cas de l'interaction entre des solveurs de contraintes symboliques et des solveurs de contraintes diophantiennes linéaires.

3.1.5 Unification de termes schématisés

Participants : Ali Amaniss, Denis Lugiez

Ali Amaniss a soutenu sa thèse [220] portant sur l'étude des schématisations d'ensemble de termes. Les points essentiels de ce travail sont une comparaison détaillée des schématisations à base de grammaires d'arbres et des différentes schématisations récurrentes d'une part, et des algorithmes pour résoudre des problèmes d'inclusion et de généralisation d'autre part. Il a montré également que plusieurs extensions naturelles du formalisme des grammaires primales, introduit par N. Hermann du projet EURECA, conduisaient à des problèmes d'unification indécidables. Les résultats de la comparaison montrent certaines inclusions, mais qu'en général les formalismes sont incomparables. Cette étude exhibe aussi des sous-classes qui sont incluses dans un autre formalisme, permettant de mieux comprendre les puissances d'expressivité. La partie algorithmique décrit un algorithme d'inclusion des I-termes proposés par H. Comon, la quantification ne portant que sur les variables entières. Ali Amaniss s'intéresse ensuite au problème de la généralisation des termes itérés. Il décrit les pièges que comporte ce problème et donne une première proposition d'algorithme pour attaquer ce problème et le résoudre partiellement. Finalement son étude de l'extension naturelle des grammaires primales consistant à augmenter le nombre de règles définissant un symbole montre qu'on obtient ainsi une schématisation dont le problème d'unification est indécidable.

En collaboration avec N. Hermann, nous avons étudié la schématisation des termes itérés sous l'aspect ensembliste. En particulier nous nous sommes intéressés aux opérations ensemblistes sur les ensembles schématisés par ces termes. Nous avons montré que certaines s'effectuent à l'aide d'algorithmes déjà connus et avons donné des contre-exemples à des extensions de propriétés naturelles sur les termes. Finalement, nous avons résolu le problème de généralisation des termes itérés par un algorithme d'énumération que nous avons raffiné pour des généralisation de termes du premier ordre [271, 270].

3.1.6 Contraintes d'ordre

Participants : Thomas Genet, Isabelle Gnaedig-Antoine, Hélène Kirchner

Les méthodes communément utilisées en preuve de terminaison de la réécriture sont les ordres sur les termes. En pratique, la manipulation de ces ordres demande encore une forte interaction avec l'utilisateur. L'un des buts de notre travail est de la mécaniser.

Dans cette optique, nous travaillons sur la résolution de contraintes d'ordre GPO. Rappelons que l'ordre GPO est un ordre générique, regroupant sous un formalisme unique les ordres usuels (RPO, LPO, ordres polynomiaux...). Il est construit sur des fonctions de terminaison qui, sur des instances particulières, donnent les ordres usuels. La définition de notre algorithme de construction d'un ordre GPO particulier à partir d'un système de réécriture a été raffinée. L'algorithme se déroule en deux phases successives : la transformation des contraintes d'ordre GPO en contraintes d'ordre sur les fonctions de terminaison, et la résolution de ces dernières contraintes, pour obtenir, quand c'est possible, des ordres particuliers adéquats. Une implantation a été réalisée en ECLIPSE, qui permet de calculer les contraintes sur les fonctions de terminaison, et, dans le cas particulier où ces fonctions donnent un LPO, de calculer automatiquement une précédence appropriée. Elle a été testée avec succès sur des systèmes de taille conséquente, notamment sur des spécifications algébriques conditionnelles, utilisées par *SPIKE* dans le cadre de preuves de propriétés par récurrence. L'algorithme s'avère, grâce à l'utilisation de structures partagées pour termes et contraintes, plus efficace, dans le cas du LPO, que les implantations classiques de cet ordre, construisant également une précédence [243, 273, 272].

D'autre part, toujours dans le but de mécaniser la preuve de terminaison, nous étudions sa modularité sur la réécriture avec stratégies, et en particulier sur une relation de réécriture qui consiste à normaliser un ensemble de termes par un système de règles R_1 puis par un système de règles R_2, puis à itérer le processus. Cette approche devrait permettre d'obtenir des preuves de terminaison, là où l'approche modulaire générale échoue.

3.1.7 Déduction pour la résolution de contraintes

Participants : Carlos Castro, Claude Kirchner

Les techniques que nous développons pour restreindre les espaces de recherche en déduction automatique peuvent être spécialisées au problème de la résolution ou de la satisfaisabilité de contraintes. Nous avons formalisé en ELAN (cf. section 3.2.1) les problèmes de satisfaction de contraintes (CSP). Les algorithmes de résolution de contraintes (par exemple la recherche par génération et test, ou la réduction de problèmes par consistence de nœuds ou d'arcs) sont exprimés par des règles de réécriture coordonnées par des stratégies [239, 238]. Un système de calcul pour résoudre des ensembles de contraintes unaires et binaires sur des ensembles finis est proposé. Cette formalisation permet de mieux analyser en termes d'actions et contrôle les algorithmes de satisfaction de contraintes souvent implantés comme des boîtes noires. Les preuves de correction en sont ainsi facilitées. Le prototypage en ELAN permet d'expérimenter de nouvelles combinaisons de techniques et de les évaluer.

3.2 Mécanisation de la déduction

Nos travaux reliés à la mécanisation de la déduction portent sur des aspects fondamentaux à étudier en préalable à des preuves de propriétés de logiciels. Nous travaillons ainsi sur un cadre uniforme pour décrire différentes logiques et les exécuter directement à partir de leurs descriptions. Nous étudions également comment mécaniser le raisonnement par récurrence qui est fondamental en mathématiques et qu'il est donc nécessaire de bien savoir mettre en œuvre pour effectuer des preuves formelles de propriétés. Enfin, nous cherchons à améliorer l'efficacité des procédés de déduction à l'aide de techniques de parallélisation, ceci dans le but de pouvoir passer à l'échelle et traiter des applications concrètes significatives.

