CS51 **Spring 2024**

Code Review 9: Semantics

1 Practice Problems

Problem 1.1 (Free Variables)**. Instructions**: Write a box around the free variables in the following expressions. ~~For the~~ Circle bound variables, draw an arrow indicating its corresponding binding occurrence.

1.
```
let x = y + 2 in x + 3
in let f = (fun z -> y + 5)
in let z = 8 in f z
```
FV = {y}

2.
```
fun z -> a + b + c + z
```
FV = {a, b, c, z}

3.
```
x (fun x -> fun x -> x (fun x -> x))
```
FV = {x}

4.
```
let y = y + 2 in
let g = (fun x -> y + 5)
in (fun z -> z d) g
```
FV = {y, d}

5.
```
let x = 5 in
x + (fun a -> z (fun z -> a y)) 5
```
FV = {z, y}

Problem 1.2 (Substitution)**.** Write out the result of the following substitutions.

1.
```
((x + 1) + (y + 2)) [x -> 3]
```
= 3+1+y+2

2.
```
(let x = x + 1 in x + 2) [x -> 3]
```
= let x = 3+1 in x+2

3.
```
(let x = y + 2
in let z = 5
in z + x + j a)[x -> 5][j -> (fun x -> x * x)][a -> 4]
```

4.
```
(let y = y + 2 in
let g = (fun x -> y + 5)
```

See pg.

/ 3

```
in (fun z -> z d) g))[y -> 8][d -> 4]
```

5.
```
(let x = 5 in
x + (fun a -> z (fun z -> a y)) 5)[a -> 2][z -> fun x -> x]
```

Problem 1.3 (Substitution Semantics). Use the Substitution Semantic Evaluation Rules to derive the result of the following expression.

```
let x = 8 in
let y = x * 3 in
(fun x -> x + 4) 5 + x + y
```

Omit solution but see examples in book and lab

Problem 1.4 (Environment Semantics).

1. Define an expression that evaluates to different values under lexical and dynamic semantic systems
2. Write out two derivations for the result of the expression: one using lexical semantics and the other using dynamic semantics. → Omit soln
3. Derive the evaluation for the result of the following expression

```
let f = (fun x -> x * 2) in
let x = ref 42 in
(x := !x - 10; !x) + f !x ;;
```

Omit Solution

Problem 1.5 (Extending Semantics Models). Extend the substitution semantics and environment semantics (under mutable storage) models to support the additional language constructs:

1. Tuple Pairs. In addition, define semantic rules for the functions `fst` and `snd`.
2. Conditional Expressions. Define semantic rules for operators `||` and `&&`. In addition, define a semantic rule for the `if e1 then e2 else e3` construct
3. Lists. Define semantic rules for the `hd` and `tl` functions, and the `::` constructor.

Problem 1.2

3.

$$\left(\begin{array}{l}\text{let } x = y+2 \text{ in} \\ \text{let } z = 5 \text{ in} \\ z + x + 5\ a\end{array}\right)[x \mapsto 5][5 \to (\text{fun } x \to x * x)]$$

$$\Rightarrow \begin{array}{l}\text{let } x = y+2 \text{ in} \\ \text{let } z = 5 \text{ in} \\ z + x + (\text{fun } x \to x * x)\ a\end{array}[a \mapsto 4]$$

$$= \boxed{\begin{array}{l}\text{let } x = y+2 \text{ in} \\ \text{let } z = 5 \text{ in } z + x + (\text{fun } x \to x * x)\ 4\end{array}}$$

4

$$\begin{array}{l}(\text{let } y = y+2 \text{ in} \\ \text{let } g = \text{fun } x \to y+5 \text{ in}\end{array}$$

(fun z -> z d) g) [y ↦ 8]

⇒

let y = 8+2 in

let g = fun x -> y+5 in

(fun z -> z d) g) [d ↦ 4]

= let y = 8+2 in ✱

let g = fun x -> y+5 in

(fun z -> z 4) g

5. 1st substitution a ↦ z ⇒

let x = 5 in

x + ((fun a -> z (fun z -> 9 y)) 5

2nd sub. $z \mapsto \text{fun } x \to x \cdot x$

$\Rightarrow \; x + (\text{fun } a \to (\text{fun } x \to x \cdot x)(\text{fun } z \to ay)) 5$

$$\boxed{x + (\text{fun } a \to (\text{fun } x \to x \cdot x)(\text{fun } z \to ay)) 5}$$

Problem 1.4

```
let x = 5 in
let f = fun y -> x + 7 in
let x = 3 in
f x ;;
```

Problem 1.5

1)

$$E, S \vdash (P, Q) \Downarrow$$

$$E, S \vdash P \Downarrow v_P, S'$$

$$E, S' \vdash Q \Downarrow v_Q, S''$$

$$\Downarrow (v_P, v_Q), S''$$

$$E, S \vdash \text{fst } P \Downarrow$$

$$E, S \vdash P \Downarrow (v_1, v_2), S'$$

$$\Downarrow$$

$$v_1, S'$$

Similarly for snd

2)

$$E, S \vdash P \,\&\&\, Q \Downarrow$$

$$E, S \vdash P \Downarrow true, S'$$

$$E, S' \vdash Q \Downarrow true, S''$$

$$\Downarrow true, S''$$

$$E, S \vdash P \,\&\&\, Q \Downarrow$$

$$E, S \vdash P \Downarrow true, S'$$

$$E, S' \vdash Q \Downarrow false, S''$$

$$\Downarrow false, S''$$

Add 2 more cases for ll (P = false and Q = true) and (P = false and Q = false) in a similar fashion.

ll: Do the same thing writing out rules for cases accordingly.

$$\dfrac{E, S \vdash P \Downarrow true, S' \qquad E, S' \vdash Q \Downarrow v_Q, S''}{\text{if } P \text{ the } Q \text{ else } R \Downarrow v_Q, S''}$$

Add another case when P = false and evaluate R instead.

3.

note: :: is right associative

metavar to represent [] list

$$\frac{E, S \vdash q \Downarrow [v_2, \ldots, v_n], S' \qquad E, S' \vdash p \Downarrow v_1, S''}{E, S \vdash p :: q \Downarrow [v_1, v_2, \ldots, v_n], S''}$$

$$\frac{E, S \vdash L \Downarrow [v_1, \ldots, v_n], S'}{E, S \vdash \mathrm{hd}\ L \Downarrow v_1, S'}$$

Define rule similarly for tail

replacing v_1 in last line with $[v_2, \ldots, v_n]$