

1 Présentation de l'unité de recherche

1.1 Composition de l'unité

1.1.1 Sémantique et interprétation abstraite

- Responsable : Patrick Cousot (Prof. Paris 9 détaché ENS et École Polytechnique)
- Membres :
 - Régis Cridlig (Ingénieur du Corps de l'Armement option Recherche)
 - Éric Goubault (Ingénieur du Corps des Mines)
 - Philippe Granger (Chercheur contractuel)
 - Klaus Havelund (Boursier du Danemark)
 - Bruno Monsuez (Boursier de l'École Polytechnique)

2 Description des thèmes de recherche

2.1 Sémantique et interprétation abstraite

2.1.1 Introduction

Le thème de recherche concerne les méthodes de définition des sémantiques des langages de programmation qui servent de fondements logiques aux méthodes de spécification, de preuve et d'analyse sémantique de programmes par interprétation abstraite. Nous nous intéressons aux formalismes qui s'appliquent aussi bien aux langages impératifs que fonctionnels, logiques, parallèles, etc. En partant si possible d'un modèle opérationnel, notre approche consiste à dériver la hiérarchie des autres sémantiques par des procédés constructifs d'approximation discrète.

L'étude des méthodes de preuve est une étape intermédiaire dans notre démarche conçue comme l'approximation de la sémantique d'un programme relative à une classe donnée de propriétés (invariance, fatalité, congruence observationnelle, etc.).

L'analyse sémantique de programmes consiste à déterminer automatiquement les propriétés d'un programme choisies dans une classe donnée. Nous procédons par interprétation abstraite, comprise comme méthode d'approximation effective de la sémantique.

2.1.2 G^∞ SOS

G^∞ SOS [CC92] est une méthode de spécification de sémantiques par système de règles qui généralise SOS de Plotkin en utilisant des systèmes itérés bien-fondés de définitions bi-inductives. Elle permet de décrire des objets finis ou infinis. En particulier G^∞ SOS permet de décrire simplement et de manière uniforme les comportements finis et infinis des programmes, ce qui est indispensable, par exemple, pour dériver une sémantique dénotationnelle, une analyse de nécessité ou exprimer les hypothèses d'équité dans un programme parallèle. Nous avons étudié les méthodes de preuves de propriétés des objets spécifiés en G^∞ SOS et montré qu'elles généralisent les méthodes classiques de preuves de programmes. Nous avons montré comment, par interprétation abstraite, dériver des spécifications approchées et illustré le procédé pour dériver des hiérarchies de sémantiques abstraites à partir d'une description opérationnelle des comportements finis ou infinis possibles. En particulier la sémantique relationnelle généralise la sémantique "big-step" de Plotkin en incorporant les comportements infinis.

2.1.3 Sémantiques du parallélisme

Les sémantiques du parallélisme se composent de plusieurs familles. Il y a principalement comme pour les langages séquentiels, les sémantiques opérationnelles et dénotationnelles. À cela il faut ajouter les qualificatifs, sémantiques par entrelacements ou sémantiques “vraiment” parallèles.

Les sémantiques opérationnelles usuelles (SOS...) font partie de la famille sémantique opérationnelle par entrelacement. Elles simulent le parallélisme par toutes les possibilités combinatoires d'exécution sur un processeur. Elles confondent donc non-déterminisme et parallélisme. On a introduit (sur des idées provenant originellement de Vaughan Pratt) une sémantique opérationnelle vraiment parallèle, distinguant donc non-déterminisme d'avec le parallélisme, qui peut se comprendre comme étant une généralisation des systèmes de transitions à des systèmes contenant des transitions entre transitions . . . c'est à dire des systèmes de transitions, ou automates, de dimension supérieure. Ces automates se doivent donc d'être vus comme des formes de dimension plus ou moins élevée sur lesquelles on a des actions du temps. Une formulation algébrique est obtenue à partir de la notion d'algèbre homologique de complexe de complexes.