3.2.1 Logique exécutable

Participants : Peter Borovanský, Claude Kirchner, Hélène Kirchner, Pierre-Etienne Moreau, Marian Vittek

ELAN est un cadre logique qui permet de formaliser, sur une base uniforme, différentes logiques et différents résolveurs de contraintes. La logique de réécriture introduite par J. Meseguer (SRI, Menlo Park) est à la base de la sémantique du mécanisme d'évaluation d'ELAN. En ELAN, les règles peuvent être conditionnelles et sont enrichies par une construction d'affectations locales. Nous avons ajouté à ce formalisme une notion de *stratégie* qui constitue la principale originalité du langage. Un *système de calcul* est une théorie de la logique de réécriture qui est enrichie par un ensemble de stratégies décrivant des calculs et contrôlant l'application des règles de réécriture. Elles sont utilisées à la fois pour décrire le déroulement des preuves menant à un résultat, et pour restreindre l'espace de recherche. Les stratégies s'expriment dans un langage d'expressions construites à partir de noms de règles et de stratégies, par l'opérateur de concaténation, les opérateurs d'itération et les opérateurs de choix qui permettent une gestion relativement fine de l'exploration de l'arbre de recherche. Ainsi ELAN peut être utilisé soit comme un cadre logique pour coder d'autres logiques, soit pour décrire et exécuter des processus décrits par des règles déterministes ou non. Une présentation générale du langage et de ses principales caractéristiques est faite dans [234]. Un certain nombre d'applications y sont mentionnées (voir également en section 3.1.1 et section 3.1.7).

Un compilateur décrit dans [262] a été conçu et implanté pour ELAN. Son originalité principale est de permettre une exécution efficace de la réécriture avec stratégies non déterministes. Son efficacité provient d'une implantation astucieuse du non déterminisme, d'une gestion optimisée de la mémoire, et de techniques de filtrage efficaces. Sur des programmes ne comportant pas d'opérateurs associatifs commutatifs, ELAN compilé se compare avantageusement avec les versions compilées de CAML ou ECLIPSE. Notre effort porte maintenant sur la généralisation du compilateur à l'ensemble des programmes ELAN incluant des opérateurs associatifs et/ou commutatifs.

La logique de réécriture permet de coder d'autres logiques mais aussi de se coder elle-même et présente par là un caractère réflexif connu depuis une dizaine d'années. Dans [252], les capacités reflexives d'ELAN sont étudiées et une théorie universelle pour la classe des programmes ELAN est proposée. Un interpréteur de programmes ELAN est écrit en ELAN et construit, en même temps qu'une dérivation, le terme de preuve associé. Deux modules prédéfinis sont proposés pour la réalisation d'un système réflexif et diverses applications sont mentionnées.

Dans [235], l'idée de contrôler la réécriture par des stratégies est développée en proposant un langage de stratégies plus puissant, dont la sémantique opérationnelle est encore basée sur la réécriture. Un premier langage de stratégies dites élémentaires correspond à la version d'ELAN décrite dans [234]. L'interprétation de stratégies comme ensemble de termes de preuves y est détaillée. Ce langage est ensuite enrichi par des stratégies définies par l'utilisateur sous forme de règles de réécriture. L'implantation de ces idées se poursuit en exploitant la réflexivité de la logique de réécriture.

3.2.2 Preuves par récurrence

Participants : Narjès Berregeb, Adel Bouhoula, Michaël Rusinowitch

Dans le cadre de nos travaux sur la mécanisation des preuves par récurrence et sur le logiciel *SPIKE*, plusieurs extensions ont été étudiées.

Les théories avec opérateurs associatifs et commutatifs interviennent fréquemment dans les problèmes de vérifications de logiciels ou de circuits. Ces opérateurs nécessitent un traitement spécifique car ils engendrent des preuves complexes ou même divergentes. Nous avons poursuivi notre étude des schémas d'induction et construit un algorithme pour déterminer les variables d'induction. L'avantage de cette approche par rapport aux autres techniques de preuve par induction implicite est qu'elle n'utilise pas l'unification AC mais seulement le filtrage AC qui est moins complexe. Nous avons montré la correction et la complétude réfutationnelle (sous certaines restrictions) du nouveau système d'inférence obtenu [232]. Ces résultats ont été implantés et les expérimentations sur des spécifications de circuits ont permis d'obtenir des preuves plus naturelles et demandant moins d'intervention de la part de l'utilisateur qu'avec d'autres systèmes comme NQTHM, LP, PVS. Une démonstration du système a eu lieu à la conférence RTA [233].

Les concepts d'observabilité permettent de mieux comprendre les problèmes de correction de programmes. Dans l'approche observationnelle des objets sont considérés comme égaux si leurs comportements observables sont identiques. Bien qu'un grand nombre de travaux aient été consacrés aux aspects sémantiques de l'observabilité, il existe peu d'études sur les techniques de preuves et encore moins d'implantations. Nous avons développé une nouvelle approche pour intégrer la méthode de *Context Induction* de Hennicker dans le cadre de l'induction implicite. L'aspect le plus original de notre technique est la notion de test-contexte, analogue à celle de test-substitution, qui permet de schématiser l'ensemble des contextes possibles d'un terme. Les premières expérimentations ont donné des résultats très prometteurs.

Nous avons proposé des procédures de preuve par récurrence et de test de complétude suffisante pour les spécifications conditionnelles paramétrées [223]. Pour tester qu'un symbole de fonction f est suffisamment complet, on construit un arbre de motifs pour f et on teste la réductibilité par cas de toutes les feuilles de l'arbre. En se restreignant aux règles conditionnelles à préconditions booléennes, cette procédure apporte à l'utilisateur une aide à la construction de définitions complètes.

Nous avons proposé un schéma général de procédure de preuves par récurrence avec ensemble test ainsi qu'une technique simple de preuve de correction [237]. Cette méthode permet de réfuter plus de conjectures non valides qu'avec les méthodes précédentes grâce à des nouvelles notions d'*ensemble test*, de *positions de récurrence* et de *témoin d'incohérence*. Nous avons proposé également des algorithmes permettant de calculer un *ensemble test* ainsi que les *positions de récurrence* d'une spécification conditionnelle définie à l'aide des constructeurs non-libres. A la différence des travaux antérieurs, la complétude réfutationnelle de cette procédure n'est pas restreinte aux spécifications booléennes.

La plupart des prouveurs existants considèrent uniquement des spécifications suffisamment complètes sur des constructeurs libres. Le prouveur *SPIKE* n'interdit pas l'utilisation des spécifications avec relations entre les constructeurs mais dans ce cas la complétude réfutationnelle de la procédure de preuves par récurrence n'est plus garantie et l'efficacité du prouveur est très limitée en pratique. Nous avons proposé, en collaboration avec Jean-Pierre Jouannaud (LRI, université Paris Sud), une nouvelle méthode de test de complétude suffisante et de preuves par récurrence dans le cadre de la logique des clauses de Horn avec égalité et prédicat d'appartenance [264]. Ce cadre est plus adéquat pour manipuler les constructeurs non-libres car si les constructeurs doivent satisfaire des équations comme $pred(succ(X)) = X$ pour les entiers relatifs, alors l'ensemble des termes constructeurs clos peuvent être reconnus par un automate, d'après un résultat de H. Comon. Cet automate peut être traduit par une spécification dans laquelle les constructeurs deviennent libres dans les nouvelles sous-sortes associées aux états de l'automate, par exemple $pred$ devient libre dans la sorte des entiers négatifs.

