Programming Language Qualifying Exam
Spring 2014

Part I: Languages, Compilation, Abstraction, and Grammars (50 points)

Note: For essay questions, concise answers are better than lengthy, confused answers.

1. [5 points] Data that is passed to a function is called *boxed* if a reference is passed instead of a copy of the data. Discuss some advantages and disadvantages of boxing.
2. [5 points] What is tail recursion, and what is its benefit?
3. [5 points] In practice, is the C language statically-typed, dynamically-typed, or a mix? Justify your answer.
4. [5 points] What is a closure, why are they appealing to programmers, and what challenges do they offer to compiler writers?
5. [5 points] One of the major changes from C++ to Java was the replacement of pointers with references. Discuss the advantages and liabilities of this change.
6. [5 points] If you read the output of a compiler, you will often see that the code makes no attempt to use abstraction. Why is this acceptable?
7. [20 points total] Consider the following grammar (a, b, c, and d are terminal symbols).

$$\begin{aligned} S &\rightarrow c\ S\ d \mid T \\ T &\rightarrow a \mid a\ T\ b\ T \end{aligned}$$

7a. [10 points] Construct the Characteristic Finite State Machine for the above grammar. Include the LR tables (i.e., Action and Go To).

7b. [5 points] Is this language LR(1)? LL(1)? What are the advantages/disadvantages of LR(1) and LL(1) languages?

7c. [5 points] Is this language ambiguous? (If so, give an ambiguous string.) Is there any advantage to having an ambiguous grammar?

Part II: Verification (50 points)

8. [5 points] Let $IF \equiv$ **if** $B_1 \rightarrow S_1$ □ $B_2 \rightarrow S_2$ **fi** be a nondeterministic **if** statement: If exactly one of the B's is true then we execute the corresponding S. If both of the B's are true, then we nondeterministically select exactly one of the S's and execute it. If neither of the B's is true, we produce a runtime error. Give $wp(IF, q)$ and $wlp(IF, q)$.
9. [15 points] Let $S \equiv$ `b[i] := b[k]; b[k] := b[j]` and let $q \equiv$ `b[i] < b[j]`. Calculate $wlp(S, q)$. Use logical/arithmetical manipulations to end up with a reasonably simple predicate.

10. [30 points] Write a binary search program S that uses a loop and satisfies

$$\{\mathtt{n} \geq 1 \land \mathtt{f(0)} < \mathtt{x} \leq \mathtt{f(n)}\}\ S\ \{0 \leq \mathtt{left} < \mathtt{n} \land \mathtt{f(left)} < \mathtt{x} \leq \mathtt{f(left+1)}\}$$

The function `f` is nondecreasing and `n` and `x` are constants. Feel free to use other variables in S (e.g., a variable `right`). Formally prove your result is totally correct by giving a valid annotation of the program (including loop bound and invariant). List any side conditions you need in addition to the annotated program. Simpler code and conditions are better than unnecessarily complex code or conditions.