Action Concertée Incitative

SÉCURITÉ & INFORMATIQUE

APRON: Analyse de PROgrammes Numériques

Rapport de mi-parcours (juin 2006)

1 Liste des équipes impliquées

1.1 Équipe ATIP, CRI, EMP

Participants à APRON au 1er mai 2006 :

- François Irigoin, MR École des mines de Paris
- Corinne Ancourt, CR École des mines de Paris
- Duong Nguyen, doctorant depuis 01/31/2002, non financé depuis le 01/31/2005
- Sebastian Pop, doctorant depuis 01/31/2003, bourse École des mines

1.2 Équipe Synchrone, Vérimag

Participants à APRON au 1er mai 2006 :

- Nicolas Halbwachs, DR CNRS
- Laure Gonnord, doctorante depuis octobre 2003, allocataire MENRT
- Mathias Péron, doctorant depuis octobre 2005, BDI CNRS

David Merchat a quitté le projet en juin 2005.

1.3 Projet VERTECS, IRISA

Participants à APRON au 1er mai 2006 :

- Jeannet Bertrand, CR INRIA (projet POP-ART à l'Inria Rhone-Alpes)
- Le Gall Tristan, doctorant, allocataire MENRT, 09/2004

1.4 Équipe Interprétation abstraite et sémantique, École normale supérieure

Participants à APRON au 1er mai 2006 :

- Julien Bertrane, thésard depuis septembre 2005 (bourse AMN),
- Patrick Cousot, Professeur
- Jérôme Feret, Post-doctorant, CDD

- Laurent Mauborgne, MdC
- Antoine Miné, Post-doctorant, CDD
- Xavier Rival, Post-doctorant, CDD

1.5 Équipe sémantique, preuves et interprétation abstraite, École Polytechnique

Participants à APRON au 1er mai 2006 :

- Guillaume Capron, doctorant depuis le 1er octobre 2004, allocation MENRT (projet APRON)
- Radhia Cousot, DR CNRS
- Francesco Logozzo, chercheur CDD
- Élodie-Jane Sims, doctorante depuis le 1er octobre 2002, CDD

2 Changements apportés à l'organisation du projet

Bertrand Jeannet a quitté le projet IRISA VERTECS et a rejoint le projet POP-ART de l'Inria Rhone-Alpes, il reste impliqué dans le projet APRON.

Les participants n'ont pas procédé à une révision officielle des objectifs, mais la meilleure connaissance des outils des uns et des autres nous a conduit à mettre l'accent sur l'interface commune plutôt que sur les benchmarks communs. L'activité benchmark a été relancé en septembre 2005, mais elle nécessitera un effort renouvelé de la part de tous, d'autant plus que l'activité *interface commune* nécessite un travail supérieur aux prévisions.

3 Résumé des principales avancées

Le détail des avancées est donné équipe par équipe, en annexe, pour conserver une taille raisonnable à ce rapport de mi-parcours. Les résumés des principales avancées sont par contre listés ici domaine par domaine, comme dans la figure 1 `http://apron.cri.ensmp.fr/apronV11.ps` de la proposition APRON.

3.1 Outils (ASTRÉE,NBAC, PIPS,...)

Des développements relatifs aux domaines abstraits préexistants et nécessaires au projet ont été réalisés au sein des outils:

- définition d'un environnement de benchmark pour les opérateurs polyédriques (D. Nguyen, EMP), section 6.1, `http://www.cri.ensmp.fr/people/duong/polybench/polybench.html`;
- constitution de plusieurs bases de benchmark (D. Nguyen, EMP), section 6.1;

- benchmarking des implantations d'opérateurs utilisés par l'outil PIPS (D. Nguyen, EMP), section 6.1.
- nouvel outil, Cibai, pour l'analyse et la vérification de programmes orientés objets par interprétation abstraite, section 6.18 (Polytechnique)
- intégration d'une interface de la bibliothèque APRON pour les opérations polyédriques dans GCC [20]

3.2 Amélioration des domaines abstraits existants

Afin d'exploiter et partager au mieux les ressources de chacun des partenaires, les développements suivants ont été réalisés:

