Homework - 3

Venkata Akshith Reddy Kasireddy(A20455209)
Sai Vishal Kodimela (A20453006)
Atharva Kadam (A20467229)

1. Given program can be translated to one of the following programming language:

```
p := 1;
k := 0;
x := 0;
while (k++ < n)
do
        x = x+1;
        p = p*x;
od;
```

2. Here we have S ≡ if x > 0 then x := x*z else if y > 0 then y := y*z fi fi.

2a.

Let us consider σ = {x = 2, y = 6, z = 4}.

⟨ S, {x = 2, y = 6, z = 4} ⟩

= ⟨ if x > 0 then x := x*z else if y > 0 then y := y*z fi fi, {x = 2, y = 6, z = 4} ⟩

// Here we are using = instead of → because we are just expanding S which isn't a semantic operation.

→ ⟨ x := x*z, {x = 2, y = 6, z = 4} ⟩ // because σ ⊨ x > 0

→ ⟨ E, {x = 8, y = 6, z = 4} ⟩ // evaluate the assignment expression

2b.

Let us consider $\sigma = \{x = -2, y = 8, z = 5\}$

$\langle$ S, $\{x = -2, y = 8, z = 5\}$ $\rangle$

= $\langle$ if x > 0 then x := x*z else if y > 0 then y := y*z fi fi, $\{x = -2, y = 8, z = 5\}$$\rangle$

// Here we are using = instead of → because we are just expanding S which isn't a semantic operation.

→ $\langle$ y := y*z, $\{x = -2, y = 8, z = 5\}$ $\rangle$ // because $\sigma \not\models x > 0$ and $\sigma \models y > 0$

→ $\langle$ E, $\{x = -2, y = 40, z = 5\}$ $\rangle$ // evaluate the assignment expression

2c.

Let us consider $\sigma = \{x = -1, y = -2, z = 6\}$

$\langle$ S, $\{x = -1, y = -2, z = 6\}$ $\rangle$

= $\langle$ if x > 0 then x := x*z else if y > 0 then y := y*z fi fi, $\{x = -1, y = -2, z = 6\}$ $\rangle$

// Here we are using = instead of → because we are just expanding S which isn't a semantic operation.

→ $\langle$ skip, $\{x = -1, y = -2, z = 6\}$ $\rangle$ // because $\sigma \not\models x > 0$ and $\sigma \not\models y > 0$

→ $\langle$ E, $\{x = -1, y = -2, z = 6\}$ $\rangle$

3. Given W ≡ while k ≠ n do S od where S ≡ k := k+1; x := x + k*k and $\sigma_0 = \{k = 0, x = 1, n = 4\}$. Evaluation of $\langle W, \sigma_0 \rangle$ is as follows:
$\langle W, \sigma_0 \rangle$ = $\langle$ while k ≠ n do S od, σ_0 $\rangle$

$\rightarrow \langle S; W, \sigma_0 \rangle$ //As the values in σ_0 satisfies $k \neq n$, the loop of W begins.

$= \langle k := k+1;\ x := x + k*k\ ;\ W, \sigma_0 \rangle$

// Here we are using = instead of → because we are just expanding S which isn't a semantic operation

$\rightarrow \langle x := x + k*k\ ;\ W, \sigma_0[k \mapsto 1] \rangle$

$\rightarrow \langle W, \sigma_1 \rangle$ // Whereas $\sigma_1 = \sigma_0\ [k \mapsto 1]\ [x \mapsto 2]$

$\rightarrow \langle S; W, \sigma_1 \rangle$ // As $\sigma_1 \models k \neq n$ $(\sigma_1 \models 1 \neq 4)$

$\rightarrow \langle k := k+1;\ x := x + k*k\ ;\ W, \sigma_1 \rangle$

$\rightarrow \langle W, \sigma_2 \rangle$ // Whereas $\sigma_2 = \sigma_1\ [k \mapsto 2]\ [x \mapsto 6]$

$\rightarrow \langle k := k+1;\ x := x + k*k\ ;\ W, \sigma_2 \rangle$ // because $\sigma_2 \models k \neq n$ $(\sigma_2 \models 2 \neq 4)$

$\rightarrow \langle W, \sigma_3 \rangle$ // Whereas $\sigma_3 = \sigma_2\ [k \mapsto 3]\ [x \mapsto 15]$

$\rightarrow \langle k := k+1;\ x := x + k*k\ ;\ W, \sigma_3 \rangle$ // because $\sigma_3 \models k \neq n$ $(\sigma_3 \models 3 \neq 4)$

$\rightarrow \langle W, \sigma_4 \rangle$ // Whereas $\sigma_4 = \sigma_3\ [k \mapsto 4]\ [x \mapsto 31]$

As the values in σ_4 does not satisfy $k \neq n$, the expression breaks the loop and returns to the final state.

