COMP 520 - Compilers

Lecture 15 (April 4, 2017)

PA4 submission details

• PA4 due date: Mon Apr 17 (11:59 PM)
The PA4 checkpoint

• your pa4 directory should have
  – miniJava package
    • Compiler.java
    • SyntacticAnalyzer
    • AbstractSyntaxTrees
    • ContextualAnalyzer
    • CodeGenerator (new subpackage)

  – mJAM package (supplied on our web page)
    • Interpreter.java
    • Disassembler.java
    • Instruction.java
    • Machine.java
    • ObjectFile.java

• mJAM is needed only to check everything is working
  – pa4 testing will not copy your mJAM, it uses the mJAM as distributed

• pa4 readiness check will be available:  ../../../check/pa4.pl
Compiling and running miniJava programs (Unix)

• **Compiling test.java**
  
  - `java miniJava/Compiler test.java`
    
    • use mJAM.ObjectFile to write `test.mJAM` (note spelling!), be sure that it is written in the same directory as `test.java`
    
    • do not run the generated program as part of compilation!

• **Disassembling test.mJAM**
  
  - `java mJAM/Disassembler test.mJAM`
    
    • should write `test.asm` in same directory as `test.mJAM`

• **Executing test.mJAM**
  
  - `java mJAM/Interpreter test.mJAM`
    
    • `System.out.println` results from test.java will appear on stdout prefixed by “>>> “

• **Debugging test.mJAM**
  
  - `java mJAM/Interpreter test.mJAM test.asm`
    
    • Show machine data store and state, show code, set/remove breakpoints, single instruction execution
    
    • Type “?” for help
Check results

- To compare miniJava and java semantics of program `foo.java`

1. Run as miniJava program
   ```
   java miniJava/Compiler foo.java
   java mJAM/Interpreter foo.mJAM
   ```

2. Run as java program
   ```
   javac foo.java
   java foo.class
   ```

- Note that mJAM `println` prefixes output with ">>> "
• This package is also available on our web page
  – generates code for the Counter.java example (lec 16)
    • illustrates the Machine interface to generate mJAM instructions

  – .. then executes the generated code using mJAM
    • the Interpreter is started in debug mode so you can trace execution of the example code
    • to simplify the testing of your code generator you can install a similar shortcut to automatically execute generated code (e.g. in your compiler driver)
      – If you do so, be sure to restore standard functionality when submitting PA4