CS558 Programming Languages – Fall 2023 – Study Questions Lecture 2b

These questions are intended for self-study, to help review and deepen your understanding of the lecture. Sample answers are available. There is nothing to hand in.

1. Use experimental semantics to determine how many times the Scala equivalent of the FORTRAN `DO` loop on slide 3 executes its body. How confident are you that your answer generalizes to other values of the loop start, stop, and step values? What other informal semantics could you consult to increase your confidence?

2. Consider the following Hoare triples. For each one, either give a proof tree showing that it is valid, or else argue why no such tree can exist.

(a) `{ X = 2 } Y := X { Y = 2 }`

(b) `{ X > 2 } X := X + 1 { X > 3 }`

(c) `{ X = 2 } X := 3 { X = 3 }`

(d) `{ X <= 0 } X := -X { X <= 0 }`

(e) `{ X = 0 }  X := X + 1; X := X - 1 { X = 0}`

(f) `{ False } X := 3 { X = 2 }`

(g) `{ X = 2 } if X > 0 then Y := 3 else Y := 4 endif { Y = 3 }`

(h) `{ X = 10 } while (X > 0) do X := X - 1 { X = 0 }`

3. Suppose that, inspired perhaps by the `COME FROM` statement in Lecture2a, we want to introduce a new compound statement `after`, the informal semantics of

s_1 `after` s_2

is to first execute s_2 and then execute s_1. Give a suitable Hoare logic rule of inference for the `after` statement. For example, you should be able to use your rule to prove the triple

`{ X = 0 } X := X * 2 after X := X + 1 { X = 2 }`