

# Etude d'un solveur de contraintes à base de domaines abstraits

Université de Nantes, France

Charlotte Truchet  
Université de Nantes

Charlotte.Truchet@univ-nantes.fr

Marie Pelleau  
Université de Montréal

marie.pelleau@cirrelt.ca

## Contexte

La Programmation par Contraintes (CP) étudie des problèmes formalisés comme une conjonction de prédicats du premier ordre, les contraintes, qui définissent des relations entre les variables du problème [5]. La CP offre des méthodes de résolution génériques efficaces avec des applications variées en logistique, conception, bioinformatique, musique...

Ce stage porte sur un prototype de solveur de contraintes, Absolute [4], développé grâce à une collaboration entre l'équipe TASC du LINA à Nantes, et l'équipe Interprétation Abstraite à l'ENS. Absolute est basé sur la notion de domaine abstrait en interprétation abstraite (AI), une théorie de l'approximation de sémantiques pour la preuve de programmes [1]. Les outils d'AI intègrent indifféremment de nombreux types de domaines abstraits dans un même cadre, certains cartésiens (boîtes), d'autres relationnels (octogones, polyèdres...) [2]. On peut voir un domaine abstrait comme une représentation d'une famille spécifique de contraintes. Absolute utilise ce lien pour redéfinir les opérateurs de CP de façon générique et résoudre des problèmes de contraintes. Il s'agit d'un thème pluridisciplinaire nouveau, Absolute étant le premier solveur de contraintes sans contraintes de ce type.

## Sujet

Le solveur Absolute est encore un prototype. Le but de ce stage est de développer son moteur de résolution et d'en améliorer les heuristiques (choix de variables, choix des opérateurs de split, choix des propagateurs / domaines abstraits). Dans un premier temps, il s'agira de tester le solveur intensivement sur des benchmarks classiques. Il sera en particulier intéressant de déterminer, en fonction des contraintes du problème à résoudre, quel domaine abstrait est efficace ou non. Dans un second temps, en fonction des résultats de la première étape, il faudra proposer des heuristiques génériques adaptées des heuristiques classiques en contraintes [3], et enrichies avec l'utilisation des différents domaines abstraits.

## Informations pratiques

Ce stage est basé à l'Université de Nantes, au LINA (UMR 6241):

<http://www.lina.univ-nantes.fr/>.

Absolute est développé en OCaml, le candidat ou la candidate doit avoir des bases de programmation en OCaml.

## References

- [1] Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977. ACM Press, New York, NY.
- [2] Bertrand Jeannet and Antoine Miné. Apron: A library of numerical abstract domains for static analysis. In *Proceedings of the 21th International Conference Computer Aided Verification (CAV 2009)*, volume 5643 of *Lecture Notes in Computer Science*, pages 661–667. Springer, June 2009.
- [3] E. Monfroy, C. Castro, B. Crawford, R. Soto, F. Paredes, and C. Figueroa. A Reactive and Hybrid Constraint Solver. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(1):1–22, 2013.
- [4] Marie Pelleau, Antoine Miné, Charlotte Truchet, and Frédéric Benhamou. A Constraint Solver based on Abstract Domains. In Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 7737 of *Lecture Notes in Computer Science*, pages 434–454, Rome, Italie, 2013. Springer-Verlag.
- [5] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier, 2006.