- définition d'une interface commune, utilisable pour divers domaines abstraits numériques et par les analyseurs ASTRÉE `http://www.astree.ens.fr/`, NBAC `http://www.inrialpes.fr/pop-art/people/bjeannet/nbac/index.html` et PIPS `http://www.cri.ensmp.fr/people/pips/` (coopération EMP, LIENS, IRISA, VÉRIMAG)
- implantation de cette interface commune pour les polyèdres `http://apron.cri.ensmp.fr/library/` (B. Jeannet, IRISA/VÉRIMAG) et pour les octogones `http://www.di.ens.fr/~mine/oct/` (A. Miné, LIENS)
- factorisation des polyèdres, voir section 6.3 (VÉRIMAG)
- caractérisation expérimentale du calcul des invariants de boucle dans PIPS (F. Irigoin, EMP)

3.3 Nouveaux domaines abstraits

Toujours dans le but d'améliorer les résultats des analyses, de nouveaux domaines abstraits ont été utilisés et développés:

- treillis des adresses, section 6.5 (VÉRIMAG)
- abstraction d'ensemble de fonctions, section 6.7 (IRISA)
- amélioration de la précision et de la robustesse des fonctions de transferts multidomaines par linéarisation, section 6.9 (ENS)
- domaines abstraits paramétrés par les besoins de l'analyse, section 6.14 (ENS)
- analyse de programmes effectuant des opérations de bas-niveau sur la mémoire (utilisation d'unions en C, accès non alignés), section 6.10
- abstractions de propriétés non-linéaires pour des domaines d'application spécifique (filtres numériques linéaires, dérive de la précision), section 6.11 (ENS)
- domaine abstrait pour la logique de séparation, section 6.17 (Polytechnique)
- génération et sélection de tests symboliques, section 6.6 (IRISA)
- vérification de systèmes d'états finis communiquants par des canaux FIFO non bornés, section 6.8 (IRISA)

3.4 Nouvelles analyses

- analyse de systèmes redondants: certification de systèmes synchrones communiquant par produit de domaines (ENS)
- vérification de programmes par optimisation semi-définie, section 6.15 (ENS)
- Analyse statique par interprétation abstraite de propriétés réactives bornées dans le temps, section 6.16 (Polytechnique)

3.5 Adaptabilité

Des stratégies tenant compte du flot de contrôle dans les analyses ont été affinées par:

- utilisation du partitionnement de traces pour retarder les unions, définition de stratégies pour les structures de contrôle usuelles, implantation dans ASTRÉE et expérimentation, section 6.13 (ENS)
- représentation des graphes de contrôle non structurés par des graphes de contrôle structurés non déterministes (EMP)

3.6 Analyses itératives

Afin d'améliorer la précision des résultats, des analyses itératives supplémentaires ont été développées:

- accélération abstraite, section 6.4 (VÉRIMAG)
- calcul itératif de transformers et de préconditions pour les analyses modulaires inter-procédurales, section 6.2 (EMP)

3.7 Expériences et comparaisons

Ce travail n'est pas vraiment entamé, et l'hétérogénéité des outils et des domaines qui a été mis en évidence au fil des réunions du projet rend difficile la constitution préalable d'un ensemble de benchmarks significatifs.

Néanmoins, la base de benchmark polyédral constitué par l'équipe ATIP du CRI à Fontainebleau a pu être utilisé pour valider la factorisation de polyèdre développée par David Merchat à Grenoble.

4 Réalisations effectuées dans le cadre du projet

4.1 Définition d'une interface commune multi-domaine

Un des buts du projet APRON énoncé dans la proposition est la conception d'une interface commune aux différents domaines abstraits existants pour les variables numériques.

En effet un grand nombre de librairies existent, implémentant différents domaines. Malheureusement leur API et les fonctionnalités proposées diffèrent suffisamment pour que changer de domaine abstrait dans un analyseur requière un effort non négligeable.

La conception d'une interface commune a été longue, car il s'agissait de satisfaire les besoins spécifiques de tous les membres du projet, tout en restant simple.

