Programming Language Qualifying Exam
Spring 2013

Answer the following five problems.

1. Language and Compilation

(a) (4 pts) Explain the difference between interpreted, compiled, and byte-code languages. Give an example of each.

(b) (4 pts) Give the pros and cons of using mutable vs immutable data structures for software that will run on a machine with a large number of cores.

(c) (4 pts) Give the pros and cons of static typing vs dynamic typing when designing a programming language.

(d) (4 pts) Give the pros and cons of call-by-value vs call-by-reference, from the point of the user of a language.

(e) (4 pts) Give the pros and cons of having vs not having closures in a programming language, both from the point of view of a user of the language and the point of view of an implementor of a translator or compiler for the language.

2. Abstraction

(a) (5 pts) What is an abstract data-type?

(b) (5 pts) Suppose we add cast operators to C so that you can cast a pointer to its underlying integer memory address and vice versa. For example, say `x` is stored at location `0x0000AB00` and `p` is an `int *` that contains the address `0x0000AB00`; then `(int) p` would evaluate to the integer `0x0000AB00` and `p = (int *) 0x0000AB00;` would leave `p` unchanged. Argue for or against the claim that this would radically change how one writes programs in C.

3. Grammars

Consider the following grammar:

$$
\begin{array}{rcl}
S & \rightarrow & S\ y \\
 & | & T \\
T & \rightarrow & a\ T\ b\ T \\
 & | & a
\end{array}
$$

(a) (5 pts) Construct the Characteristic Finite State Machine for the above grammar. You must show the LR tables (i.e., Action and Go To tables) for credit.

(b) (5 pts) Convert the above grammar to an LL grammar (or explain why it is already LL).

(c) (5 pts) What advantage results from a grammar being LL?

(d) (5 pts) Is the above grammar ambiguous? Give a proof with your answer.

4. Weakest Precondition

(a) (5 pts) Define *weakest precondition* and *weakest liberal precondition.*

(b) (5 pts) Let S be a program and let T and F stand for true and false respectively. In English, explain what $WLP(S, T) \wedge \neg\, WP(S, T) \Rightarrow F$ indicates. (Note: "explain," not simply "translate.")

(c) (10 pts) Consider the following program S. Let the postcondition $R \equiv x = y$. determine formally the conditions (if any) under which this program returns the correct answer. For full credit, logically simplify the precondition as much as possible.

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

5. Loop Verification

(a) (5 pts) In order to verify the correct operation of a loop, you need to check five formulas. What are they?

(b) (10 pts) The following program is incomplete. The precondition is that $A[0 \ldots n\text{–}1]$ represents an unsigned number in base b ($b \geq 2$, $n \geq 1$, and each $A[i]$ is ≥ 0 and $< b$). The postcondition is that v is the number that A represents. E.g., if $b = 2$ and $A = (1, 0, 0)$, then we should terminate with $v = 4$.

```
v := A[0];
K := 0;
while (K < n-1) do
…
od
```

You will need to determine the loop invariant and loop body. Formally prove your result by giving a valid annotation of your program.