Issues in Mechanizing Metatheory

Gyesik Lee

ROPAS

ROSAEC Workshop, Jan. 8, 2010

One Big Issue

> *How close are we to a world where every paper on programming languages is accompanied by an electronic appendix with machine-checked proofs?*
>
> *– POPLmarck Challenge*

POPLmark Challenge

- Proposed by the PL club at U. Pennsylvania, 2005.
- A set of benchmarks designed to evaluate the state of mechanization in the metatheory of programming languages.
- Focused on difficult issues to formalize such as binders with α-conversion.
- For better reasoning about the languages in which the software is written.
- Good for POPL papers.

POPLmark Challenge (Cont.)

> *To gauge progress in this area, we issue here a set of challenge problems, dubbed the POPLmark Challenge, chosen to exercise many aspects of programming languages that are known to be difficult to formalize.*
>
> *– POPLmark Challenge*

- Binding
- α-conversion
- Induction
- Substitution

But, something is missing.

Another Big Issue

> *As practitioners of machine-checked proof about real compilers, we have interests that are similar but not identical. We want to formally relate machine-checked proofs to actual implementations, not particularly to LaTeX documents.*
>
> *– CIVmark*

CIVmark

- "*A list-machine benchmark for mechanized metatheory*". A. Appel and X. Leroy, 2006.
- CIV = Compiler Implementation Verification
- Interests similar to those of POPLmark, but not identical.

CIVmark (Cont.)

- Emphasis on the importance of efficient definitions and implementations.
  - For representation of a type-checker algorithm in a mechanized metatheory (MM).
  - For formal, mechanical, and automatic derivation of an efficient implementation of the type-checker from the algorithm represented in the MM.

Issues: Machine Syntax

- Syntax of values, naturals, etc:
    - Inductive reasoning should be possible.
- Expansion of functions:
    - Representation of $f[v \mapsto a]$
    - Conditions for expansion

Issues: Operational Semantics

- Association of values to variables
  - Choice of functions or relations
- Operations on mathematical mappings:
- Inductive specification of mathematical relations such as *instructions*, *programs*

Issues: Type Systems

- Representation of environment for type assignments
- Specification of program typing: sequence of labeled environments
- Inductive specification of instruction typings, program typings, etc.

Summary: Mechanization Tasks

- Representation of an operational semantics
- Representation of a type system
- Correctness proof
- Representation of an efficient type-checking algorithm
- Termination of the type-checking algorithm
- Soundness of the type-checker