The APRON library for Numerical Abstract Domains

Bertrand Jeannet[1], Antoine Miné[2], and *al*[3]

[1] INRIA
[2] École Normale Supérieure
[3] CRI/École des Mines, École Polytechnique, Vérimag/CNRS

The APRON library is dedicated to the static analysis of the numerical variables of a program by Abstract Interpretation [1]. The aim of such an analysis is to infer invariants about the values of numerical variables, like "at control point k, variables x, y and z satisfy the property $1 \leq x + y \leq z$".

In this context, the APRON library provides a common interface to various libraries implementing numerical abstract domains.

Motivation and Principles. Many abstract domains have been designed and implemented for analysing the possible values of numerical variables during the execution of a program, *cf.* Figs. 1–2. Their implementations usually provides a set of core functionalities. However their API diverge largely (datatypes, signatures, . . .), which does not facilitate their diffusion and experimental comparison w.r.t. efficiency and precision aspects. The APRON library aims to provide:

- A uniform API for existing numerical abstract domains;
- A higher-level interface to the client tools, by factorizing functionalities that are largely independent of abstract domains.

From an abstract domain implementor point of view, the benefits of the APRON library are:

- The ability to focus on core, low-level functionalities;
- The help of generic services adding higher-level services for free.

For the client static analysis community, the benefits are an unified, higher-level interface, allows experimenting, comparing and combining abstract domains.

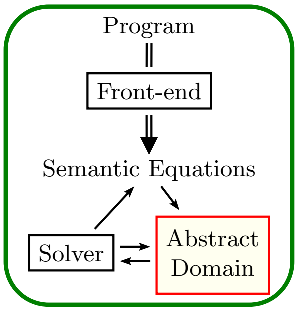

Fig. 1. Typical static analyser

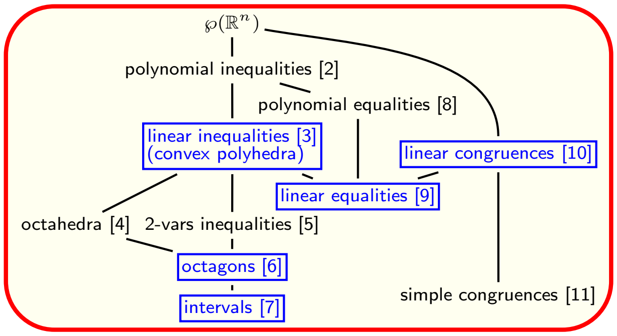

Fig. 2. Some abstract domains for numerical variables, partially ordered w.r.t. their expressiveness.

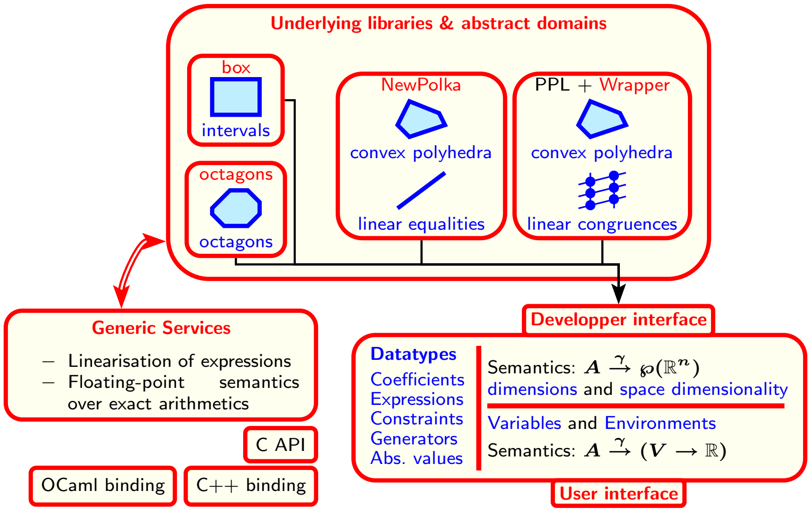
Underlying libraries & abstract domains
box
intervals
octagons
octagons
NewPolka
convex polyhedra
linear equalities
PPL + Wrapper
convex polyhedra
linear congruences
Generic Services
– Linearisation of expressions
– Floating-point semantics over exact arithmetics
C API
OCaml binding
C++ binding
Developper interface
Datatypes
Coefficients
Expressions
Constraints
Generators
Abs. values
Semantics: $A \xrightarrow{\gamma} \wp(\mathbb{R}^n)$
dimensions and space dimensionality
Variables and Environments
Semantics: $A \xrightarrow{\gamma} (V \to \mathbb{R})$
User interface

Fig. 3. Organisation of the APRON library

Implementation. Fig. 3 depicts the organisation of the APRON library. The existing underlying libraries connect to the developper interface, using domain-independent datatypes, and exploiting common services. Independent libraries like PPL [12] can be connected using a wrapper. Client tools connect to the higher-level user interface, where variables (or addresses) and environments replace geometrical notions like dimensions and space dimensionality.

The APRON library is written in C ANSI, with an object-oriented and thread-safe design. Both multi-precision and floating-point numbers are supported. A wrapper for the OCaml language is available, and a C++ wrapper is on the way. It is distributed under the LGPL license and available at `http://apron.cri.ensmp.fr/`. Future plan includes the connection of other abstract domains, as well as a generic combinator for the reduced product of abstract domains.

References

1. Cousot, P., Cousot, R.: Abstract interpretation and application to logic programs. Journal of Logic Programming **13**(2–3) (1992)
2. Bagnara, R., Rodríguez-Carbonell, E., Zaffanella, E.: Generation of basic semi-algebraic invariants using convex polyhedra. In: SAS'05. Volume 3148 of LNCS.
3. Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: POPL'78,
4. Clarisó, R., Cortadella, J.: The octahedron abstract domain. In: SAS'04. Volume 3148 of LNCS.
5. Simon, A., King, A., Howe, J.: Two variables per linear inequality as an abstract domain. In: LOPSTR'02. Volume 2664 of LNCS.
6. Miné, A.: The octagon abstract domain. In: AST'01 in Working Conference on Reverse Engineering 2001
7. Cousot, P., Cousot, R.: Static determination of dynamic properties of programs. In: 2nd Int. Symp. on Programming, Dunod, Paris (1976)
8. Müller-Olm, M., Seidl, H.: Program analysis through linear algebra. In: POPL'04.
9. Karr, M.: Affine relationships among variables of a program. Acta Informatica **6** (1976)
10. Granger, P.: Static analysis of linear congruence equalities among variables of a program. In: TAPSOFT'91. Volume 493 of LNCS.
11. Granger, P.: Static analysis of arithmetical congruences. Int. Journal on Computer Mathematics **30** (1989) 165–190
12. Bagnara, R., Ricci, E., Zaffanella, E., Hill, P.M.: Possibly not closed convex polyhedra and the Parma Polyhedra Library. In: SAS'02. Volume 2477 of LNCS.