Sound Semantic Static Analysis for Multiple Languages in the MOPSA Project

Antoine Miné*[1]

[1]Laboratoire dÍnformatique de Paris 6 (LIP6) – Sorbonne Universite, Centre National de la Recherche Scientifique : UMR7606 – 4 Place JUSSIEU 75252 PARIS CEDEX 05, France

Résumé

We present MOPSA, an ongoing project to design a static analyzer by abstract interpretation targeting multiple languages. Static analyzers aim at inferring, at compile time, properties of program executions, and help ensuring their correctness. They perform an interpretation of the program in an abstract domain of properties that is approximate (to ensure efficiency) but sound with respect to the semantic of the language (no false negative). The classic approach to multi-language analyzers is to use language-specific front-ends to translate programs into a common, lower-level language (e.g., a subset of C, an intermediate representation, or a bytecode) before the analysis. The back-end then iterates over a fixed language using a data abstraction which is composed of a collection of abstraction modules that can be plugged in and out to achieve various cost/precision trade-offs.
MOPSA strives to achieve a higher degree of modularity and extensibility by considering value abstractions, iterators, and control-flow abstractions uniformly as domain modules. Each module can extend the language syntax and rewrite expressions and statements to simpler ones dynamically to be processed by further modules. Hence, we avoid the pitfalls of fixing a common intermediate representation and we allow the rewriting to exploit any information found by the analysis so far. We will present MOPSA's architecture and some of our applications. These include a value analysis for C programs that also uses a specification language for C library functions. We also present type and value analyses for a significant subset of Python, as well as a value analysis for Python programs calling native C functions. Despite the variation in the languages analyzed and the properties inferred, these analyses share many common abstractions.

*Intervenant