Quiz 8b Rubric

1. (3 points) Recall that in Scheme, for a function call, we are not guaranteed whether the order of evaluation of the arguments is from left to right or from right to left (or some other order). For example, the expression (- (count) (count)) where count is a freshly created counter could evaluate to 1 or -1. We now look at the following expression:

(– (foo (mystery 0))

(foo (mystery 1)))

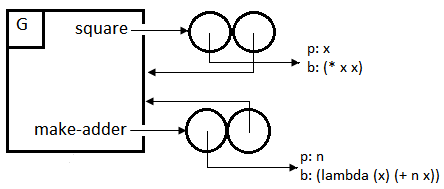
For each of the following cases, state whether we are guaranteed that the value of the expression does not depend on the order of evaluation of arguments.

Note: “Functional, ignores its argument” means that the procedure is functional, and that nowhere in the body of the procedure is the argument used – for example, (lambda (x) 42).

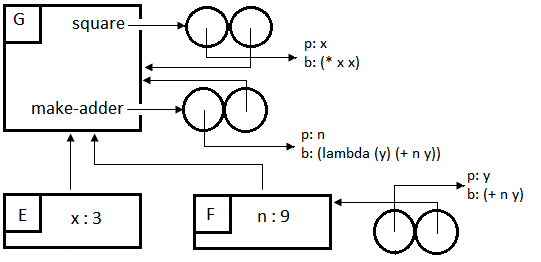
|  |  |  |
| --- | --- | --- |
| Programming style of foo | Programming style of mystery | Answer (Yes/No) |
| Functional | Functional | Yes |
| Non-functional | Non-functional | No |
| Functional | Non-functional | No |
| Functional, ignores its argument | Non-functional | Yes |
| Non-functional | Functional | No |
| Non-functional | Functional, ignores its argument | No |

0.5 points for each answer, all or nothing.

1. (1 + 1 + 1 + 1 points) Consider the initial environment diagram given below:



Suppose we then evaluated an expression, which evaluated without causing an error. This resulted in the following environment diagram:



* 1. How many times was a user-defined procedure called? How do you know?

Two times, because there were two new frames.

0.5 points for the number of times, 0.5 points for explanation.

* 1. Which user-defined procedures were called? How many times? How do you know?

square was called once, and make-adder was called once. Since a new frame must have bindings for the parameters, we know E1 corresponds to square and E2 to make-adder. Grading: 0.5 points for which procedures called and number of times it was called, and 0.5 points for explanation.

* 1. Suggest what expression we typed in to get this environment diagram. (There are infinitely many possibilities; you only need to give one.)

(make-adder (square 3))

0.5 points for an expression that creates the diagram, but with extra things, for example (define add-9 (make-adder (square 3))).

* 1. Is there any expression I can type into STk that will use the value of x defined in frame E? If yes, give such an expression. If no, explain why not. (Remember that when typing expressions into STk, we are in the global frame G.)

No, because now that we are in G, we cannot “re-enter” the frame, and since no procedure points to E and E has no children, there will never be a frame that extends E. Thus, we can no longer access the value of x.

0.5 points for “No”, 0.5 points for the explanation.

1. (3 points) Draw the environment diagram after all of the following expressions have been evaluated, and say what the final expression evaluates to:

> (define (make-nth n)

(lambda (lst)

(if (= n 0)

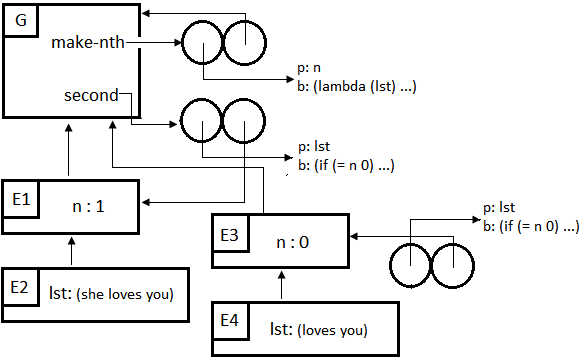
(car lst)

((make-nth (- n 1)) (cdr lst)))))

> (define second (make-nth 1))

> (second ‘(she loves you))

loves



0.5 points for the answer “loves” (not 1 point since you can guess what it would be based on the names of each procedure).

Remaining 2.5 points are for the diagram:

-0.5 points if missing the third procedure

-1 point if missing 1 or more frames

-0.5 points for each missing binding, up to a maximum of -1 point

-0.5 points for most other mistakes, but you should use your best judgment.

Other rubric – use if the student gets 0.5 points or lower out of 2.5 from previous rubric.

+0.5 points for creating a frame with a binding for n or lst

+0.5 points for creating at least 1 of the procedures in the correct answer.