HW3: Induction

Golam Md Muktadir

May 4, 2019

$$\frac{\langle x, \sigma\rangle \Downarrow \sigma(x) \qquad \dfrac{\langle y, \sigma\rangle \Downarrow \sigma(y) \qquad \langle z, \sigma\rangle \Downarrow \sigma(z)}{\langle y+z, \sigma\rangle \Downarrow \sigma(y)+\sigma(z)}}{\langle x*(y+z), \sigma\rangle \Downarrow \sigma(x)*(\sigma(y)+\sigma(z))}$$

1 Problem 1

1.1 Question

In the WHILE language, prove or disprove the equivalence of the two commands:

$$t := x; x := y; y := t$$

and

$$t := y; y := x; x := t$$

(where x; y, and t are distinct locations).

1.2 Answer

We can disprove it with a counter example with a state s in which two sequences yield different results.

Let

$$s = [t \to 0, x \to 5, y \to 100]$$

Then,

$$\langle t := x; x := y; y := t, s\rangle \Downarrow s[t \to 5, x \to 100, y \to 5]$$

$$\langle t := y; y := x; x := t, s\rangle \Downarrow s[t \to 100, x \to 100, y \to 5]$$

So, location, t, has different values after execution of the sequences under this s.

2 Problem 2

2.1 Question

In the WHILE language, prove that if

$$\langle \text{while b do } y := y - x, s\rangle \Downarrow s'$$

then there exists an integer k such that

$$s(y) = s'(y) + k * s(x)$$

Please make it explicit if/when you reason by induction on derivations, stating your induction hypoth- esis.

2.2 Answer

We can prove it by induction on the derivation of while command.
Let us define,

W $\doteq$ while b do $y := y - x$
$\sigma_i \doteq s[y := y - (k-i) * x]$
P(n) $\doteq \langle W, \sigma_n \rangle \Downarrow \sigma_o$

So,

$\sigma_o = s[y := y - k * x] = s'$
$\sigma_k = s[y := y - x]$

Base case: $i = 0$

$$\text{D: } \frac{\langle b, \sigma_o \rangle \Downarrow false \qquad \langle skip, \sigma_o \rangle \Downarrow \sigma_o}{\langle W, \sigma_o \rangle \Downarrow \sigma_o} \text{ (by } if^{ff}_{sos}, while_{sos} \text{)}$$

So, P(0) holds.

Induction case: $\forall i <= n, P(i)$ **holds,** $n \in \mathbb{N}$ We need to show that P(n+1) holds, too.

$$\text{D: } \frac{D1 : \langle b, \sigma_o \rangle \Downarrow true \qquad D2 : \langle y := y - x, \sigma_{n+1} \rangle \Downarrow \sigma_n \qquad D' : \langle W, \sigma_n \rangle \Downarrow \sigma_o}{\langle W, \sigma_{n+1} \rangle \Downarrow \sigma_o} \text{ (by } while_{sos} \text{)}$$

Derivation of D2:

$$\text{D2: } \frac{\langle y := y - x, s[y := y - k * x + (n+1) * x] \rangle \rightarrow s[y := y - k * x + (n) * x]}{\langle y := y - x, \sigma_{n+1} \rangle \Downarrow \sigma_n} \text{ (by } ass_{sos} \text{)}$$

Now D' is true by our induction hypothesis. So, P(n+1) holds.

So, we proved σ_o to be our terminal state. We can see from the definition of it that,

$\sigma_o(y) = s(y) - k * x$
$\Rightarrow s'(y) = s(y) - k * x$
$\Rightarrow s(y) = s'(y) + k * x$ (Proved)

3 Problem 3

3.1 Question

In the WHILE language, prove:

$$\forall c1, c2, c3 : c1; (c2; c3) \approx (c1; c2); c3$$

3.2 Answer

$\langle\rangle \Downarrow$

$$\text{LS: } \dfrac{\langle c1, s_o\rangle \Downarrow s' \qquad \dfrac{\langle skip, s'\rangle \Downarrow s' \qquad \dfrac{\langle c1, s'\rangle \Downarrow s'' \qquad \langle c2, s''\rangle \Downarrow s}{\langle c2, c3, s'\rangle \Downarrow s}}{LS' : \langle (c2, c3), s'\rangle \Downarrow s}\text{ (seq)}}{\langle c1; (c2; c3), s_o\rangle \Downarrow s}\text{ (seq)}$$

So, replacing LS' with commands changing states only in its derivation sequence, we get

$$\text{LS: } \dfrac{L1 : \langle c1, s_o\rangle \Downarrow s' \qquad L2 : \langle c1, s'\rangle \Downarrow s'' \qquad L3 : \langle c2, s''\rangle \Downarrow s}{\langle c1; (c2; c3), s_o\rangle \Downarrow s}\text{ (seq)}$$

$$\text{RS: } \dfrac{\dfrac{\langle skip, s_o\rangle \Downarrow s_o \qquad \dfrac{\langle c1, s_o\rangle \Downarrow t' \qquad \langle c2, t'\rangle \Downarrow t''}{\langle c2, c3, s_o\rangle \Downarrow t'}}{RS' : \langle (c1, c2), s_o\rangle \Downarrow t''}\text{ (seq)} \qquad \langle c3, t''\rangle \Downarrow t}{\langle (c1; c2); c3, s_o\rangle \Downarrow t}$$

So, replacing RS' with commands changing states only in its derivation sequence, we get

$$\text{RS: } \dfrac{R1 : \langle c1, s_o\rangle \Downarrow t' \qquad R2 : \langle c2, t'\rangle \Downarrow t'' \qquad R3 : \langle c3, t''\rangle \Downarrow t}{\langle (c1; c2); c3, s_o\rangle \Downarrow t}$$

Now, because of determinism of L1 and R1, we have,
$s' = t'$
$\Rightarrow R2 = \langle c2, s'\rangle \Downarrow t''$
$\Rightarrow s'' = t''$ (determinisim of L2 and R2)
$\Rightarrow R3 = \langle c3, s''\rangle \Downarrow t$
$\Rightarrow s = t$ (determinisim of L3 and R3)
Thus, $\langle c1; (c2; c3), s_o\rangle \Downarrow s \approx \langle (c1; c2); c3, s_o\rangle \Downarrow t$
$\Rightarrow c1; (c2; c3) \approx (c1; c2); c3$ (Proved)