

Semantic Array Dataflow Analysis

Paul Iannetta*

Univ Lyon, Université Claude Bernard Lyon 1
CNRS, ENS de Lyon, Inria,
LIP, F-69342, LYON Cedex 07, France
paul.iannetta@ens-lyon.fr

Lionel Morel[†]

Univ Grenoble Alpes, CEA, List
F-38000 Grenoble, France
lionel.morel@cea.fr

Laure Gonnord

Univ Lyon, Université Claude Bernard Lyon 1
CNRS, ENS de Lyon, Inria,
LIP, F-69342, LYON Cedex 07, France
laure.gonnord@ens-lyon.fr

Tomofumi Yuki

Inria, Univ Rennes, CNRS, IRISA
F-35000 Rennes, France
tomofumi.yuki@inria.fr

Abstract

Multi-core processors, and parallel processing in general, are now broadly used. Their horizon of applications ranges from mobile platforms to high-performance computing. However, allowing non-expert programmers to harness the parallelism in recent hardware requires significant advances in the entire compilation chain. It also means that the general forms of sequential programs, *e.g.*, with while loops and data-dependent control structures, should be amenable to parallelization.

The polyhedral model [2] is a powerful algebraic framework that is at the core of many advances in optimization and code generation of numerical kernels. One reason of its success is the practicality of the operations that are expressed in terms of algebraic computations on affine sets.

However, one of its limitations is that it only applies to programs with regular control and loops with static bounds. So as to cope with these intrinsic limitations, we propose to build a semantic foundation to the polyhedral model, hence allowing its extension with static analysis techniques which will open the door to the parallelization of irregular programs. For example,

- Inherently polyhedral programs written in a way that is not compliant with the syntactic restrictions of the polyhedral model;
- Polyhedral programs that have already been subject to a pre-compilation pass and which do not expose their polyhedral behavior as clearly as before;

*This work was partially funded by the French National Agency of Research in the CODAS Project (ANR-17-CE23-0004-01)

[†]Part of this work was carried out while the author was with INSA Lyon, Université de Lyon.

- Programs with while loops and more general data structures such as tree or maps.

As a first step, we propose to revisit the dataflow analysis [1] based on an operational semantics of programs. In this work in particular, we propose:

- A semantic definition of the notion of dependency *à la polyhedral model*, on a general imperative language;
- A rephrasing of the classical array dataflow analysis [1] in our setting, which enables us to recover classical results from the community;
- The definition of the notion of *covertly-regular* programs for which the polyhedral model algorithms and tools can be applied as-is;
- A notion of *approximated polyhedral model* for programs with non polyhedral control, on which we can compute an over-approximated set of dependencies.

The closest related work is Polly [3], a tool which strives to apply the ideas of the polyhedral model on program written in a low level intermediate representation has an objective similar to ours. The analysis in Polly can be viewed as a low-level version of our *covertly regular* loops detection without loop instrumentation. Our specificity is to fully characterize these semantically polyhedral loops and also to leave room for handling non-polyhedral programs through approximations.

References

- [1] Paul Feautrier. 1991. Dataflow analysis of array and scalar references. *International Journal of Parallel Programming* 20, 1 (1991), 23–53.
- [2] Paul Feautrier and Christian Lengauer. 2011. The Polyhedron Model. In *Encyclopedia of Parallel Programming*, David Padua (Ed.). Springer.
- [3] Tobias Grosser, Armin Groesslinger, and Christian Lengauer. 2012. Polly - performing polyhedral optimizations on a low-level intermediate representation. *Parallel Processing Letters* 22, 04 (2012), 1250010–1–1250010–28.