

Programming Language Qualifying Exam

Spring 2012

Answer the following five problems.

1. Languages and Compilation

- (a) Explain the difference between interpreted, compiled, and byte-code languages. Give an example of each.
- (b) Newer languages such as Scala and Clojure support and encourage the use of immutable data structures—once an object is created it cannot be modified. Explain how the proliferation of multi-core computing makes this style beneficial.
- (c) Suppose a programming team wanted to create a compiler for a programming language *Blub* to run on an X86 CPU. To do this, they created *two* compilers. One compiler compiled Blub to an intermediate code, and the second compiler compiled the intermediate code to X86 assembly. Give at least two reasons the programming team would want to do things this way, rather than compiling to X86 directly.
- (d) The call by value parameter passing style is the most commonly seen of the parameter passing styles. Define call by value, and give a reason for its popularity. Does it have any disadvantages?
- (e) Closures are sometimes said to be a poor man's object, and objects are sometimes said to be a poor man's closure. Define both objects and closures, and explain why they are said to be so similar.

2. Abstraction

- (a) What is an abstract data-type?
- (b) Suppose you are working as a developer at a widget factory. The widget machine has an on-board JVM. Programmers design the widgets using the WidgetWizard software, which then writes a .class file to be uploaded to the widget machine. One day you decompile the .class file and discover that the code output by the software is simply a jump-table hard coded into a huge integer array. There is no use of abstract data types, and no attempt to make the code readable. Is this acceptable? Why or why not?

3. Grammars

Consider the following grammar:

$$\begin{array}{l} S \rightarrow y x \\ \quad \rightarrow y E \\ E \rightarrow E z E \\ \quad | a S \end{array}$$

- (a) Construct the Characteristic Finite State Machine for the above grammar.
- (b) Convert the above grammar to an LL grammar (or explain why it is already LL).
- (c) What advantage results from a grammar being LL?
- (d) Is the above grammar ambiguous? Give a proof with your answer.

4. Weakest Precondition

- (a) Define *weakest precondition* and *weakest liberal precondition*.
- (b) In English, explain what $WP(S, T) = T$ indicates. (Note, we say explain, not simply translate.)
- (c) Consider the following program S . Let the postcondition $R \equiv x = y$. Determine formally the conditions under which this program returns the correct answer.

```
x := x * x;
y := y + y;
if x > y then x := x + 1
           else y := y + 1
fi
```

5. Loop Verification

- (a) In order to verify the correct operation of a loop, you need to check five formulas. What are they?
- (b) Fix the bug in the following program (if there is one), and formally prove the result. The postcondition is $\forall i. 0 \leq i < |A|. b[i] = \sum_{j=0}^i a[j]$ So, if $a = [1, 2, 3, 4]$ then on termination we should have $b = [1, 3, 6, 10]$.

You will need to determine the loop invariant.

```
i := 0 ;
s := 0 ;
do i < |A| ->
  b[i] = s ;
  s := s + a[i]
od
```