Le sous-groupe d'APRON travaillant sur ce projet s'est réuni en décembre 2004, février, mars, juin, et novembre 2005. Les résultats sont:

- La conception d'une interface C et OCaml pour les domaines abstraits numériques. Cette interface est plus spécifiquement divisée en plusieurs niveaux. Le niveau le plus bas, le niveau 0, est l'interface que doit supporter une librairie cliente, les niveaux au-dessus fournissant des fonctionalités communes à toutes les librairies.

- Un "rationale" qui reflète les discussions qui ont eu lieu et qui justifie les choix effectués.

Le logiciel développé est en cours de dépôt à l'APP et devrait être distribué au mois de juin 2006 sous license LGPL.

Participation aux réunions de spécification et à l'implantation de l'interface commune: l'interfaçage du domaine abstrait des octogones à la bibliothèque multi-domaines est en cours. Cette implantation bénéficiera par rapport à la bibliothèque existante de diverses améliorations en termes de précision et de coût décrites dans [19].

4.2 Réalisation de bibliothèques instantiant cette interface

- Une importante activité d'implémentation (menée essentiellement par l'IRISA):
  - Une librairie C implémentant les fonctionnalités de base de l'interface de niveau de 0 (types de données, représentation et manipulation des coefficients, des expressions, etc...), ainsi que l'interface OCaml correspondante.
  - Une librairie C implémentant les fonctionnalités de niveau 1 (qui rajoute principalement la liaison automatique entre noms/adresses et dimensions physiques), ainsi que l'interface OCaml correspondante.
  - La documentation correspondante a également été rédigée (au format TEXINFO, à partir duquel on peut produire les formats LaTeX, HTML et INFO).
  - 8600 lignes de C ont été écrites, auxquelles il faut ajouter l'interfaçage avec OCaml (2800 lignes de CamlIDL).

- Une telle interface n'est utile que si des librairies clientes s'y conforment. Actuellement, les domaines abstraits sont disponibles pour l'interface commune:
  - La librairie de polyèdres convexes NewPolka;
  - Une librairie d'intervalles (écrite spécifiquement dans le cadre d'APRON)
  - La librairie d'octogones d'Antoine Miné (en cours).

4.3 Organisation de congrès

Organisation scientifique du *First International Workshop on Numerical & Symbolic Abstract Domains*, 21 janvier, 2005, Paris, France. Voir: `http://www.polytechnique.fr/vmcai05/workshop.html`.

5 Publications obtenues dans le cadre du projet

[1] J. Bertrane. Static analysis by abstract interpretation of the quasi-synchronous composition of synchronous programs. In *Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, number 3385 in LNCS, pages 97–112. Springer-Verlag, 2005.

[2] G. Capron. Static analysis of time-bounded reactive properties of boolean symbols. In *Proceedings of the 21st ACM Symposium on Applied Computing (SAC 2006), Dijon, France, April 23-27, 2006*, 2006.

[3] P. Cousot. Parametric abstract domains. In *International workshop on Numerical & Symbolic Abstract Domains (NSAD 2005)*, 2005.

[4] P. Cousot. Proving program invariance and termination by parametric abstraction, lagrangian relaxation and semidefinite programming. In *Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, pages 1–24, Paris, France, LNCS 3385, January 17–19 2005. Springer, Berlin.

[5] Jérôme Feret. The arithmetic-geometric progression abstract domain. In *Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, number 3385 in LNCS, pages 42–58. Springer-Verlag, 2005.

[6] Jérôme Feret. Numerical abstract domains for digital filters. In *International workshop on Numerical & Symbolic Abstract Domains (NSAD 2005)*, 2005.

[7] L. Gonnord and N. Halbwachs. Combining widening and acceleration in linear relation analysis. In *13th International Static Analysis Symposium, SAS'06*, Seoul, Korea, August 2006.

[8] N. Halbwachs, D. Merchat, and L. Gonnord. Some ways to reduce the space dimension in polyhedra computations. *Formal Methods in System Design*, 2006. Accepted for publication.