3.2.3 Déduction modulo une théorie équationnelle

Participants : Michaël Rusinowitch, Laurent Vigneron

Nos travaux concernant la déduction modulo une théorie équationnelle ont permis de définir un système basé sur la résolution et la paramodulation, remplaçant les axiomes de la théorie considérée par des algorithmes d'unification, de filtrage et des règles de déduction spécifiques. Ce système de déduction a été démontré complet pour une classe de théories englobant l'ensemble des théories régulières. Ce résultat a été obtenu grâce à la définition d'une procédure calculant les extensions nécessaires à la théorie traitée [261].

Le logiciel **daTac**, implantant ces techniques pour le cas des théories associatives-commutatives, est utilisé avec le Prof. A. Wasilewska (Université de l'Etat de New York, Stony Brook) pour étudier des algèbres dites «approximantes» (*rough algebras*) [259]. Ces algèbres sont appliquées à la classification d'informations incertaines en base de données. **daTac** a servi non seulement à démontrer des propriétés de ces algèbres, mais également à engendrer plusieurs centaines de théorèmes inconnus jusque là.
Nous avons également mis au point une procédure permettant de comparer deux algèbres avec **daTac** [260].

3.2.4 Déduction concurrente

Participants : Iliès Alouini, Claude Kirchner, Christopher Lynch, Christelle Scharff

Notre travail sur la déduction concurrente nous a permis de concevoir et d'implanter d'une part la réécriture conditionnelle concurrente et d'autre part la complétion concurrente basée sur la notion de graphe SOUR.

Jusqu'à maintenant, nous avions traité exclusivement le cas des systèmes de réécriture non conditionnels, tant du point de vue du modèle que de l'implantation. Nous avons conçu et implanté une transformation des systèmes conditionnels en systèmes non conditionnels qui préserve au mieux la concurrence inhérente du système et qui repose naturellement sur le modèle de réécriture non conditionnelle concurrente [230]. Cette transformation s'implante facilement dans le modèle non-conditionnel que nous avons développé les années précédentes puisqu'en particulier les drapeaux d'irréductibilité sont de toute façon nécessaire pour détecter la fin du processus de normalisation. Par ailleurs le modèle d'implantation a été étendu : on peut maintenant expérimenter en utilisant une répartition atomique où un noeud du terme à réduire correspond à un processus, ou bien en utilisant un modèle à granularité plus grosse où un processus agit sur plusieurs noeuds du terme. Le prototype RECO [269] écrit en utilisant les bibliothèques PVM et MPI fonctionne sur réseau de station de travail et sur les machines Power Challenge Array et Origin 2000 du Centre Charles Hermite. Les problèmes de répartition de charge initiale et dynamique, spécifique à la réécriture concurrente, sont maintenant à étudier.

Une autre approche de la déduction concurrente que nous avons développée cette année est basée sur la notion de graphe SOUR. Cette structure de données permet de représenter au niveau objet toutes

les opérations nécessaires de passage au **S**ous terme, d'**O**rientation, d'**U**nification et de **R**éécriture. Contrairement aux structures existantes jusqu'alors, le symbole de réécriture est accessible directement au niveau objet, ce qui permet d'implanter naturellement et directement les règles de déduction avec contraintes. Dans le modèle que nous avons développé, chaque noeud d'un graphe SOUR est un processus et les arcs sont des liens de communication. Par conséquent la concurrence est au niveau des termes et la localisation des transformations permet d'éviter le recours à une mémoire globale ou à des synchronisations explicites. Par ailleurs cette approche permet d'implanter des stratégies de déduction qui autorisent la simplification sans restriction. Enfin il n'y a jamais inconsistance du graphe SOUR sous-jacent par effets de bord des actions asynchrones [250]. La procédure a été implantée dans le cas de la complétion close en C, en utilisant PVM et la librairie Leda conçue au MPI à Saarbrücken.

3.2.5 Complétion pour le routage dans les graphes de Cayley

Participants : Claude Kirchner, Christopher Lynch, Pauline Strogova

Nous travaillons avec Dominique Fortin de l'action ARCHI à Rocquencourt sur l'application des méthodes de déduction automatique à la recherche de chemins dans des réseaux réguliers.

Les réseaux qui nous intéressent ici sont des réseaux basés sur des graphes ayant une structure algébrique de groupe fini. Pour résoudre le problème de routage dans cette structure, nous mettons en œuvre des techniques de réécriture dans les groupes. Le problème rencontré ici est de faire passer à l'échelle les techniques standard de réécriture, car les méthodes usuelles basées sur la complétion de systèmes de réécriture engendrent très rapidement des tailles de systèmes et de règles extrèmement importantes. Pour réaliser la complétion de façon plus efficace, nous avons développé une technique issue de celle utilisée pour les graphes SOUR. Les graphes PATCH permettent de représenter les relations entre générateurs du groupe et intègrent les axiomes d'associativité et d'inversibilité. Il en résulte une représentation permettant de réaliser la complétion de groupes [253, 274, 222]. Nous avons donné les règles d'inférence de la complétion dans ce formalisme et montré leur correction par rapport à la complétion standard.

3.2.6 Modularité et vérification des systèmes de réécriture

Participant : Bernhard Gramlich

Nous nous intéressons aux méthodes, techniques et critères généraux permettant de vérifier d'importantes propriétés relatives aux systèmes de réécriture comme la terminaison ou la confluence. Il est notamment primordial d'analyser les aspects modulaires de la réécriture, c'est-à-dire sous quelles conditions la combinaison (union) des systèmes de réécriture hérite d'une propriété connue sur les systèmes composants.

