# Langages et Outils pour la Déduction sous Contraintes

IRIN - Nantes, LIFO - Orléans,
INRIA - Rocquencourt, LORIA - Nancy

2 mars 1998

On propose ici la constitution d'un groupe de travail intégré au GDR-ALP et dédié à un ensemble de thèmes regroupables sous l'appellation de *contraintes*. Ce groupe de travail pourrait s'intituler *LODEC* (cf. le développement plus haut) logiquement se rattachant au pôle "contraintes" du GDR-ALP. Les travaux visés pourront cependant intéresser des membres des pôles "spécification-test-preuve" de ce même GDR, et "algorithmique" du GDR-ARP.

## Problématiques et thèmes d'activité

On sait que les outils de déduction symbolique par les approches de la programmation logique et de la démonstration automatique ont beaucoup gagné en expressivité, et en efficacité, en adoptant le formalisme des contraintes. Des perspectives nouvelles ont pu ainsi voir le jour pour leur développement et leur optimisation : on peut citer, par exemple, l'emploi de la programmation logique inductive en apprentissage symbolique, l'apport des méthodes de calcul formel pour optimiser la résolution de contraintes numériques, ou encore l'emploi des algorithmes dits génétiques pour définir des stratégies de normalisation en présence d'un grand nombre de règles de réécriture non confluentes.

On peut aussi noter que la programmation non impérative (fonctionnelle ou par contraintes) se prête "plus naturellement" à des objectifs de déboguage et de validation, voire de génération de test, de logiciels. Chacun de ces développements envisageables comporte de nombreuses étapes de calculs symboliques, et la parallélisation d'au moins certaines d'entre elles peut contribuer à leur optimisation.

La déduction sous contraintes est ainsi un domaine encore en devenir, susceptible de regrouper, autour de ses outils et ses langages, des chercheurs d'horizons parfois variés mais dont les approches sont appelées à se compléter fructueusement. Voici donc une liste non-exhaustive de sujets, tous centrés sur la déduction (symbolique et/ou numérique) sous contraintes mais l'abordant sous des points de vue différents. Ils constitueront les principaux thèmes de travail du groupe LODEC.

i) L'optimisation des solveurs de contraintes par parallélisation ([12], Projet CCOPS sous [1]), et/ou par utilisation du calcul formel [10].

ii) Contraintes ensemblistes et Synthèse de programmes, dans la conception et validation de logiciels [21].

iii) L'optimisation de l'apprentissage symbolique [18], et de l'extraction de connaissances.

iv) Descriptions de stratégies de réécriture (éventuellement concurrentes), et génétiques dirigées par but [19], dans l'environnement ELAN [14]. L'un des intérêts ici est de définir les stratégies elles-mêmes par la réécriture [5].

v) La parallélisation des outils de démonstration automatique (unification, filtrage et réécriture) [8, 3, 2], mais aussi les systèmes déduction eux mêmes [13]. La conception et la validation automatique de circuits programmables [4] sera l'une des applications principales.

vi) Approches de diagnostic déclaratif, et de diagnostic par interprétation abstraite, pour programmes logiques contraints. Extension aux programmes concurrents avec contraintes [6].

vii) La programmation parallèle et concurrente à l'aide de langages logiques ou fonctionnels [7, 9, 15, 17] et les outils de mise au point.

Parmi les travaux visés, certains mettront l'accent sur l'aspect langages et structures en vue des outils de déboguage, validation, et d'apprentissage, notamment en thèmes ii), iii), vi), et en partie iv) et vii). D'autres viseront plus spécifiquement les outils d'optimisation, en particulier par une approche "parallèle" ; l'objectif alors sera l'identification et la programmation effective parallèle, de problèmes de calcul symbolique dans un cadre aussi général que possible ; la finalité étant de concevoir de nouvelles bibliothèques ou de nouvelles structures de langage. L'introduction de structures de haut niveau dans les langages et systèmes concurrents ou parallèles nécessite en effet des techniques issues de la recherche en programmation ; et inversement, la parallélisation du calcul symbolique pose des problèmes d'algorithmique théoriquement difficiles.

Quelle que soit la thématique visée, LODEC se proposera d'être un lieu d'échanges *à deux sens* entre les chercheurs, chacun pouvant à la fois y apporter sa contribution et/ou y trouver des solutions.

## Participants (principaux)

S.Anantharaman (LIFO,Orléans), F.Benhamou (IRIN,Nantes), J.Chassin (IMAG,Grenoble), E.Chailloux (LITP,Paris), P.Codognet (INRIA,Rocquencourt), G.Ferrand (LIFO,Orléans), J.-L.Giavitto (LRI,Orsay), G.Hains (LIFO,Orléans), C.Kirchner (LORIA,Nancy), F.Saubion (LERIA,Angers), L.Vigneron (LORIA,Nancy), C.Vrain (LIFO,Orléans).

## Organisation pratique

Des résultats sont attendus dans un delai de 10 à 16 mois, dans les 3 premiers groupes de thèmes énoncés plus haut ; il semble raisonnable d'afficher un delai de 24 à 30 mois pour les autres.

