Specifying Programming Languages

Ryan Mulligan

April 19, 2011

Formal Definitions

A formal programming language definition allows you to reason mathematically—or formally—about your programming language. In other words, formal definitions let you obtain a proof that goes from your program to a result.

Definitions

Syntax What the program looks like. Usually defined with a context free grammar.

Semantics The meaning behind the syntax.

Specification The syntax and semantics of a certain domain.

Program An object valid under its intended syntax.

Uses

- Correctness
- Executable
- Deeper Understanding

Correctness

how a program behaves with respect to some specification

- partial
- total (Halt?)

Execution

If your specification is written in the right way you can get an executable implementation for free. (K does this using Maude)
Example: Maude and Piano Numbers

Maude's Grand Piano

Declarative and Executable

```
mod NAT is
  *** Syntax
  sort Nat .
  op 0 : -> Nat [ctor] .
  op s : Nat -> Nat [ctor] .
  op _+_ : Nat Nat -> Nat .
  *** Semantics
  vars N M : Nat .
  eq 0 + s(N) = s(N) .  eq s(N) + M = s (N + M) .
endm
```

Understanding

If you specify a system you will be forced to have a deep understanding of it. Easier

- Implementation
- Debugging

and you can sleep at night, maybe...

Structural Operational Semantics (SOS)

A Simple Language

My programs are statements that store the result of arithmetic expressions into variables and it returns a variable.

σ	store
a	arithmetic expression
i	integer
x	a variable

Challenge: Write the BNF for this language. (If time.)

Big Step SOS

A big step moves the program (P) from one state to another.
$\frac{\langle a1, \sigma \rangle \Downarrow \langle i1 \rangle, \langle a2, \sigma \rangle \Downarrow \langle i2 \rangle}{\langle a1 + a2, \sigma \rangle \Downarrow \langle i \rangle}$ where i is the sum of i1 and i2
Challenge: Write rule for multiplication, division.

Small Step SOS

A small step moves encapsulates a single computation.

$$\frac{\langle a1,\sigma\rangle \longrightarrow \langle a1',\sigma\rangle}{\langle a1+a2,\sigma\rangle \longrightarrow \langle a1'+a2,\sigma\rangle}$$

Challenge: What else do we need for Plus.

K

$$
\begin{aligned}
&AExp, Statement, Program \leq K \\
&Int \leq KResult \\
&Config ::= Int | Program | state(State) \\
&K ::= K | K \curvearrowright K
\end{aligned}
$$

Operator Definitions

$$\_ + \_[strict\ \ extends +_Int]$$
$$a1 + a2 \rightleftharpoons a1 \curvearrowright \square + a2$$

Variables

$$k(\frac{x}{\sigma[x]}\rangle state(\sigma)$$
$$k(\frac{x{:=}i}{.}\rangle state(\frac{\sigma}{\sigma[x \leftarrow i]})$$

Further

- Come to SIGPLan
- Maude?