[9] B Jeannet, D. Gopan, and T. Reps. A relational abstraction for functions. In *Static Analysis Symposium, SAS'05*, volume 3148 of *LNCS*, September 2005.

[10] B. Jeannet, T. Jéron, V. Rusu, and E. Zinovieva. Symbolic test selection based on approximate analysis. In *11th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 3440 of *LNCS*, Edinburgh (UK), April 2005.

[11] T. Legall, B. Jeannet, and T. Jéron. Verification of communication protocols using abstract interpretation of fifo queues. In *Algebraic Methodology and Software Technology, AMAST '06*, LNCS, July 2006.

[12] T. Legall, B. Jeannet, and H. Marchand. Contrôle de systèmes symboliques, discrets ou hybrides. *Technique et Science Informatiques*, 2006. to appear.

[13] R. Leino and F. Logozzo. Loop invariants on demand. In *Proceedings of the the 3rd Asian Symposium on Programming Languages and Systems (APLAS'05), Tsukuba, Japan, November 3-5, 2005*, volume 3780 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[14] F. Logozzo. Cibai: An abstract interpretation-based static analyzer for modular analysis and verification of java classes. Submitted for publication.

[15] L. Mauborgne and X. Rival. Trace Partitioning in Abstract Interpretation Based Static Analyzers. In *European Symposium On Programming (ESOP'05)*, volume 3444 of *LNCS*, pages 5–20, Edimburgh (UK), April 2005. Springer.

[16] D. Merchat. Réduction du nombre de variables en analyse de relations linéaires. Thèse, Université Joseph Fourier, May 2005.

[17] A. Miné. Symbolic methods to enhance the precision of numerical abstract domains. In *Verification, Abstract Interpretation and Model Checking (VMCAI), 2006*, volume 3855 of *Lecture Notes in Computer Science*, pages 348–363. Springer-Verlag, 2005.

[18] A. Miné. Field-sensitive value analysis of embedded c programs with union types and pointer arithmetics. In *Languages, Compilers, and Tools for Embedded Systems 2006 (LCTES)*. ACM Press, June 2006. à paraître.

[19] A. Miné. The octagon abstract domain. *Higher-Order and Symbolic Computation*, 2006. à paraître.

[20] Sebastian Pop, Georges-André Silber, Albert Cohen, Cédric Bastoul, Sylvain Girbal, and Nicolas Vasilache. Graphite : Polyhedral analyses and optimizations for gcc. In *GNU Compilers Collection Developers Summit 2006*, Ottawa, Canada, 28-30 June 2006.

[21] Duong Nguyen Que. *Robust and Generic Abstract Domain for Static Program Analysis: the Polyhedral Case*. PhD thesis, Ecole des Mines de Paris, 2006.

[22] Duong Nguyen Que and Francois Irigoin. Benchmarking polyhedral algorithms: Satisfiability and dual conversion. In *Numerical and Symbolic Abstract Domain*, 21 january 2005.

[23] X. Rival and L. Mauborgne. The trace partitioning abstract domain. *ACM Transactions On Programming Languages And Systems (TOPLAS)*, submitted in 2005. en cours de soumission.

[24] E-J Sims. An abstract domain for separation logic formulae. In *Proceedings of the 1st International Workshop on Emerging Applications of Abstract Interpretation (EAAI06)*, Vienna, Austria, 2006.

[25] E-J. Sims. Extending separation logic with fixpoints and postponed substitution. *Theoretical Computer Science*, 351(2):258–275, 2006.

6 Annexe: présentations plus détaillées des avancées

6.1 Constitution et exploitation d'une base de benchmarks pour les opérateurs polyédriques

Ce travail fait partie de la thèse de Duong Nguyen [21]. Les deux aspects sont traités dans les chapitres 5 et 6. Une partie de ces travaux a été présentée à NSAD05 [22].

6.2 Détection d'invariants affines de boucles par une analyse statique modulaire (EMP)

Ce travail expérimental a pour objet de comprendre comment un calcul direct d'invariant se comporte par rapport aux calculs usuels par points-fixes sur différents types de boucles rencontrés d'une part dans les programmes de calcul scientifiques, et, d'autre part, dans les domaines traités par les autres partenaires du projet APRON.