L'étude d'un certain nombre d'invariants algébriques (ou topologiques avec l'intuition géométrique que l'on a de ces objets) conduit à la classification d'un certain nombre de phénomènes utiles pour l'étude de programmes. La déformation des chemins (homotopie) est à relier avec la notion de séquentialisation, quant aux deux foncteurs d'homologie que l'on est très naturellement amené à définir, ils classifient les branchements et les confluences, et donc permettent de donner des conditions (nécessaires ou suffisantes) pour vérifier des équivalences sémantiques.

On a également suffisamment de structure (algébrique) pour être en mesure de définir de façon dénotationnelle des langages en utilisant comme dénotations, ces automates de dimension supérieure. Utilisant la théorie de l'interprétation abstraite, on espère pouvoir donner des analyses statiques compositionnelles de propriétés dynamiques complexes. On pense en particulier à des problèmes d'allocation dynamique de processus.

À terme, on souhaiterait également dégager une notion de parallélisme indépendante de tout présupposé tel entrelacement (un processeur) ou tel “vrai” parallélisme (une infinité de processeurs). Dans la tradition sémanticienne, ceux-ci sont en effet des hypothèses externes : on a déjà restreint la classe de propriétés que l'on peut étudier à une certaine catégorie de machines. On pense ici utiliser des notions abstraites de symétrie, et de préciser les comportements sur des types de machines en utilisant certaines représentations (au sens mathématique du terme) de ces groupes de symétries.

2.1.4 Algèbre de processus et le “Fork Calculus”

Nous concevons une théorie des systèmes communicants basée sur le calcul FC (“Fork Calculus”) pour une algèbre de processus de la famille de CCS, mais qui en diffère par le mode de création de processus parallèles. CCS dispose d'un opérateur binaire parallèle $|$ et deux processus p et q sont exécutés en parallèle par $p|q$. En FC, cet opérateur binaire a été remplacé par un opérateur unaire $\text{fork}(p)$ qui permet d'activer le processus p en parallèle avec le reste du programme [HL93].

Nous avons donné une sémantique opérationnelle pour ce calcul et étudié des notions d'équivalence faible et forte. Notre principale contribution a été de fournir une caractérisation simple et directe des congruences induites par ces équivalences standards. Contrairement aux algèbres de processus classiques, l'équivalence forte n'est pas une congruence par rapport aux opérateurs de FC. De plus une axiomatisation correcte et complète de la congruence forte induite a été produite qui met

en jeu des lois correspondant aux règles classiques d'expansion.

Ce travail sur le calcul FC trouve son origine dans l'étude des propriétés sémantiques du langage de programmation CML (Concurrent ML) qui est une extension de ML avec des primitives de concurrence et communication à la CCS. Le modèle du parallélisme utilisé en CML est celui de l'exécution concurrente d'expressions qui communiquent par échange synchrone de messages sur des canaux typés. Le parallélisme s'obtient par un opérateur similaire au "fork" de FC. L'étude sémantique de CML devrait nous permettre de définir une notion d'équivalence entre expressions CML.

2.1.5 Fondements de l'interprétation abstraite

Notre travail sur les fondements de l'interprétation abstraite concernent l'étude de la relation entre une sémantique concrète (ou exacte) et une sémantique abstraite (ou approchée). Il concerne à la fois l'approche dite des correspondances de Galois dans laquelle nous avons cherché à relaxer les conditions d'existence d'une meilleure approximation (ce qui fait l'objet d'un article dans le numéro spécial du "Journal of Logic and Computation" sur l'interprétation abstraite [CC92b]) et celle dite de l'élargissement/rétrécissement (widening/narrowing) pour le choix d'approximations minimales et l'accélération de la convergence par extrapolation (qui fait l'objet d'un papier invité à PLILP'92 [CC92]).

On a également étudié des améliorations générales des analyses statiques faisant appel à des itérations locales, et dont l'emploi s'est avéré fondamental pour la précision des résultats en pratique [Gra92].

2.1.6 Interprétation abstraite de programmes logiques

En ce qui concerne les applications de l'interprétation abstraite à l'analyse de programmes logiques, nous avons montré comment appliquer le cadre formel exposé ci-dessus, ce qui nous a conduit à proposer la combinaison itérée d'analyses de haut en bas et d'analyses de bas en haut ainsi que l'utilisation de domaines de propriétés infinis. Ce travail a été présenté dans une article paru dans le numéro spécial du "Journal of Logic Programming" sur l'interprétation abstraite [CC92a].