$\rightarrow \langle E, \sigma_4 \rangle$ //because $\sigma_4 \not\models k \neq n$ $(\sigma_4 \not\models 4 \neq 4)$

4. Given $S \equiv$ if $x > 0$ then $x := x*z$ else if $y > 0$ then $y := y*z$ fi fi.

2a. Here we have $\langle S, \{x = 2, y = 6, z = 4\} \rangle$. Let us consider state $\sigma = \{x = 2, y = 6, z = 4\}$. Then, following is the denotational semantics:

$M(S, \sigma) = M(S, \{x = 2, y = 6, z = 4\})$

$= M(\text{if } x > 0 \text{ then } x := x*z \text{ else if } y > 0 \text{ then } y := y*z \text{ fi fi}, \{x = 2, y = 6, z = 4\})$

$= M(x := x*z, \{x = 2, y = 6, z = 4\})$ // because $\sigma \models x > 0$

= {x = 8, y = 6, z = 4}

2b. Here we have ⟨ S, {x = -2, y = 8, z = 5} ⟩. Let us consider state σ = {x = 2, y = 6, z = 5}.

Then, following is the denotational semantics:

M(S, σ) = M(S, {x = -2, y = 8, z = 5})

= M(if x > 0 then x := x*z else if y > 0 then y := y*z fi fi, {x = -2, y = 8, z = 5})

= M(y := y*z, {x = -2, y = 8, z = 5}) // because σ ⊭ x > 0 and σ ⊨ y > 0

= { x = -2, y = 40, z = 5 }

2c. Here we have ⟨ S, {x = -1, y = -2, z = 6} ⟩. Let us consider state σ = {x = -1, y = -2, z = 6}.

Then, following is the denotational semantics:

M(S, σ) = M(S, {x = -1, y = -2, z = 6})

= M(if x > 0 then x := x*z else if y > 0 then y := y*z fi fi, {x = -2, y = 8, z = 6})

= M(skip, {x = -1, y = -2, z = 6}) // because σ ⊭ x > 0 and σ ⊭ y > 0

= {x = -1, y = -2, z = 6}

5. Here we have W ≡ while k ≠ n do S od where S ≡ k := k+1; x := x+k*k. If we consider σ(k) < σ(n) (if value of k is less than value of n), then there arises a diverging loop(loop diverges) but if we consider σ(k) ≥ σ(n), we can terminate the loop, so the set for σ will be as follows:

{σ ∈ Σ | σ(k) ≥ σ(n)}

σ = { k = 1, x = 1, n = 0} and σ = { k = 2, x = 2, n = 1} are some of the counter examples for set σ where ⟨ W, σ ⟩ →* ⟨ E, ⊥ ⟩.

6.Case 1: square root of negative number

Here, we will consider the σ(k)=-1. By doing that, now our program S with the states σ = {m = α, k = -1, b = β} will be as follows:

S ≡ x := b[m-2] / sqrt(k) for ⟨ *m*:= α ; *k*:= *-1*, b = β, σ ⟩

$M(S,\ \sigma) = \{\sigma[b[\alpha -2] / sqrt(-1)]\}$

// which will throw square root of negative number error

Thus, for $\sigma = \{m = \alpha, k = -1, b = \beta\}$

$\sigma(e) = \perp e$

$M(S, \sigma) = \{\perp e\}$

Case 2: Divide by zero

Here, we will consider the $\sigma(k)=0$. By doing that, now our program S with the states $\sigma = \{m = \alpha, k = 0, b = \beta\}$ will be as follows:

$S \equiv x := b[m-2] / sqrt(k)$ for $\langle\ m := \ \alpha\ \ ;\ k := \ 0, b = \beta,\ \sigma\ \rangle$

$M(S,\ \sigma) = \{\sigma[b[\alpha -2] / sqrt(0)]\}$

// which will throw division by zero error

Thus, for $\sigma = \{m = \alpha, k = 0, b = \beta\}$

$\sigma(e) = \perp e$

$M(S, \sigma) = \{\perp e\}$

Case 3: Array index out of bound

The array b has length δ, so if we try to access an element at the position greater than or equal to δ, program S will throw an error. Consider now our program S with the states $\sigma = \{m = \delta+3, k = \gamma, b = \beta\}$ will be as follows:

$S \equiv x := b[m-2] / sqrt(k)$ for $\langle\ m := \delta+3\ ;\ k := 0, b = \beta,\ \sigma\ \rangle$

$M(S, \sigma) = \{\sigma[b[\delta +3-2] / sqrt(\gamma)]\} \rightarrow \{\sigma[b[\delta +1] / sqrt(\gamma)]\}$

// which will throw array index out of bound error

Thus, for $\sigma = \{m = \alpha, k = 0, b = \beta\}$

$\sigma(e) = \perp e$

$M(S, \sigma) = \{\perp e\}$