Ce travail a permis de proposer des extensions de l'algorithme utilisé pour résoudre des difficultés bien caractérisées mathématiquement, et des réductions des effets négatifs de l'analyse modulaire par des approches intra- et inter-procédurale.

Les relations existantes entre cette approche et l'interprétation abstraite usuelle n'ont pas encore été explorées en détail dans le cadre du projet. De même, l'application de cette technique à des benchmarks fournis par les partenaires d'APRON n'a pas encore été effectuée.

6.3 Factorisation des polyèdres (VÉRIMAG)

Ce travail concerne une amélioration importante de l'algorithmique des polyèdres convexes, consistant à détecter qu'un polyèdre est un produit cartésien de polyèdres de dimensions plus petites, et à opérer, autant que faire se peut, sur des polyèdres ainsi factorisés. Le coût des opérations sur les polyèdres étant souvent exponentiel en la dimension de l'espace, cette approche permet souvent des gains de performances importants. Ce travail a fait l'objet de la thèse de David Merchat [16], et un article [8] vient d'être accepté dans le journal "Formal Methods in System Design".

6.4 Accélération abstraite (VÉRIMAG)

Quoique l'approximation due à l'élargissement puisse être arbitrairement raffinée en retardant l'application de l'opérateur d'élargissement, le coût d'une analyse devient rapidement prohibitif avec l'accroissement du retard. Les approches existantes pour améliorer la précision de l'opérateur d'élargissement sur les polyèdres ne sont pas complètement satisfaisantes, dans la mesure où elles ne garantissent pas une meilleure précision de la limite obtenue, et où elles peuvent pénaliser les performances de l'analyse. Nous examinons une optimisation de l'analyse de relations linéaires consistant à calculer, lorsque c'est possible, l'effet exact (abstrait, i.e., sous forme de polyèdre) de certaines boucles d'un programme. Nous avons identifié un certain nombre de situations où ce calcul est possible à moindre coût, assurant ainsi une amélioration de la précision, sans

pénaliser les performances (en fait, en les améliorant souvent significativement). Par ailleurs, cette technique est parfaitement compatible avec l'application de l'élargissement, et ne restreint donc pas l'applicabilité de l'analyse. Ce travail fait l'objet de la thèse de Laure Gonnord, et un article [7] a été publié.

6.5 Treillis des adresses (VÉRIMAG)

Dans de nombreux contextes, et en particulier dans les systèmes sur puce, des objets sont identifiés par des adresses. Ces adresses sont généralement traitées comme des entiers dans le programmes, mais les opérations effectuées sur ces entiers sont souvent limitées (peu de calcul). Par ailleurs, les informations pertinentes sur les adresses, lorsqu'on s'intéresse aux accès aux objets, sont surtout des relations d'égalité et de non-égalité. Nous étudions un domaine abstrait adapté à l'analyse de ces propriétés, qui soit moins puissant et coûteux que les domaines numériques classiques, et qui rende compte des non-égalités. Une première version de ce domaine est en cours d'implémentation en conformité avec l'interface commune APRON. Ce travail fait partie de la thèse de Mathias Péron.

6.6 Génération et sélection de tests symboliques (IRISA)

Nous nous sommes intéressés à la génération automatique de tests pour tester la conformité d'une implémentation boîte noire par rapport à une spécification formelle, et ce dans un cadre symbolique, par opposition à des travaux antérieurs fondés sur des méthodes d'états finis. La principale difficulté abordée est la sélection de cas de tests selon des objectifs de tests, qui sont ici des scénarios que l'on veut observer durant l'exécution du test. Des solutions efficaces à ce problème ont été élaborées dans le cadre de modèles d'états finis, fondé sur le calcul exact de l'ensemble des états coaccessibles depuis les états désignés par l'objectif de test. Nous les avons étendues à des modèles symboliques, en montrant que l'on peut utiliser des analyses de coaccessibilité approchées [10]. Notre seconde contribution a été de formaliser un critère de qualité pour les tests générés (ont-ils été bien sélectionnés), et de formaliser la relation entre la qualité des tests générés et les approximations utilisées dans l'algorithme de sélection.