On connait pour l'instant très peu de résultats concernant la confluence des systèmes de réécriture non-terminants. Dans [246] nous avons développé un nouveau critère de confluence pour des systèmes de réécriture linéaires à gauche non nécessairement terminants, qui est basé sur la notion de *paires critiques parallèles*. Ce critère a été montré incomparable aux autres résultats connus. En utilisant cette fois la notion associée de *paires critiques parallèles partagées*, un nouveau critère de confluence pour les systèmes de réécriture conditionnels terminants (joints) a été prouvé dans [244]. La réécriture suivant une stratégie de l'intérieur vers l'extérieur, sa relation avec la réécriture en général, tout comme les aspects modulaires respectifs, sont étudiés dans [247, 245, 224]. De nouvelles conditions pour l'équivalence entre réécriture en général et réécriture de l'intérieur vers l'extérieur sont prouvées dans [247], ce qui implique directement les nouveaux résultats de modularité pour la terminaison des systèmes combinés non-conditionnels. Des contre-exemples dans [245] montrent que, pour la réécriture conditionnelle en général, les propriétés élémentaires comme la terminaison faible et la confluence locale ne sont même pas préservées par extension de signature. Les résultats positifs de préservation et une analyse détaillée de la modularité pour des combinaisons de systèmes de réécriture conditionnels (disjoints et partageant les constructeurs) sont présentés dans [224].

3.2.7 Implantation efficace de la réécriture

Participant : Rakesh Verma

Nous avons travaillé sur des améliorations et des nouvelles fonctionnalités pour Smaran, un interpréteur pour la réécriture (développé à l'université de Houston) qui est basé sur une approche par fermeture de congruence (congruence closure). La principale caractéristique de ce système est de sauvegarder l'histoire complète des réductions dans une structure de données compacte. Nos travaux récents sont tout-à-fait encourageants au vu des performances actuelles de Smaran. Les nouvelles fonctionnalités incluent des types de données prédéfinis ainsi que des outils pour la déclaration et la vérification des types.

4 Actions industrielles

4.1 GIHP-Champagne

Participants : Farid Ajili, Carlos Castro, Eric Domenjoud, Claude Kirchner

Le GIHP-Champagne assure un service de transport porte-à-porte des personnes à mobilité réduites (PMR) sur l'agglomération de Reims. Le service compte actuellement une vingtaine de véhicules et de chauffeurs et assure entre 250 et 300 transports par jour, nombre qui est en forte augmentation. Or l'optimisation du planning journalier des chauffeurs occupe actuellement une personne à temps plein cinq heures durant, et ce, pour un résultat relativement décevant puisque les chauffeurs restent inactifs 40 % du temps. Le GIHP-Champagne nous a donc sollicité pour étudier la possibilité d'informatiser son service transport.

Nous avons proposé une modélisation du problème à base de contraintes portant sur l'allocation des ressources et les temps de trajets, et qui prennent en compte les différentes contraintes liées au transport des PMR [267]. A partir de cette modélisation, nous avons réalisé un prototype en C++ au moyen de la bibliothèque de gestion de contraintes Ilog Solver de la société Ilog.

4.2 Cnet

Participants : Adel Bouhoula, Claude Kirchner, Michaël Rusinowitch, Sorin Stratulat

Le but de ce projet est de mettre à profit les techniques de preuve automatique dans le cadre de la spécification et de la vérification d'applications liées aux télécommunications. Nous espérons couvrir au moyen de l'induction certains cas où le nombre d'états du système est infini. Nous avons effectué une étude de cas sur une version très simplifiée d'un problème de détection d'interaction de services. Il s'agit de combinaisons de services du types *renvoi d'appels* et *filtrage d'appels par liste noire* [276].

5 Actions nationales et internationales

5.1 Actions nationales

Nous participons au projet groupé DEMOTHEO du GDR Programmation et Outils pour l'Intelligence Artificielle, sur le thème contraintes et déduction automatique. Ce projet regroupe les équipes PROTHEO, DEMONS du LRI et des chercheurs de Strasbourg et Rouen. Dans ce contexte, notre projet a participé à une action du GDR Programmation sur le développement d'outils de démonstration pour la vérification de programmes de type industriel. Nous dirigeons également deux groupes de travail du GDR/PRC AMI, le groupe "Déduction et calcul formel" et le groupe "Méthodes effectives de décidabilité".

5.2 Actions internationales

5.2.1 Europe de l'ouest

Nous participons au réseau Capital Humain et Mobilité CONSOLE (CONstraint SOLving in Europe), créé en décembre 1994, sur la résolution de contraintes symboliques. Ce réseau fait intervenir les universités et les centres de recherches en informatique d'Orsay (Université: Jean-Pierre Jouannaud), Barcelone (Université: Robert Nieuwenhuis), Lille (Université: Max Dauchet), Imperial College (Mark Wallace), Padoue (Université: Moreno Falaschi), Saarbrücken (MPI: Harald Ganzinger).

Nous avons participé à un groupe de travail Basic Research Action d'ESPRIT ayant pour acronyme *COMPASS* et titre *a COMPrehensive Algebraic approach to System Specification et development*. Cette action a pris fin le 31/3/96 et nous participons à l'élaboration du nouveau projet intitulé *COFI* (*Common Framework Initiative*), pour définir un langage et une méthode de spécifications formelles issue de la communauté des spécifications algébriques.

Nous participons au groupe de travail Basic Research Action d'ESPRIT qui, sous l'acronyme *CCL* et le titre *Construction of Computational Logics,* fait intervenir les universités et les centres de recherche en informatique de Barcelone (Université : Fernando Orejas), Madrid (Université : Mario Rodriguez-Artalejo), Munich (Université : Tobias Nipkow), Orsay (Université : Jean-Pierre Jouannaud), Saarbrücken (Université : Gert Smolka), Saarbrücken (Max Planck Institut : Harald Ganzinger) ainsi que la société COSYTEC (Mehmet Dincbas). Cette action a été reconduite à compter du 1/9/96 sous l'acronyme *CCL II*.

Nous participons également au réseau d'excellence COMPULOG qui regroupe de nombreux sites européens travaillant sur la *programmation logique*.

Nous avons coordonné le réseau Capital Humain et Mobilité *SOL*, créé en septembre 93, sur la résolution de contraintes en nombres entiers et sur les domaines finis. Cette action a pris fin le 30/4/96.

Nous participons au *ERCIM Working Group on Constraints* dirigé par K. Apt (CWI, Amsterdam).

Nous entretenons par ailleurs des relations étroites avec les universités de Kaiserslautern et de Saarbrücken qui sont des universités proches ayant un grand laboratoire de recherche en informatique, ainsi qu'avec le DFKI et le MPI. Nous travaillons en effet sur des sujets très voisins : la réécriture et la déduction concurrente à Kaiserslautern avec Jürgen Avenhaus et Klaus Madlener, les algèbres à sortes ordonnées et les contraintes associées avec Gert Smolka à Saarbrücken, la déduction automatique avec Harald Ganzinger à Saarbrücken (Max Planck Institut).

5.2.2 Amérique

La coopération NSF-INRIA avec l'Université de l'Illinois à Urbana-Champaign se poursuit sur le thème de la résolution de contraintes et ses applications. Elle implique du côté américain N. Dershowitz, C. Liu et E. Reingold.