2.1.7 Analyse de nécessité par interprétation abstraite

Nous avons montré que l'analyse de nécessité (strictness analysis) peut se déduire d'une sémantique relationnelle (approchant elle-même une sémantique opérationnelle) par interprétation abstraite avec correspondance de Galois, ce qui n'est pas possible en partant d'une sémantique dénotationnelle [CC93].

Les rapports entre l'analyse de strictness par interprétation abstraite et l'analyse par projections de Wadler et Hughes ont été étudiés [Cri93]. En particulier il a été montré comment, au moyen d'une sémantique approchée basée sur un treillis de propriétés projectives, on peut expliquer par interprétation abstraite de programmes l'analyse par projections et en dériver les règles d'analyse permettant de calculer de manière effective des propriétés d'invariance des programmes fonctionnels du premier ordre dont celles de strictness pour les fonctions sur des domaines de valeurs plats mais aussi paresseux.

Un important corollaire de ces travaux est l'équivalence entre l'analyse de strictness traditionnelle au moyen des idéaux de Scott et l'analyse disjonctive utilisant les *smash*-projections. Un autre sujet d'intérêt est l'étude des sous-catégories fermées de PERs qui pourraient généraliser les propriétés projectives à l'ordre supérieur.

2.1.8 Inférence de type par interprétation abstraite

Traditionnellement, le typage est une phase conceptuelle distincte de la définition de la sémantique dynamique du langage. Un système d'inférence de types peut se décomposer globalement comme suit : tout d'abord une algèbre de types, ensuite un ensemble de règles logiques assignant à chaque expression du langage un type, enfin un algorithme de reconstruction des types calculant pour chaque expression le type le plus général. Classiquement cet algorithme est basé sur l'algorithme d'unification de Robinson.

Une autre approche consiste à utiliser le formalisme de l'interprétation abstraite. Les travaux d'Alan Mycroft et de Neil Jones ont, en 1984, ouvert la voie. Cependant, certaines difficultés font que ces travaux ne connaissent qu'un relatif succès dans le typage de Prolog. Dans le domaine des langages fonctionnels, ces travaux n'ont connu, à quelques exceptions près, que peu de développement.

Notre travail a consisté à développer une méthode générique de construction de systèmes de types par interprétation abstraite [Mon91, Mon92]. De plus, nous montrons que le système d'Hindley-Milner peut être formulé en utilisant les techniques de l'interprétation abstraite. En conséquence, nous montrons comment les techniques d'interprétation abstraite peuvent permettre d'améliorer un tel système de types. Tout d'abord en exhibant des opérateurs d'élargissement plus permissif que celui de Milner et autorisant certains types polymorphes définis de manière récursive. Ensuite, en montrant comment il est possible d'adapter le système de types à une sémantique paresseuse du langage ML, de manière à ce que la β -réduction soit préservée [Mon93].

De même nous avons montré comment introduire de nouvelles algèbres de types [Mon92] permettant de typer de manière décidable des expressions comme $\lambda x.xx$ ou même les opérateurs de point fixe Y ou θ .

Un autre intérêt de cette approche novatrice est de pouvoir combiner l'analyse de types avec d'autres types d'analyse. Nous développons actuellement un système de types pour un langage ML parallèle autorisant l'existence de canaux non typés.

À ce jour, nous nous intéressons à une extension de notre formalisme à des propriétés différentes de celles des types. Nous avons en développement un système d'inférence des propriétés de nécessité pour les fonctions d'ordre supérieur à l'aide des méthodes initialement conçues pour les systèmes de types.

2.1.9 Analyse de programmes parallèles par interprétation abstraite

Après nous être penchés sur différents modèles de coordination pour les langages parallèles (variables partagés et envois de messages), nous nous sommes intéressés au paradigme issu du langage Linda qui généralise facilement la plupart des autres approches. Il est basé sur un espace commun contenant des structures de données distribuées qui peuvent être créées, lues et importées par des processus concurrents de façon asynchrone.

