Fall 2019 Ph.D. Qualifying Exam – Languages

Your Test ID Number: ________

Instructions

Write your test id number above and on each page of your answers. This exam is closed book and closed notes. You must pass both parts to pass the test.

Part 1: CS 536 [50 points]

(1) (8 points) For the nondeterministic program below, assume w1 and w2 are wlp(S1, q1) and wlp(S2, q2) respectively, and assume p1 → w1 and p2 → w2. Is this triple valid if p is p1 ∨ p2? Give a brief argument.

{ p1 ∨ p2 } if p1 → {w1 } S1 { q1 } □ p2 → {w2 } S2 { q2 } fi { q1 ∨ q2 }

(2) (8 points) Calculate the indicated weakest precondition below, following the definition of wp. Show your work and expand predicate[expr ⁄ var] substitutions as far as you can. You can ignore possible out-of-range indexes for b. If you wish, you may logically simplify your predicates, during and at the end of the calculation.

wp(b[e2] := e3, a < b[e1])

(3) (8 points) Calculate the indicated strongest postcondition below, following the definition of sp. Again, show your work; you may logically simplify your predicates. You can ignore possible division by zero.

sp(p(x, y, z) , x := x * y; x := x / z)

Questions 4 and 5 refer to the following program:

```
{ p1 ∧ p2 }
[     { p1 } await B1 then x := e1 end { q1 }
||    { p2 } await B2 then y := e2 end { q2 }
]
{ q1 ∧ q2 }
```

(4) (8 points) What are the interference-freedom tests for the program?

(5) (8 points) What are the deadlock-freedom tests for the program?

(6) (10 points) Fill in the proof outline below by giving definitions for p1, … p5 (but not p0). Use wp for p1 and sp for p3, but write out the results using pred [expr / var] substitution notation, not wp(…) or sp(…). Make the outline totally correct by including the bound wherever it's needed. You can assume all expressions evaluate without error.

```
{ p0 }              // p0 is given to us
{ p1 }              // use wp and pred [ expr / var ] notation
x := e1;  x := e2;
{ inv p } { bd e }
while B do
    { p2 }
    x := e3
    { p3 }          // use sp and pred [ expr / var ] notation & expand p2 into its parts
    { p4 }
od
{ p5 }
```

Part 2: CS440 [50%]

(1) General (20%, each question 5%): Use no more than two sentences to answer each of the following questions.

a. Give two advantages of having a programming language (such as Java or C++) that supports object-oriented programming.

b. Give one advantage of a statically typed language (e.g. Java) over a dynamically typed language (e.g. Python) and vice versa.

c. Give one advantage of pointers in C over references in Java and vice versa.

d. Java has *anonymous inner class* (some languages called something similar *closures)*. What is it? And why it is desirable?

(2) Compiler technology (20%)

a. (10%) Consider the following grammar (*a, b, c* and *d* are terminal symbols).

$$S \text{ -> } c\ S\ d \mid T$$

$$T \text{ -> } a \mid a\ T\ b\ T$$

Construct a finite state machine and its corresponding regular expression that will accept sentences generated by the grammer.

b. (10%) Show that for every RL grammar there exists a LL grammar.

(3) Programming language design (10%):

Java has *checked exceptions* but C# does not.

a. (3 points) What is a *checked exception*? Give an example code fragment that will NOT pass exception checking.

b. (4 points) Give a reason for and a reason against checked exceptions.

c. (3 points) Suppose that you are the language designer who must make the decision of whether to include checked exceptions, what will be your position and why?