Une coopération NSF-CNRS avec l'Université de l'Etat de New York à Stony Brook est engagée sur le thème de la déduction avec contraintes et la parallélisation des preuves. Elle implique L. Bachmair et I.V. Ramakrishnan du côté américain.

5.3 Séjours de chercheurs

Adel Bouhoula a effectué un séjour de neuf mois au SRI International (Menlo Park, Californie). Il y a travaillé, en collaboration avec Jean-Pierre Jouannaud (LRI, université Paris Sud) et José Meseguer (SRI International) un cadre logique nommé *la logique équationnelle avec prédicat d'appartenance* [263]. La sémantique opérationnelle de ce cadre logique ainsi que les techniques de complétion, de complétude suffisante, de preuves par récurrence et de paramétrisation ont été réalisées et seront implantées dans le prouveur *SPIKE*.

5.4 Responsabilités éditoriales et professionnelles

Eric Domenjoud est membre du conseil de laboratoire du CRIN.

Isabelle Gnaedig est chargée de Mission à la Communication pour l'INRIA Lorraine.

Claude Kirchner est vice-président de la Commission d'Evaluation de l'INRIA et membre du Comité des Projets de l'INRIA Lorraine.

Claude Kirchner est membre de la commission de spécialistes (27ème section) des universités de Nancy.

Claude Kirchner est président du comité d'organisation des conférences RTA (Rewriting Techniques and Applications) et membre du comité d'organisation des conférences LICS.

Claude Kirchner a été membre des comités de programme des conférences LICS'96 (27-30 juillet, New Brunswick, New Jersey), DISCO'96 (18-20 septembre, Karlsruhe, Allemagne), First International Workshop on Rewriting Logic and its Applications (3-6 septembre, Pacific Grove, Californie).

Claude Kirchner est membre des comités éditoriaux des revues *Journal of Symbolic Computation*, *Journal of Logic Programming* et *Journal of the Egyptian Mathematical Society*.

Claude Kirchner est membre du conseil d'administration de l'"European Association for Programming Languages and Systems".

Hélène Kirchner est responsable du Comité des Bourses ainsi que membre du Comité des Projets de l'INRIA Lorraine et du conseil de laboratoire du CRIN en tant que responsable de l'axe LPCA.

Hélène Kirchner participe aux Commissions Recherche et Prospectives de SPECIF et à son Conseil d'Administration.

Hélène Kirchner est membre du groupe d'experts GE4 Informatique du MENESR.

Hélène Kirchner a été membre des comités de programmes de AMAST'96, FROCOS'96, JICSLP'96, CAAP'96 (présidente).

Hélène Kirchner est membre du groupe IFIP WG 14.3 (Foundations of Systems Specifications), et du comité éditorial du journal *Annals of Mathematics and Artificial Intelligence*.

Hélène Kirchner est responsable du groupe "Déduction et Calcul Formel" du GDR/PRC AMI depuis janvier 96.

Denis Lugiez est membre élu du conseil national des universités en 27ème section.

Denis Lugiez est responsable du groupe "Méthodes Effectives de décidabilité" du GDR/PRC AMI depuis janvier 96.

Michaël Rusinowitch est vice-président du Comité des Projets de l'INRIA Lorraine. Il est également membre du conseil de laboratoire du CRIN et responsable adjoint du DEA informatique de Nancy.

Michaël Rusinowitch a été membre des comités de programme suivants: International Conference on Automated Deduction (CADE-13), KI German Conference on Artificial Intelligence, IEEE International Conference on Tools with Artificial Intelligence, Deuxième Conférence Nationale sur la Résolution Pratique des Problèmes NP-complets.

6 Diffusion des résultats

6.1 Actions d'enseignement

Plusieurs membres du projet sont intervenus dans le cadre du DEA d'informatique de Nancy :

Cours de logique, Denis Lugiez, Michaël Rusinowitch

Démonstration automatique avancée, Adel Bouhoula

Déduction et Programmation avec Contraintes, Claude Kirchner, Hélène Kirchner

Organisation et encadrement des travaux pratiques autour de *SPIKE* dans le cadre du module *Programmation Avancée* du DU d'Architecture de l'Université Henri Poincaré - Nancy I, Adel Bouhoula

Participation à l'Ecole des jeunes chercheurs du GDR Programmation, Bordeaux, avril 96, cours *Réécriture et Calcul équationnel*, Claude Kirchner

Le projet comprend également des moniteurs et ATERs qui enseignent dans différentes universités de l'académie dans le cadre normal de leurs activités.

6.2 Participation à des colloques, conférences, manifestations diverses

Mis à part les congrès et colloques où furent présentés nos travaux, nous avons participé au cours de l'année aux évènements suivants :

Séminaire de l'AFPL et du groupe de travail AFCET Programmation Logique sur les langages de programmation à contraintes concurrents, Paris, 25 janvier 96, Carlos Castro

Séminaire d'évaluation du programme 5 de l'INRIA, Dourdan, 22-23 février 96, Claude Kirchner

FROCOS'96, Munich, mars 96, Hélène Kirchner

CAAP-ESOP-CC'96, Linköping, Suède, avril 96, Hélène Kirchner

Journées Françaises de Programmation Logique et Programmation à Contraintes, Clermont-Ferrand, 5-7 juin 96, Carlos Castro

FLOCS, 27 juillet-3 août 96, Bernhard Gramlich, Claude Kirchner

First International Workshop on Rewriting Logic and its Applications, septembre 96, Asilomar, Californie, Adel Bouhoula

LACL'96, Nancy, 23-25 septembre 96, Hélène Kirchner

Séminaire d'évaluation du programme 3b de l'INRIA, 24-25 octobre 96, Claude Kirchner

10ème anniversaire du LFCS, Edinburgh, 9 novembre 96, Hélène Kirchner

Groupe de conception du projet européen COFI (Common Framework Initiative), Edinburgh, 10-12 novembre 96, Hélène Kirchner

Journée d'évaluation du GDR Programmation (démonstration du logiciel *SPIKE*), Michaël Rusinowitch

Journées du GDR Programmation (démonstration du logiciel **daTac**), Orléans, novembre 96, Laurent Vigneron

6.3 Conférences invitées, tutoriels, séminaires

Nous avons donné au cours de l'année les séminaires suivants :

FSEG, Université de Sfax, Janvier, 96, Adel Bouhoula

Faculté des Sciences de Tunis, Janvier, 96, Adel Bouhoula

Ecole Supérieure des postes et des télécommunications de Tunis, Janvier, 96, Adel Bouhoula