Nous avons défini un langage dérivé du paradigme Linda dont les processus séquentiels sont modélisés sur les réductions du λ -calcul. Il en a été donné une sémantique opérationnelle par entrelacements sur un multi-ensemble de processus ; si l'on considère plusieurs multi-ensembles typés (ou *bags*) de structures de données et de processus actifs distribués, cela donne une combinaison intéressante d'un langage fonctionnel (ML) avec le modèle Linda.

Pour pouvoir analyser la coordination des processus à travers un espace commun, nous avons donné une sémantique "vraiment parallèle" à une algèbre de processus, le Linda-Calcul, qui décrit uniquement le comportement observable de manière externe des processus concurrents [CG93]. Cette sémantique ouvre la voie à la définition d'interprétations abstraites de programmes parallèles afin d'obtenir des informa-

tions statiques sur la synchronisation et la sérialisation (servant à l'allocation et la répartition des tâches), et sur les communications inter-processus qui peuvent être implémentées efficacement selon les cas par envois de messages ou par des structures de données partagées.

2.1.10 Analyses de congruence

On a d'une part achevé de développer toute une série d'analyses nouvelles conçues pour découvrir des propriétés de congruence vérifiées par les variables numériques des programmes (par exemple, propriétés du type :

$$\sum_{i=1}^n a_i x_i \equiv b \pmod{m}$$

où les x_i sont n variables entières (voire rationnelles), et les a_i , b et m sont des coefficients découverts par l'analyse) [Gra91, Gra91]. Celles-ci admettent de nombreuses applications pratiques (telles que la vectorisation automatique).

2.1.11 Perspectives

L'interprétation abstraite a donné naissance à un domaine de recherche actuellement en plein essor en Europe, aux États-Unis, en Australie et au Japon comme le montre par exemple les numéros spéciaux des revues "Journal of Logic Programming" et "Journal of Logic and Computation" qui y sont consacrés. Les applications sont nombreuses qu'elles soient théoriques dans le domaine de la sémantique des langages ou des preuves de programmes ou pratique dans les domaines de la mise au point, la compilation optimisée, la vectorisation et parallélisation des programmes. Nous contribuons à en développer les bases théoriques et les applications.

3 Collaborations

3.1 Sémantique et interprétation abstraite

- Projet MRT avec Bull et l'IMAG ;
- Projet et groupe de travail ESPRIT "Sémantique" avec l'Imperial College, et les Universités de Glasgow, Copenhague et Aarhus ;
- P. Cousot est membre des groupes de travail WG 2.3 et WG 2.4 de l'IFIP.

4 Missions et exposés dans des séminaires

4.1 Sémantique et interprétation abstraite

- P. Cousot : *Specification of infinite processes in G^∞ SOS*, IFIP working group 2.3, Pouilly en Auxois, 30 septembre – 4 octobre 1991.
- P. Cousot (avec R. Cousot) : *Relational Abstract Interpretation of Higher Order Functional Programs*, in: Actes JTASPEFL'91 "Analyse statique en programmation équationnelle, fonctionnelle et logique", Bordeaux, 9–11 octobre 1991, BIGRE 74, Octobre 1991, pp 33-36.
- P. Cousot (avec R. Cousot) : *Comparison of the Galois connection and widening/narrowing approaches to abstract interpretation*, in: Actes JTASPEFL'91 "Analyse statique en programmation équationnelle, fonctionnelle et logique", Bordeaux, 9–11 octobre 1991, BIGRE 74, Octobre 1991, pp 107–110.