6.7 Abstraction d'ensemble de fonctions (IRISA)

Nous avons formalisé une analyse de forme (Shape analysis) développée par D. Gopan et T. Reps, dans laquelle on analyse non seulement la structure du tas mémoire durant l'exécution d'un programme, mais aussi la valeur des champs numériques des enregistrements alloués dynamiquement. Il s'avère que cette analyse utilise une nouvelle forme d'abstraction d'ensemble de fonctions. Nous avons formalisé le domaine abstrait correspondant [9], et étudié ses propriétés en terme de préservation de propriétés. Ce nouveau domaine permet d'abstraire des ensembles de fonction de manière plus précise que les méthodes existantes, tout en restant représentable en machine.

6.8 Vérification de systèmes communiquants (IRISA)

Nous nous sommes intéressé à l'analyse de systèmes d'états finis communiquant par des canaux FIFO non bornés [11] . Contrairement à la plupart des travaux existants fondés sur des techniques d'accélération, nous avons appliqué des méthodes d'interprétation abstraite. Nous avons montré que l'utilisation des langages réguliers avec un opérateur d'extrapolation fournit une méthode simple et élégante au problème, et qui se révèle souvent aussi précise que les méthodes existantes, et dans certains cas plus expressive. Dans le cas de plusieurs canaux, la méthode peut être appliquée soit de manière non-relationnelle (les propriétés sur les différents canaux restent indépendantes), soit de manière relationnelle (ce qui augmente l'expressivité des propriétés représentables). Nous étudions actuellement comment étendre cette méthode à des systèmes transmettant des valeurs numériques sur les canaux. Ce travail peut aussi être utilisé pour abstraire des piles d'appel, dans le contexte de programmes (impératifs) récursifs.

6.9 Amélioration de la précision des fonctions de transferts dans les domaines numériques abstraits (ENS)

Un cadre théorique, formalisé par interprétation abstraite [17], fournit des hypothèses suffisantes pour définir des manipulations symboliques et prouver leur sûreté par rapport à la sémantique des entiers, rationnels, réels, mais également des entiers modulaires et nombres à virgule flottante. Deux manipulations spécifiques sont proposées. D'abord une méthode dite de "linéarisation" capable d'abstraire les expressions non-linéaires apparaissant dans les fonctions de transferts en expressions affines avec coefficient intervalles. Ceci étend la champ d'application des domaines relationnels linéaires (tels que les polyèdres ou les octogones) aux programmes non-linéaires. Cette transformation s'accompagne d'une simplification qui améliore sensiblement la précision des domaines non-relationnels (intervalles) ou faiblement relationnels (octogones). Une deuxième méthode symbolique, dite de propagation des constantes symboliques, améliore la précision des fonctions de transferts en découvrant de nouvelles opportunités de simplification. Celle-ci rend l'analyse statique robuste contre plusieurs classes de transformations de programmes couramment utilisées en compilation et optimisation.

6.10 Analyse de programmes effectuant des opérations de bas-niveau sur la mémoire (ENS)

Les opérations à considérer sont, par exemple, l'utilisation de types "union" ou de trans-typage de pointeurs en C [18]. Le but est triple. D'abord, rendre sûres les analyses numériques du type "field-sensitive" en cas d'accès arbitraires à la mémoire (par exemple, des accés non-alignés ou avec chevauchement des octets). Deuxièmement, analyser précisément les idiomes standards liés aux programmes embarqués de bas-niveau (en particulier la programmation de périphériques intelligents par accès directs à des registres en mémoire partagée). Troisièmement, appliquer le travail déjà réalisé sur les domaines numériques à l'analyse de l'arithmétique de pointeurs.