Prototyping Deduction with constraints in Rewriting Logic, Conférence invitée, Fourth International Symposium on Artificial Intelligence and Mathematics, Fort-Lauderdale, Florida, janvier 96, Hélène Kirchner

On the use of orderings in automated theorem proving, Conférence invitée, American Mathematical Society, Short Course on Artificial Intelligence, Orlando, Florida, janvier 96, Hélène Kirchner

Higher-order unification via explicit substitutions, Université de l'Illinois, Urbana-Champaign, 13 février 96 et Université de Chicago, 14 février 96, Claude Kirchner

Présentation du Groupe Déduction et Calcul Formel, séminaire du GDR-PRC AMI, La Bussière, février 96, Hélène Kirchner

Déduction automatique avec contraintes symboliques dans les théories équationnelles, séminaire du LABRI, Bordeaux, février 96, Laurent Vigneron

Sortes ordonnées et déduction équationnelle, réunions du groupe AMI "esquisses", Jussieu, mars et avril 96, Hélène Kirchner

Déduction automatique modulo E et contraintes, séminaire du CRID, Dijon, avril 96, Laurent Vigneron

SRI International, USA, Mai, 96, Adel Bouhoula

Déduction avec fonctions partielles, appartenances et égalités, séminaire, INRIA-Sophia Antipolis, mai 96, Hélène Kirchner

Démonstration dans une théorie équationnelle et contraintes, séminaire du LIFL, Lille, mai 96, Laurent Vigneron

Automated Study of Rough Algebras, séminaire, Université de Varsovie, mai 96, Laurent Vigneron

Logique de réécriture, systèmes de calcul et ELAN, séminaire, Université de Metz, juin 96, Hélène Kirchner

Toward the Concurrent Implementation of Computational Systems, Conférence invitée, Fifth International Conference on Algebraic and Logic Programming, Aachen, Germany, septembre 96, Claude Kirchner

E-Unification d'ordre supérieur via les substitutions explicites, réunion du groupe AMI "mélange de systèmes de réécriture algébriques et de systèmes logiques", octobre 96, Christophe Ringeissen

Combinaison et intégration d'algorithmes d'unification : premier ordre et ordre supérieur, séminaire du Laboratoire d'Informatique Fondamentale d'Orléans, octobre 96, Christophe Ringeissen

Esquisses et sous-sortes, Journée AMI, Groupe Déduction et Calcul Formel, Paris, décembre 96, Hélène Kirchner

Routage dans les graphes de Cayley, Journées RUMEUR I3S, Sophia Antipolis, décembre 96, Pauline Strogova

Automated Reasoning in Equational Theories, Conférence invitée, Conference on Logic, Algebra and Computer Science, Varsovie, décembre 1996, Laurent Vigneron

6.4 Jurys de thèses

Nous avons pris part aux jurys de thèses et d'habilitations suivants:

Claude Kirchner : Fabio Gadducci (Pise), Ali Amaniss (UHP-Nancy I), Eric Monfroy (UHP-Nancy I), Pauline Strogova (UHP-Nancy I), Frédéric Saubion (Orléans).

Hélène Kirchner : Bruno Salinier (Bordeaux 1), Zhenyu Qian (habilitation, Brême), Malek Mouhoub (UHP-Nancy I), Sébastien Limet (Orléans).

Denis Lugiez : Ali Amaniss (UHP-Nancy I).

Michaël Rusinowitch : Florent Jacquemard (Orsay), Eric Monfroy (UHP-Nancy I), Serge Muller (Nice), Abdenbi Drissi-Talbi (Rouen).

6.5 Diffusion de logiciels

Les logiciels *SPIKE* et ELAN sont accessibles depuis le serveur W3 de l'INRIA-Lorraine.

7 Publications

Livres et monographies

[219] H. KIRCHNER (réd.), *Trees in Algebra and Programming - CAAP 96. Proc. 21st International Colloquium , Linköping, Sweden*, *Lecture Notes in Computer Science*, *1059*, Springer-Verlag, 1996.

Thèses

[220] A. AMANISS, *Méthodes de schématisation pour la démonstration automatique*, thèse de doctorat, Université Henri Poincaré – Nancy 1, septembre 1996.

[221] E. MONFROY, *Collaboration de solveurs pour la programmation logique à contraintes*, thèse de doctorat, Université Henri Poincaré – Nancy 1, novembre 1996.

[222] P. STROGOVA, *Techniques de Réécriture pour le Traitement de Problème de Routage dans des Graphes de Cayley*, thèse de doctorat, Université Henri Poincaré – Nancy 1, décembre 1996.

Articles et chapitres de livre

[223] A. BOUHOULA, «Using Induction and Rewriting to Verify and Complete Parameterized Specifications», *Theoretical Computer Science 170*, décembre 1996, p. 245–276.

[224] B. GRAMLICH, «On Termination and Confluence Properties of Disjoint and Constructor-Sharing Conditional Rewrite Systems», *Theoretical Computer Science 165*, 1, septembre 1996, p. 97–131.

[225] C. LYNCH, «Local Simplification», *Information and Computation*, To appear.

[226] C. LYNCH, «Oriented Equational Logic Programming is Complete», *Journal of Symbolic Computation*, To appear.

[227] P. NARENDRAN, M. RUSINOWITCH, «Any ground associative-commutative theory has a finite canonical system», *Journal of Automated Reasoning 17*, 1, 1996, p. 131–143.

[228] C. RINGEISSEN, «Combining Decision Algorithms for Matching in the Union of Disjoint Equational Theories», *Information and Computation 126*, 2, mai 1996, p. 144–160.

Communications à des congrès, colloques, etc.

[229] F. AJILI, C. KIRCHNER, «Combining Unification and Built-In Constraints», *in : 10th International Workshop on Unification, Munich (Germany)*, K. Schulz, S. Kepser (réd.), juin 1996.

[230] I. ALOUINI, C. KIRCHNER, «Toward the Concurrent Implementation of Computational Systems», *in : Proceedings International Conference on Algebraic and Logic Programming, Aachen (Germany)*, M. Hanus, M. Rodríguez-Artalejo (réd.), *Lecture Notes in Computer Science*, *1139*, Springer-Verlag, p. 1–31, Aachen (Germany), septembre 1996.

[231] L. BACHMAIR, I. V. RAMAKRISHNAN, L. VIGNERON, «Efficient Term Rewriting Techniques», *in : CCL'96 Workshop*, San Lorenzo (Spain), septembre 1996.