- P. Cousot (avec R. Cousot) : *Fixpoint extrapolation techniques et Relational semantics of higher-order functions*, Esprit BRA “Sémantique” meeting, Mont Saint Michel, 27 octobre–2 novembre 1991.
- P. Cousot : *Formal derivation of a denotational semantics from a G^∞ SOS semantics by abstract interpretation*, IFIP working group 2.4, Madrid, 18–22 mai 1992.
- P. Cousot (avec R. Cousot) : *Constructing Hierarchies of Semantics by Abstract Interpretation*, conférence invitée à WSA’92, Bordeaux, 23–25 septembre 1992 ; séminaire à l’Université de Padoue, 1992.
- R. Cridlig : *présentation du compilateur Camlot*, séminaire CAML, INRIA-Rocquencourt, octobre 1991 ; séminaire “Functional Languages”, Cambridge, juin 1992 ; séminaire “Sémantique et interprétation abstraite”, ENS, février 1993.
- R. Cridlig : *analyse par projections*, séminaire “Sémantique et interprétation abstraite”, ENS, décembre 1993.
- R. Cridlig : communication [Cri92] à l’“ACM Sigplan ML Workshop on ML and its Applications”, San Francisco, juin 1992.
- R. Cridlig : participation à LFP, San Francisco, juin 1992.
- R. Cridlig : participation à WSA ’92, Bordeaux, septembre 1992.
- R. Cridlig : visite du laboratoire d’informatique de l’Imperial College à Londres, avril 93.
- R. Cridlig : invitation à la réunion Esprit *Sémantique* au Mont Saint-Michel en septembre 1991.
- Éric Goubault : *Higher-dimensional automata*, séminaire “Theory and Formal Methods”, Imperial College (Londres), septembre 1991.
- Éric Goubault : *Réécriture de graphes*, séminaire École des Mines/Paris VI, avril 1992.
- Éric Goubault : *More on higher-dimensional automata*, séminaire “Theory and Formal Methods”, Imperial College (Londres), septembre 1992.
- Éric Goubault : *Parallélisme, automates et topologie algébrique*, séminaire “Sémantique et interprétation abstraite”, ENS, novembre 1992.
- Éric Goubault : *Calcul des fractions et théorie de l’homotopie*, groupe de travail Topologie Algébrique de l’ENS, trois exposés, janvier 1993.
- Éric Goubault : *Automates de dimension supérieure*, séminaire LITP/Orsay, mai 1993.
- Éric Goubault : communication [GH93] au “Symposium on Term Graph Rewriting” à l’Université Catholique de Nimègue (Hollande), décembre 1991.
- Éric Goubault : communication [GJ92] à la conférence “Third International Conference on Concurrency Theory” à l’Université de New York à Stony Brook, août 1992.
- Éric Goubault : invitation aux réunions Esprit *Sémantique* sur l’Ile de Barra (Grande Bretagne) en juin 1991, au Mont Saint-Michel en septembre 1991.

- Éric Goubault : participation à un workshop à Dagstuhl (Allemagne), avril 1992.
- Éric Goubault : invitations à l’Imperial College (Londres) en septembre 1992 et avril 1993.
- P. Granger : séminaire INRIA Rocquencourt, avril 1991.
- P. Granger : séminaire École des Mines de Paris, septembre 1991.
- P. Granger : séminaire Carnegie Mellon University, janvier 1992.
- P. Granger : exposé aux journées Esprit II Basic Research Action 3124 “*Sémantique*”, Barra Island, avril 1991 et Mont Saint-Michel, Novembre 1991.
- P. Granger : présentation de [Gra91] à la “Fourth International Joint Conference on the Theory and Practice of Software Development”, Brighton, Royaume-Uni, avril 1991.
- P. Granger : présentation de [Gra92] à la “12th Foundations of Software Technology and Theoretical Computer Science Conference”, New Delhi, Inde, Décembre 1992.
- K. Havelund : participation à la conférence FORTE’93 (France).
- K. Havelund : invitation en diverses occasions à l’Université de Aalborg (Danemark) et Edimbourg (Écosse).
- K. Havelund : communication [HL93] à ICALP’93.
- B. Monsuez : *Polymorphic Typing by Abstract Interpretation*, “Theory Seminar”, Université de Cambridge, 5 juin 1992.
- B. Monsuez : *Inférence de types par interprétation abstraite*, séminaire “Sémantique et interprétation abstraite”, ENS, 23 octobre 1992.
- B. Monsuez : *Inférence de types pour langages fonctionnels parallèles*, séminaire “Sémantique et interprétation abstraite”, ENS, 5 mars 1993.
- B. Monsuez : *Inférence de types par interprétation abstraite*, séminaire du “groupe logique” de Nancy, 1^{er} avril 1993.
- B. Monsuez : présentation de [Mon91] à JTASPEFL’91, Bordeaux, octobre 1991.
- B. Monsuez : présentation de [Mon92] à FST&TCS, New Dehli, Inde, décembre 1992.
- B. Monsuez : présentation de [Mon92] à WSA 92, Bordeaux, octobre 1992.