6.11 Abstractions de propriétés non-linéaires pour des domaines d'applications spécifiques (ENS)

Les domaines linéraires (intervalles, octogones, polyèdres) sont beaucoup utilisés dans les analyseurs généralistes. Ces domaines offrent un bon compromis entre complexité et précision. Mais ils ne peuvent exprimer les propriétés inductives qui sont requises pour borner certaines variables dans des applications plus spécifiques.

Nous avons proposé des domaines abstraits pour deux applications particulières :

- Nous avons défini dans [6] une famille de domaines abstraits pour gérer les filtres numériques linéaires. Dans de telles applications, bien que les fonctions de transferts soient linéaires, les invariants inductifs ne le sont pas. Pour résoudre ce problème, nous séparons le comportement idéal des filtres des erreurs d'arrondi. Puis, nous représentons symboliquement la contribution due au passé immédiat ; la contribution due passé lointain et aux erreurs d'arrondi est bornée en utilisant des intervalles et des ellipses. Nous étendons actuellement ce cadre de travail aux filtres non-linéaires (tels que les systèmes différentiels sigmoïdaux).

- Nous avons proposé dans [5] un domaine abstrait pour borner la dérive des calculs qui divergent lentement. Pour cela, nous majorons l'effet d'une itération de la boucle principale sur chaque variable par une transformation affine. Ainsi, nous obtenons une borne qui dépend exponentiellement du nombre de tours effectués dans la boucle principale.

6.12 Certification de systèmes synchrones communiquant (ENS)

Afin de vérifier l'apport de la redondance dans les systèmes embarqués construits autour de systèmes synchrones, nous essayons de prouver que [1]:

- si plusieurs systèmes similaires obtiennent des résultats proches, un consensus sera trouvé.

- si il y a une divergence entre les systèmes redondants, elle sera détectée.

Les systèmes synchrones disposent chacun d'une horloge indépendante, qui sera supposée imparfaite. Ils peuvent donc se désynchroniser. Cela engendre un nombre infini (et non dénombrable) de possibilités d'enchaînements des événements, que l'on ne peut pas tous tester. Nos analyses sont donc effectuées dans le cadre de l'Interprétation Abstraite, à partir d'un produit réduit des domaines abstraits suivants :

- le domaine des contraintes, qui exprime certains prédicats temporels du premier ordre ,

- le nombre de changements de valeurs d'une variable au cours du temps.

D'autres domaines, bornant par exemple l'intégrale de la valeur d'une variable au cours du temps, sont en cours de développement.

6.13 Partitionnement de traces (ENS)

Nous avons défini un cadre de travail pour le partitionnement de traces, utilisant le flot de contrôle [15, 23]. Notre approche consiste à distinguer les traces d'exécution selon le chemin de contrôle qu'elles ont suivi avant d'appliquer une abstraction d'état classique (comme une abstraction numérique). Un exemple très simple d'application de cette technique consiste à retarder le calcul d'unions abstraites lors de l'analyse statique au-delà de la fin d'une structure conditionnelle. Nous avons mis au point des stratégies de partitionnement adaptées aux structures de contrôle courantes (boucles, conditionnelles, appels de fonctions) et utilisant les valeurs de variables. Nous avons implémenté et testé cette technique [dans le cadre de l'analyseur Astrée] ; ainsi, nous avons pu en éprouver la précision et l'efficacité : le partitionnement permet l'analyse sans fausse alarmes de programmes où une analyse "classique" lève des milliers d'alarmes, tout en réduisant le temps d'analyse.

6.14 Domaines abstraits paramétrés (ENS)

Il s'agit de domaines abstraits qui permettent d'exprimer des contraintes sur des valeurs dont la forme syntaxique est fixée en fonctions de paramètres à déterminer par l'analyse [3]. Les paramètres peuvent être calculés soit par les méthodes traditionnelles d'itération avec accélération de la convergence ou par résolution directe en utilisant des solveurs numériques ou symboliques.

6.15 Vérification de programmes par optimisation semi-définie (ENS)

Pour vérifier des programmes semi-algébrique, nous proposons d'automatiser la méthode de preuve de Floyd/Naur/Hoare, avec pour tâche principale d'engendrer automatiquement des invariants et des fonctions de terminaison [4].