[232] N. BERREGEB, A. BOUHOULA, M. RUSINOWITCH, «Automated Verification by Induction with Associative-Commutative Operators», *in : Proceedings of the International conference on Computer Aided Verification*, New Jersey, USA, 1996.

[233] N. BERREGEB, A. BOUHOULA, M. RUSINOWITCH, «SPIKE-AC: a System for Proofs by Induction in Associative-Commutative Theories», *in : Proceedings 7th Conference on Rewriting Techniques and Applications, New Brunswick (New Jersey, USA)*, p. 428–431, 1996.

[234] P. BOROVANSKÝ, C. KIRCHNER, H. KIRCHNER, P.-E. MOREAU, M. VITTEK, «ELAN: A logical framework based on computational systems», *in : Proceedings of the first international workshop on rewriting logic*, J. Meseguer (réd.), *4*, Electronic Notes in Theoretical Computer Science, Asilomar (California), septembre 1996.

[235] P. BOROVANSKÝ, C. KIRCHNER, H. KIRCHNER, «Controlling Rewriting by Rewriting», *in : Proceedings of the first international workshop on rewriting logic*, J. Meseguer (réd.), *4*, Electronic Notes in Theoretical Computer Science, Asilomar (California), septembre 1996.

[236] A. BOUHOULA, K. BSAÏES, «A New Framework for Mechanising Logic Program Diagnosis», *in : Proceedings of the Sixth International Workshop on Logic Program Synthesis and Transformation, Stockholm (Sweden)*, août 1996.

[237] A. BOUHOULA, «A General Framework for Mechanizing Induction using Test Set», *in : Proccedings of the 4th Pacific Rim International Conference on Artificial Intelligence, Cairns (Australia)*, *Lecture Notes in Artificial Intelligence*, *1114*, Springer-Verlag, p. 1–12, août 1996.

[238] C. CASTRO, «Binary CSP Solving as an Inference Process», *in : Proceedings of the Eighth International Conference on Tools in Artificial Intelligence, ICTAI'96, Toulouse, France*, p. 462–463, novembre 1996.

[239] C. CASTRO, «Solving Binary CSP using Computational Systems», *in : Proceedings of the first international workshop on rewriting logic*, J. Meseguer (réd.), *4*, Electronic Notes in Theoretical Computer Science, Asilomar (California), septembre 1996.

[240] R. CURIEN, Z. QIAN, H. SHI, «Efficient Second-Order Matching», *in : Proceedings 7th Conference on Rewriting Techniques and Applications, New Brunswick (New Jersey, USA)*, H. Ganzinger (réd.), *Lecture Notes in Computer Science*, *1103*, Springer-Verlag, p. 317–331, juillet 1996.

[241] E. DOMENJOUD, «Undecidability of the word problem in the union of theories sharing constructors», *in : Proceedings 10th International Workshop on Unification, Munich (Germany)*, K. Schulz, S. Kepser (réd.), p. 92–96, juin 1996.

[242] G. DOWEK, T. HARDIN, C. KIRCHNER, F. PFENNING, «Unification via Explicit Substitutions: The Case of Higher-Order Patterns», *in : Proceedings of Joint International Conference and Symposium on Logic Programming*, M. Maher (réd.), The MIT press, Bonn (Germany), septembre 1996.

[243] T. GENET, I. GNAEDIG, «Termination Proofs with Efficient *gpo* Ordering Constraint Solving», *in : 10th International Workshop on Unification, Munich (Germany)*, K. Schulz, S. Kepser (réd.), juin 1996.

[244] B. GRAMLICH, C.-P. WIRTH, «Confluence of Terminating Conditional Rewrite Systems Revisited», *in : Proceedings 7th Conference on Rewriting Techniques and Applications, New Brunswick (New Jersey, USA)*, H. Ganzinger (réd.), *Lecture Notes in Computer Science*, *1103*, Springer-Verlag, p. 245–259, juillet 1996.

[245] B. GRAMLICH, «Conditional Rewrite Systems under Signature Extensions: Some Counterexamples», *in : Proceedings 10th International Workshop on Unification, Munich (Germany)*, K. Schulz, S. Kepser (réd.), p. 45–50, juin 1996.

[246] B. GRAMLICH, «Confluence without Termination via Parallel Critical Pairs», *in : Proceedings 21st International Colloquium on Trees in Algebra and Programming, Linköping (Sweden)*, H. Kirchner (réd.), *Lecture Notes in Computer Science*, *1059*, Springer-Verlag, p. 211–225, avril 1996.

[247] B. GRAMLICH, «On Proving Termination by Innermost Termination», *in : Proceedings 7th Conference on Rewriting Techniques and Applications, New Brunswick (New Jersey, USA)*, H. Ganzinger (réd.), *Lecture Notes in Computer Science*, *1103*, Springer-Verlag, p. 93–107, juillet 1996.

[248] C. HINTERMEIER, H. KIRCHNER, P. MOSSES, «Combining Algebraic and Set Theoretic Specifications», *in : Recent Trends in Data Type Specification, Proc. 11th Workshop on Specification of Abstract Data Types joint with the 9th general COMPASS workshop. Oslo, Norway, September 1995. Selected papers*, O. M.Haveraaen, O-J.Dahl (réd.), *Lecture Notes in Computer Science*, *1130*, Springer-Verlag, p. 255–273, 1996.

[249] C. HINTERMEIER, H. KIRCHNER, P. MOSSES, «R^n- and G^n-logics», *in : Higher-Order Algebra, Logic, and Term Rewriting*, G. Dowek, J. Heering, K. Meinke, B. Möller (réd.), *Lecture Notes in Computer Science*, *1074*, Springer-Verlag, p. 90–108, 1996.

[250] C. KIRCHNER, C. LYNCH, C. SCHARFF, «A Fine-grained Concurrent Completion Procedure», *in : Proceedings 7th Conference on Rewriting Techniques and Applications, New Brunswick (New Jersey, USA)*, H. Ganzinger (réd.), *Lecture Notes in Computer Science*, *1103*, Springer-Verlag, p. 3–17, juillet 1996.

[251] C. KIRCHNER, C. RINGEISSEN, «Higher Order Equational Unification via Explicit Substitutions», *in : 10th International Workshop on Unification, Munich (Germany)*, K. Schulz, S. Kepser (réd.), juin 1996.

[252] H. KIRCHNER, P.-E. MOREAU, «A reflective extension of ELAN», *in : Proceedings of the first international workshop on rewriting logic*, J. Meseguer (réd.), *4*, Electronic Notes in Theoretical Computer Science, Asilomar (California), septembre 1996.