5 Diffusion de la recherche

5.1 Sémantique et interprétation abstraite

- P. Cousot organise un séminaire bi-mensuel “Sémantique et interprétation abstraite” à l’ENS.
- Une réunion scientifique du BRA “Sémantique” a été organisée au Mont Saint Michel du 27 octobre au 2 novembre 1991.

6 Participation à l'évaluation de la recherche

6.1 Sémantique et interprétation abstraite

- P. Cousot est évaluateur des soumissions de projets auprès de la Commission des Communautés Européennes et des projets Esprit BRA Semagraph, Spec, React et Procos.
- P. Cousot est membre des comités de programmes ICCL'92, PEPM'93, FMP'93 et président de WSA'93.
- P. Cousot est membre du Comité Scientifique du LISI (Laboratoire d'Ingénierie des Systèmes d'Information, INSA Lyon et Université Claude Bernard-Lyon I).

7 Activités d'enseignement

7.1 Second cycle

- Magistère de Mathématiques Fondamentales et Appliquées à l'Informatique (P. Cousot, R. Cridlig, B. Monsuez).
- École Polytechnique (P. Cousot, R. Cridlig, B. Monsuez).

7.2 Troisième cycle

- DEA "Informatique, Mathématiques et Applications" (X-ENS-Paris 6, 7 et 11, P. Cousot, responsable).

8 Distinctions honorifiques

9 Bibliographie

Références

Livres

- [Cou92a] ← P. Cousot. *Algorithmique et programmation en Pascal (cours)*, volume 1 of *Cours de l'École Polytechnique*. Ellipses, Paris, 1992.
- [Cou92b] ← P. Cousot. *Algorithmique et programmation en Pascal (exercices et corrigés)*, volume 2 of *Cours de l'École Polytechnique*. Ellipses, Paris, 1992.
- [Hav92] K. Havelund. *RSL Tutorial*, pages 9–250. The BCS Practitioner Series. Prentice Hall, 1992. The written material is part I of the book named 'The RAISE Specification Language'. Part II of the book (pages 251–369) is the 'RSL Reference Description' and is written by A. Haxthausen and K. Havelund in collaboration.

Articles dans des revues internationales avec comité de lecture

- [BHH92] D. Bjørner, A. Haxthausen, and K. Havelund. Formal, Model-oriented Software Development Methods: From VDM to ProCoS and from RAISE to LaCoS. *Future Generation Computer Systems, North-Holland*, 7 (1991/1992), 1992. 28 pages.
- [CC92a] P. Cousot and R. Cousot. Abstract interpretation and application to logic programs. *Journal of Logic Programming*, 13(2–3):103–179, 1992.
- [CC92b] P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, August 1992.
- [GLB92] ← É. Goubault and C. Le Bris. Groupes Ext^1 et déformations de représentations de certaines algèbres de Lie nilpotentes. *Bulletin des Sciences Mathématiques*, 116(4):465–486, 1992.

Conférences invitées

- [CC92] P. Cousot and R. Cousot. Comparing the Galois connection and widening/narrowing approaches to abstract interpretation. In M. Bruynooghe and M. Wirsing, editors, *Programming Language Implementation and Logic Programming, Proceedings of the Fourth International Symposium, PLILP'92*, Leuven, Belgique, 13–17 August 1992, Lecture Notes in Computer Science 631, pages 269–295. Springer-Verlag, Berlin, Germany, 1992. (à paraître dans Acta Informatica).
- [CC93] * P. Cousot and R. Cousot. Strictness analysis by abstract interpretation of a relational semantics. In *Actes de la conférence FMP*, Lecture Notes in Computer Science, Novossibirsk, Russie, June 1993. Springer-Verlag, Berlin, Germany.

