PuF implementation

Intermediate presentation: Tools, work and problems

Merli Uus
Raivo Laanemets

December 8, 2008

Project tools

- We chose Java (both of us can use it)
- ANTLR LL(k) parser generator (k=2)
- Eclipse IDE with plugins (ANTLR plugin etc.)
- Google Code for version control
- Unit tests with JUnit (regression tests + Emma for measuring coverage)
- Apache Commons-(lang,io,cli) libraries
- ANT building tool

Current state of work

- Full grammar is implemented (but missing unary operators)
- Plain code generation almost done (but no data structures yet)
- Command-line interface (shell scripts to run the main class, debug options, etc.) - all done
- All Varmo examples work
- Optimizations:
  - Removal of unnecessary declarations (*basic1.puf*: 366 $\rightarrow$ 43 instructions)
  - Primitive code for the abstract interpretation (works more-less on *fact.puf*)
- 5200 lines of Java code without the generated parser (8500 with the parser)

Problems

- Testing is Nightmare:
  - VAM does not scale and is unstable: old Celeron vs. Q6600 - both take 2 hours to run *basic5.puf* at max speed. Q6600 crashed 4-5 times with Java exceptions at max speed before completion (syncronization issues?)
  - Tests cannot be run automatically (we have VAM source - but no time to modify it)
- ANTLR complains a lot (the base grammar was very ambiguous) although still works