[253] C. LYNCH, P. STROGOVA, «PATCH Graphs: an Efficient Data Structure for Completion of Finitely Presented Groups», *in : Proc. Third International Conference on Artificial Intelligence and Symbolic Mathematical Computation*, J. Pfalzgraf (réd.), *Lecture Notes in Computer Science*, *1138*, Springer-Verlag, p. 176–190, Steyr, Austria, septembre 1996.

[254] E. MONFROY, C. RINGEISSEN, «Domain-Independent Scheme for Constraint Solver Extension», *in : 2nd IMACS Conference on Applications of Computer Algebra*, RISC (Research Institute for Symbolic Computation), Castle Hagenberg (Austria), juillet 1996.

[255] E. MONFROY, M. RUSINOWITCH, R. SCHOTT, «Implementing Non-Linear Constraints with Cooperative Solvers», *in : Proceedings of ACM Symposium on Applied Computing*, p. 63–72, Philadelphia, PA, février 1996.

[256] E. MONFROY, «Solver Collaborations for Non-Linear Constraints», *in : 2nd IMACS Conference on Applications of Computer Algebra*, RISC (Research Institute for Symbolic Computation), Castle Hagenberg (Austria), juillet 1996.

[257] C. RINGEISSEN, «Cooperation of Decision Procedures for the Satisfiability Problem», *in : Frontiers of Combining Systems, First International Workshop, Munich (Germany)*, F. Baader, K. Schulz (réd.), *Applied Logic*, Kluwer Academic Publishers, p. 121–140, mars 1996.

[258] P. VARMAN, R. VERMA, «Tight Bounds for Prefetching and Buffer Management of Parallel I/O Systems», *in : Proceedings of the Sixteenth Conference on the Foundations of Software Technology and Theoretical Computer Science*, *Lecture Notes in Computer Science*, *1180*, Springer-Verlag, p. 200–211, Hyderabad (India), décembre 1996.

[259] L. VIGNERON, A. WASILEWSKA, «Automated Study of Rough and Modal Algebras», *in : CCL'96 Workshop*, San Lorenzo (Spain), septembre 1996.

[260] L. VIGNERON, A. WASILEWSKA, «Modal and Rough Algebras», *in : Proceedings of the CESA'96 International Multiconference (Computational Engineering in Systems Applications), Symposium on Modelling, Analysis and Simulation*, P. Borne, G. Dauphin-Tanguy, C. Sueur, S. El Khattabi (réd.), IMACS/IEEE, p. 1107–1112, Lille (France), juillet 1996.

[261] L. VIGNERON, «Extended Equations for Deduction modulo E», *in : CCL'96 Workshop*, San Lorenzo (Spain), septembre 1996.

[262] M. VITTEK, «A Compiler for Nondeterministic Term Rewriting Systems», *in : Proceedings 7th Conference on Rewriting Techniques and Applications, New Brunswick (New Jersey, USA)*, H. Ganzinger (réd.), *Lecture Notes in Computer Science*, *1103*, Springer-Verlag, p. 154–168, juillet 1996.

Rapports de recherche et publications internes

[263] A. BOUHOULA, J.-P. JOUANNAUD, J. MESEGUER, «Specification and proof in membership equational logic», *rapport de recherche*, SRI International, Computer Science Laboratory, Menlo Park, CA (USA), 1996.

[264] A. BOUHOULA, J.-P. JOUANNAUD, «Automata-driven automated induction», *rapport de recherche*, SRI International, Computer Science Laboratory, Menlo Park, CA (USA), 1996.

[265] E. MONFROY, C. RINGEISSEN, «Domain-Independent Constraint Solver Extension», *Research Report n°96-R-043*, Centre de Recherche en Informatique de Nancy, Vandœuvre-lès-Nancy, février 1996.

[266] E. MONFROY, «An Environment for Designing/Executing Constraint Solver Collaborations», *Research Report n°96-R-044*, Centre de Recherche en Informatique de Nancy, Vandœuvre-lès-Nancy, février 1996.

Divers

[267] F. AJILI, C. CASTRO, E. DOMENJOUD, C. KIRCHNER, «Rapport de fin de pré-étude du problème d'affectation des courses pour le GIHP-Champagne», Rapport de fin de contrat, novembre 1996.

[268] F. AJILI, E. CONTEJEAN, «Complete Solving of Linear Diophantine Equations and Inequations without Adding Variables», Theoretical Computer Science, To appear.

[269] I. ALOUINI, «RECO: An implementation of conditional Concurrent Rewriting on distributed memory machines», Poster Session of ALP'96, Aachen (Germany).

[270] A. AMANISS, M. HERMANN, D. LUGIEZ, «Set Operations for Recurrent Term Schematizations», To appear in Proc. of International Colloquium on Trees in Algebra and Programming, Lille, 1997.

[271] A. AMANISS, M. HERMANN, D. LUGIEZ, «Travailler avec des ensembles infinis de termes représentés par des termes iterés», Journées du GDR Programmation, Orléans, novembre 1996.

[272] T. GENET, I. GNAEDIG, «Shared Term Data Structure for Efficiently Solving gpo Ordering Constraints», To appear in Proc. of International Colloquium on Trees in Algebra and Programming, Lille, 1997.

[273] T. GENET, I. GNAEDIG, «Preuve de terminaison par résolution de contraintes d'ordre sur une structure partagée», Journées du GDR Programmation, Orléans, novembre 1996.

[274] C. LYNCH, P. STROGOVA, «Graphes PATCH: une structure de données pour la complétion efficace des groupes finiment présentés», Journées du GDR Programmation, Orléans, novembre 1996.

[275] E. PETITJEAN, «Résolution parallèle de contraintes linéaires sur les entiers naturels», Rapport de DEA, septembre 1996, INPL.

[276] S. STRATULAT, «Vérification automatisée de logiciels de télécommunications», Rapport de DEA, septembre 1996, Université Henri Poincaré – Nancy 1.

8 Abstract

The goals of the PROTHEO project are the design and implementation of tools for integrating program developments and proofs of properties, while taking advantage of the complementary nature of automated deduction, logics for programming and constraint solving.
We are interested in three research areas which strongly contribute to the previous objectives: first the problems of constraint satisfaction and solving, second the study of logics for programming or proving including the constraint paradigm, and third the proof of program properties. We are thus working on the following topics: Constraint solving, Rewriting logic, Proofs by induction, Parallelization of deduction processes.
We are developing several constraint solving algorithms and two more general deduction systems, namely *SPIKE* (an inductive theorem prover) and ELAN (a logical framework based on the rewriting logic).