Communications dans des conférences internationales avec comité de lecture

- [CC92] P. Cousot and R. Cousot. Inductive definitions, semantics and abstract interpretation. In *Conference Record of the 19th ACM Symposium on Principles of Programming Languages*, pages 83–94, Albuquerque, New Mexico, 1992. ACM Press, New York, U.S.A.
- [Cri92] R. Cridlig. An optimizing ML to C compiler. In *ACM SIGPLAN Workshop on ML and its Applications*, pages 28–36, San Francisco, California, 20–21 June 1992. ACM Press, New York, U.S.A.
- [GH93] É. Goubault and C. Hankin. A lattice for the abstract interpretation of term graph rewriting systems. In *Term graph rewriting*. John Wiley & sons, 1993.
- [GJ92] É. Goubault and T. P. Jensen. Homology of higher-dimensional automata. In *CONCUR'92*, New-York, August 1992. Springer-Verlag, Berlin, Germany.
- [Gou93] * É. Goubault. Domains of higher-dimensional automata. In *CONCUR'93*, Hildesheim, August 1993. Springer-Verlag, Berlin, Germany.

- [Gra91] P. Granger. Static analysis of linear congruence equalities among variables of a program. In S. Abramsky and T.S.E. Maibaum, editors, *TAPSOFT'91, Proceedings of the International Joint Conference on Theory and Practice of Software Development*, Brighton, U.K., Volume 1 (CAAP'91), Lecture Notes in Computer Science 493, pages 169–192. Springer-Verlag, Berlin, Germany, 1991. (une version révisée paraît dans *Mathematical Structures in Computer Science*).
- [Gra92] P. Granger. Improving the results of static analyses of programs by local decreasing iterations (extended abstract). In *Proceedings of the 12th Foundations of Software Technology and Theoretical Computer Science Conference*, pages 68–79, New Delhi, India, 18–20 December 1992, Lecture Notes in Computer Science 652, 1992. Springer-Verlag, Berlin, Germany. (la version complète, en cours de révision, sera publiée dans le *Journal of Logic and Computation*).
- [HL93] * K. Havelund and K. G. Larsen. The Fork Calculus. In *20th International Colloquium on Automata, Languages and Programming (ICALP)*, 1993. 15 pages.
- [Mon92] B. Monsuez. Polymorphic typing by abstract interpretation. In *Proceedings of the 12th Foundations of Software Technology and Theoretical Computer Science Conference*, New Delhi, India, 18–20 December 1992, Lecture Notes in Computer Science 652, 1992. Springer-Verlag, Berlin, Germany.
- [Mon93] * B. Monsuez. Polymorphic typing for call-by-name semantics. In *Actes de la conférence FMP*, Lecture Notes in Computer Science, Novossibirsk, Russie, June 1993. Springer-Verlag, Berlin, Germany.
- [vEGHN93] M. van Eekelen, É. Goubault, C. Hankin, and E. Nocker. Abstract reduction: Towards a theory via abstract interpretation. In *Term graph rewriting*. John Wiley & sons, 1993.

Autres conférences

- [HL93] * K. Havelund and K. G. Larsen. The Fork Calculus. In *Proceedings of the 4th Nordic Workshop on Program Correctness*, Bergen, Norway, 1993. (technical report by the Department of Informatics at the University of Bergen). version abrégée de [HL93], 10 pages.
- [Mon91] B. Monsuez. An attempt to find polymorphic types by abstract interpretation. *BIGRE, Actes JTASPEFL'91, Bordeaux*, IRISA, Rennes, France, 74:18–26, October 1991.
- [Mon92] B. Monsuez. Fractional types. *BIGRE, Analyse Statique WSA'92, Bordeaux*, IRISA, Rennes, France, 81–82:274–284, 23–25 September 1992.

Thèses

- [Gra91] ← P. Granger. *Analyses sémantiques de congruence*. Thèse de doctorat, École Polytechnique, July 1991.

Rapports non publiés

- [CG93] R. Cridlig and É. Goubault. Semantics and analysis of Linda-based languages. 1993.
- [Cri93] R. Cridlig. Projection analysis by abstract interpretation. 1993.