Programming Language Qualifying Exam
Fall 2008

Answer all five of the following problems.

1. **Languages and Compilation**

   (a) Explain the difference between an interpreter, a native-code compiler, and a byte-code compiler.

   (b) One of the major differences between Java and C++ is that Java does not allow pointer arithmetic. What benefit does this restriction have?

2. **Abstraction**

   (a) What is an abstract data-type? What advantages are there to using an abstract data type?

   (b) Suppose you need to implement a queue. You may either add the `enqueue` and `dequeue` methods to your `LinkList` class, or you may write a new `Queue` class that has a linked list private variable, and implement `enqueue` and `dequeue` that way. Is it necessary to provide a restricted class this way, or is it preferable to cut down on code bloat and simply use the `LinkList` class?

3. **Grammars**

   Consider the following grammar:

$$\begin{array}{lcl} S & \rightarrow & E\ x \\ E & \rightarrow & y\ S \\ & \mid & E\ x \\ & \mid & x \end{array}$$

   (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) Is the above grammar ambiguous? Give a proof with your answer.

4. **Weakest Precondition**

   (a) Give the definition of *weakest precondition*.

   (b) Suppose $WP(S, Q) = P$. Is it possible that there could be some P' such that $P' \Rightarrow P$? Is it possible that there could be some P'' such that $P \Rightarrow P''$?

   (c) Give the definition of WP for an `if` statement.

   (d) Construct a simple program S such that $WP(S, P) \cup WP(S, Q) \neq WP(S, P \cup Q)$, for some assertions P and Q.

5. **Loop Verification**

   (a) To verify a loop, you need to solve five equations. List each equation and give a one sentence description of its role in the verification.

   (b) Write a totally correct program S that uses a loop or loops so that $\{v = 2^x\}S\{v = 5^x\}$ where x is a logical integer variable. Give a full proof outline (including invariants and loop bounds). If you like, you can assume we have a function $maxp(p, q)$ that gives the largest power of p within q. (E.g., $maxp(2, 40) = 3$ because 2^3 divides 40 but 2^4 doesn't. You can also write exponentiation using infix ^ if you like (so 2^3 instead of 2^3).