Les activités du groupe seraient réparties sur 24 mois et ponctuées de 3 rencontres. On encouragera la participation de partenaires étrangers particulièrement intéressés par les thèmes de LODEC. Certains d'entre eux comme M. Van EMDEN (Univ. Victoria, Canada), M. Cl. GEYER (Univ. de Porto Allegre, Brasil), M. COLE (Univ. d'Edimbourg, G.B.), et M. VALENÇA (Univ. de Braga, Portugal), pourront aussi agir comme relais pour l'intégration du groupe LODEC à des réseaux internationaux, comme par exemple les TMR européens. Et enfin les moyens de calcul du Centre Charles Hermite pourront être utilisés par les membres du groupe LODEC, pour leurs besoins en programmation et de mise au point.

## References


[1] AITEC, Contract Research Project in 1997.
Research Institute for Information Technology, Tokyo, Japan.
`www.icot.or.jp/AITEC/FGCS/funding/itaku-H9-index-E.html`.

[2] E. Alouini.
*Etude et mise en oeuvre de la réécriture conditionnelle concurrente sur des machines parallèles à mémoire distribuée.*
PhD thesis, Université Henri-Poincaré, Nancy I, May 1997.

[3] E. Alouini and C. Kirchner.
Toward the concurrent implementation of computational systems.
volume 1139, pages 1–31, Aachen (Germany), September 1996.

[4] N. Andrianarivelo, W. Bousdira, J. Chabin, and Z. Maazouzi.
A conditional rewrite-based method for designing combinational logic circuits.
Technical Report 98-1, Laboratoire d'Informatique Fondamentale d'Orléans, 1998.

[5] P. Borovanský, C. Kirchner, and H. Kirchner.
Controlling rewriting by rewriting.
In José Meseguer, editor, *Proceedings of the first international workshop on rewriting logic*, volume 4, Asilomar (California), September 1996.

[6] DiSCiPl, Projet Esprit reactive LTR no 22532, Oct. 1996.
Debugging System for Constraint Programming.
`http://discipl.inria.fr`

[7] C. Foisy and E. Chailloux.
Caml Flight: a portable SPMD extension of ML for distributed memory multiprocessors.
In A. W. Böhm and J. T. Feo, editors, *Workshop on High Performance Functionnal Computing*, Denver, Colorado, April 1995. Lawrence Livermore National Laboratory, USA.

[8] F. Fronhöfer and G. Wrightson, editors.
*Parallelization in Inference Systems*, number 590 in Lecture Notes in Artificial Intelligence, Dagstuhl Castle, Germany, December 1990. Springer.
Proceedings of International Workshop.

[9] J.-L. Giavitto and J.-P. Sansonnet.
$8\frac{1}{2}$: data-parallélisme et data-flow.
*RAIRO Technique et Science Informatiques*, 12(5), 1993.
Numéro spécial sur les langages à parallélisme de données.

[10] L. Granvilliers.
Solving Polynomial Systems using a Symbolic-Numerical Branch and Prune Approach.
*Journal of Universal Computer Science*, 1998.
(à paraître).

[11] G. Hains.
La complexité du filtrage associatif-commutatif.
*Comptes Rendus de l'Académie des Sciences, Paris*, t.311, Série I(11):741–744, Novembre 1990.

[12] G. Hains and M. H. van Emden.
Towards high-quality, high-speed numerical computation.
In *PACRIM'97, Pacific Rim Conference on Communications, Computers and Signal Processing*, University of Victoria, B.C., Canada, August 1997. IEEE.

[13] C. Kirchner, C. Lynch, and C. Scharff.
A fine-grained concurrent completion procedure.
In Harald Ganzinger, editor, *Proceedings of RTA'96*, volume 1103, pages 3–17, July 1996.

[14] H. Kirchner et P.-E. Moreau.
Protyping completion with constraints using computational systems.
Proc. of RTA'95, Springer-Verlag, LNCS $n^o$914, pp. 438-443.

[15] A. Lallouet and Y. Le Guyadec.
Une approche data-parallèle de la programmation logique.
In Dekeyser, Liebert, and Manneback, editors, *Actes des septièmes rencontres francophones du parallélisme, RenPar'7*, Mons, Belgique, Mai-Juin 1995. PIP-FPMs.

[16] S. Limet, F. Saubion.
On Partial Validation of Logic Programs.
Proc. of AMAST'97, Springer-Verlag, LNCS $n^o$1349, pp. 365-379.

[17] F. Loulergue and G. Hains.
Parallel functional programming with explicit processes: Beyond SPMD.
In C. Lengauer, M. Griebl, and S. Gorlatch, editors, *Euro-Par'97 Parallel Processing*, Passau, Germany, , LNCS $n^o$ 1300, pp. 530-537, Springer-Verlag, August 1997.

[18] L. Martin and C. Vrain.
Induction of constraint logic programs.
In Setsuo Arikawa and Arin K. Sharma, editors, *7th International Workshop on Algorithmic Learning Theory (ALT'96), LNAI 1160*, pp. 169-177, Springer Verlag, 1996.

[19] F.L. Neves, J.J. Almeida, L.S. Barbosa, and J.N. Oliveira.
CAMILA: Prototyping and Refinement of Constructive Specifications.
Proc. of AMAST'97, Springer-Verlag, LNCS $n^o$1349, pp. xxx

[20] M. Rusinowitch and L. Vigneron.
Automated Deduction with Associative-Commutative Operators.
Journal of AAEEC, Vol. 6, $n^o$1, pp. 23-56, January 1995.

[21] A. Tellez-Arenas.
Contraintes Ensemblistes : un environnement pour la validation de logiciel couplant l'approche B et la synthèse de programme (thèse en préparation).
Rapport de Recherche (Lifo, 98-xx).