Name:

CS 421 — Small Step Semantics Activity

Mattox Beckman

The Rules

Small Step

$$< \texttt{skip}, \sigma > \rightarrow < E, \sigma >$$
$$< u := t, \sigma > \rightarrow < E, \sigma[u := \sigma(t)] >$$

$$\frac{< S_1, \sigma > \rightarrow < S_2, \tau >}{< S_1; S, \sigma > \rightarrow < S_2; S, \tau >}$$
$$E; S \equiv S$$

$$< \texttt{if } B \texttt{ then } S_1 \texttt{ else } S_2 \texttt{ fi}, \sigma > \rightarrow < S_1, \sigma > \text{ where } \sigma \models B$$
$$< \texttt{if } B \texttt{ then } S_1 \texttt{ else } S_2 \texttt{ fi}, \sigma > \rightarrow < S_2, \sigma > \text{ where } \sigma \models \neg B$$

$$< \texttt{while } B \texttt{ do } S_1 \texttt{ od}, \sigma > \rightarrow < S_1; \texttt{while } B \texttt{ do } S_1 \texttt{ od}, \sigma > \text{ where } \sigma \models B$$
$$< \texttt{while } B \texttt{ do } S_1 \texttt{ od}, \sigma > \rightarrow < E, \sigma > \text{ where } \sigma \models \neg B$$

Big Step

$$\frac{}{< i, \sigma > \Downarrow_e i} \quad \text{if } i \text{ is an integer.}$$

$$\frac{}{< u, \sigma > \Downarrow_e v} \quad \text{if } u := v \in \sigma.$$

$$\frac{< e_1, \sigma > \Downarrow_e v_1 \quad < e_2, \sigma > \Downarrow_e v_2}{< e_1 \oplus e_2, \sigma > \Downarrow_e v_1 \oplus v_2}$$

$$\frac{}{< \texttt{true}, \sigma > \Downarrow_b \texttt{true}} \qquad \frac{}{< \texttt{false}, \sigma > \Downarrow_b \texttt{false}}$$

$$\frac{}{< u, \sigma > \Downarrow_b v} \quad \text{if } u := v \in \sigma.$$

$$\frac{}{< \texttt{skip}, \sigma > \Downarrow \sigma}$$

$$\frac{< e, \sigma > \Downarrow_e v}{< x := e, \sigma > \Downarrow \sigma[x := v]}$$

$$\frac{< S_1, \sigma > \Downarrow \sigma' \qquad < S_2, \sigma' > \Downarrow \sigma''}{< S_1; S_2, \sigma > \Downarrow \sigma''}$$

$$\frac{< e_1, \sigma > \Downarrow_e v_1 \quad < e_2, \sigma > \Downarrow_e v_2}{< e_1 \sim e_2, \sigma > \Downarrow_b v_1 \sim v_2}$$

$$\frac{< B, \sigma > \Downarrow_b \texttt{true} \quad < S_1, \sigma > \Downarrow \sigma'}{< \texttt{if } B \texttt{ then } S_1 \texttt{ else } S_2 \texttt{ fi}, \sigma > \Downarrow \sigma'} \qquad \frac{< B, \sigma > \Downarrow_b \texttt{false} \quad < S_2, \sigma > \Downarrow \sigma'}{< \texttt{if } B \texttt{ then } S_1 \texttt{ else } S_2 \texttt{ fi}, \sigma > \Downarrow \sigma'}$$

$$\frac{< B, \sigma > \Downarrow_b \texttt{false}}{< \texttt{while } B \texttt{ do } S \texttt{ od}, \sigma > \Downarrow \sigma} \qquad \frac{< B, \sigma > \Downarrow_b \texttt{true} \quad < \texttt{while } B \texttt{ do } S \texttt{ od}, \sigma' > \Downarrow \sigma'' \quad < S, \sigma > \Downarrow \sigma'}{< \texttt{while } B \texttt{ do } S \texttt{ od}, \sigma > \Downarrow \sigma''}$$

Reductions

Reduce the following programs according to the semantic rules given.

Problem 1)

```
<if x>y then m:=x-y else m:=y-x fi;, {x:=10,y:=30}>
```

Problem 2)

```
<while x > 1 do x:=x/2 od, {x:=2}>
```

Make your own rules!

Problem 3)

Write a rule to explain the `when B S` statement. It executes S only if B is true.

Problem 4)

Write a rule for `do S while B od`. It is like while, but executes S at least one time.

Lambda Calculus!

Problem 5)

Write the big-step rules for λ-calculus! Use call-by-value.