

CSE 6341, Assignment #3

Due: Nov. 7, 2018

No late assignments.

Note: The second mid-term will be on Friday, Nov. 9. Topics will be everything we discuss in class until then (since the first mid-term).

1. (10 points). This problem asks you to modify the behavior of Lisp COND construct. Currently, suppose you evaluate the following:

(COND (b1 e1) (b2 e2) ... (bn en))

And suppose all the b_i 's evaluate to NIL. Then the interpreter will raise an exception and escape to the top level. The modification you are required to make is the following: if all the b_i 's evaluate to NIL, the interpreter should return the value of e_1 as the value of the entire COND construct; if, on the other hand, any of the b_i 's, say, b_3 evaluates to T, then the behavior should be the normal one, i.e., return the value of e_3 . It is only in the case that all the b_i 's evaluate to NIL that it should return the value of e_1 as the value of the entire COND construct.

Your answer *must* be at the Lisp level, *not* in terms of your implementation in Python or Java etc. In other words, you must write Lisp functions *in the design notation* to, possibly, replace one or more of the Lisp functions that are currently part of the Lisp `evaluate[]` mechanism. You may, of course, introduce any additional auxiliary functions you need but, again, they must be Lisp functions written in the design notation. If it is not possible to do this, explain why not.

2. (10 points). Which of the following are operationally valid (with respect to the standard operational model) and which invalid? Provide brief justifications for your answers (for the invalid ones your justification must include an example that demonstrates the invalidity of the result):

(a) $\{ x = 1 \} y := x \{ x = 1 \}$

(b) $\{ z = 1 \wedge (z > 0 \Rightarrow x = z) \} y := x \{ x = y = z \}$

(c) $\{ x = 1 \} \text{if } (y = x) \text{ then } y := x \text{ else } y := x \{ y = 1 \}$

(d) $\{ \text{true} \} \text{while } x < 0 \text{ do } x := x - 1 \{ x = 0 \}$