Quiz 6: F♭ with States

EN.601.426/626 Principles of Programming Languages – SP26

Name: _____

Suppose our F♭ language is extended with mutable references. We have the following syntactic constructs (`Ref` creates a new mutable reference cell and returns a cell name, `!` dereferences the cell, `:=` mutates the cell with a new value, and c is the cell name):

$$\begin{array}{rcl} e & ::= & \cdots (\text{F}\flat \text{ syntax}) \cdots \mid \texttt{Ref}\ e \mid\ !\ e \mid e_1 := e_2 \\ v & ::= & \cdots (\text{F}\flat \text{ values}) \cdots \mid c \end{array}$$

Our operational semantics relation is now of the form

$$\langle e, \sigma \rangle \Rightarrow \langle v, \sigma' \rangle$$

where σ is of the form $\{c_1 \mapsto v_1, \ldots, c_n \mapsto v_n\}$, mapping cell names to values. Here, e is the program being evaluated, and v is the *evaluation result*, σ is the *initial state*, and σ' is called the *final state*.

Assume we have a rewrite rule for the sequence operator "`;`":

$$e_1\texttt{;}e_2 \equiv \texttt{(Fun}\ x\ \texttt{->}\ e_2\texttt{)}\ e_1, \text{where } x \notin FV(e_2)$$

1. What is the evaluation result of the following program?

```
Let x = Ref 0 In x := !x + 1; !x
```

2. Desugar the following program using the rewrite rule above:

```
Let temp = Ref !x In
x := !y;
y := temp
```

3. What is the **final state** after evaluating the following program?

```
Let x = Ref 0 In
Let y = Ref 1 In
Let temp = !x In
x := !y;
y := temp
```

Suppose we have the following operational semantics rule for the mutable assignment operator:

$$\frac{\langle e_1, \sigma_1 \rangle \Rightarrow \langle c, \sigma_2 \rangle \quad \langle e_2, \sigma_2 \rangle \Rightarrow \langle v, \sigma_3 \rangle}{\langle e_1 := e_2, \sigma_1 \rangle \Rightarrow \langle v, \sigma_3[c \mapsto v] \rangle}$$

4. What is the evaluation result of the above program in problem 3?

5. Can the following program be evaluated to a value? If so, what is the result?

```
Let x = Ref 0 In
x := x;
!!!!!!!!!x
```