Nous exprimons la sémantique des programmes sous forme polynomiale et abstraions les fonctions de terminaison et les invariants sous forme paramétrique. L'implication dans la méthode de Floyd/Naur/Hoare est abstraite en des contraintes numériques par relaxation lagrangienne. La quantification universelle restante est alors traitée par programmation semi-définie. Finalement les paramètres sont calculés en utilisant des optimiseurs. Cette approche utilise les récents progrès en optimisation de contraintes matricielles linéaires ou bilinéaires utilisant des méthodes primales/duales de points intérieurs généralisant celles bien connues en programmation linéaire à l'optimisation convexe. Cette approche s'applique à la correction totale des programmes impératifs qu'ils soient séquentiels, non-déterministes, parallèles avec équité et s'étend à d'autres propriétés de sûreté et de vivacité.

6.16 Analyse statique par interprétation abstraite de propriétés réactives bornées dans le temps (Polytechnique)

Le but de ce travail est la vérification de propriétés réactives sur des macros-définitions à partir de spécifications formelles [2]. Ces macro-définitions sont utilisées pour construire des programmes embarqués à partir de spécifications

haut-niveau. Les propriétés étudiées faisant intervenir le temps et des variables entières et réelles, il est nécessaire d'utiliser des domaines numériques. Ces derniers sont associés à une représentation compacte et finie des ensembles infinis de traces (schémas d'arbres) afin de tester l'inclusion entre la sémantique de la macro-définition et sa spécification. Le domaine utilisé est principalement celui des polyèdres, mais pour certaines propriétés simples, on peut aussi utiliser les octogones voire simplement les intervalles pour certaines bornes de compteurs utilisés dans le code.

6.17 Logique de séparation (Polytechnique)

Un domaine abstrait pour la logique de séparation destiné à des analyses statiques par interprétation abstraite a été introduit. Plus précisément, c'est une extension de la logique de séparation avec points fixes [25] qui est considérée. Le domaine incorpore des informations de partage (*alias*) et de forme *shape*. Comparé aux domaines habituels de "shape"-graphes, l'originalité principale réside dans un traitement identique de toutes les valeurs (numériques, locations du tas, locations non allouées, nil,...), ainsi nous pouvons avoir des noeuds résumé (*summary nodes*) numériques [24]. Pour garder le domaine aussi général que possible, il est paramétré par un domaine numérique abstrait qui peut être spécialisé selon les besoins. Sont introduits, une sémantique en terme d'ensembles de mémoires (le modèle habituel pour la logique de séparation) et des fonctions de transfert correctes sur le domaine, y compris un élargissement et une union dont la précision/le coût peuvent être modulés selon les besoins du contexte. Ainsi que des fonctions correctes et approximantes pour la traduction de la logique vers le domaine.

6.18 Analyse et vérification de programmes orientés objets par interprétation abstraite (Polytechnique)

Un outil, Cibai [14], d'analyse statique par interprétation abstraite de classes Java a été développé. L'outil analyse des classes sans avoir à produire des annotations et sans aucune interaction avec le programmeur. Il infère automatiquement des invariants de classes, des invariants de boucle et des postconditions de méthodes. Les propriétés inférées ont pour but de vérifier l'absence d'erreurs à l'exécution dans les classes. Cibai peut vérifier

1. l'absence de division par zéro,
2. l'absence d'erreurs d'accès en dehors des bornes de tableaux,
3. l'absence de pointeurs nuls et
4. des assertions booléennes et entières simples fournies par l'utilisateur.

Les domaines abstraits utilisés par Cibai inclurent une extension du domaine des octogones pour pouvoir gérer des objets et des allocations mémoire dynamiques. Par conséquent, Cibai est le premier outil pour l'analyse de langages objets capable d'inférer des invariants numériques. L'outil a été intégré dans un démonstrateur automatique [13] pour automatiser une partie de la vérification interactive de programmes objets. La technique comprend un partitionnement

de traces où le démonstrateur de théorèmes détermine automatiquement sur quel sous-ensemble de traces l'analyse statique doit porter.