Lenguajes y compiladores
Práctico 5

Repaso.

(1) Escriba un teorema de coincidencia para la semántica operacional.
(2) Después de terminar con los ejercicios, pruebe el teorema de coincidencia (sin hacer ninguna inducción).

(1) Dado el programa $P \equiv$

newvar x := y + x **in**
  while x > 0 **do** **if** x > 0 **then skip else fail**

Caracterice (sin calcular la semántica) los estados σ en los cuales P se comporta como **skip**.

$\{\sigma \in \Sigma : \sigma y + \sigma x \leq 0\}$ (si entra al while no termina)

(2) Demostrar o refutar las siguientes equivalencias usando semántica denotacional:

(a) c; **while true do skip** $\equiv$ **while true do skip**.
(b) c; **fail** $\equiv$ **fail**.
(c) **newvar** v := e **in** $v := v + 1$; **fail** $\equiv$
  newvar w := e **in** $w := w + 1$; **fail**.
(d) **while** b **do fail** $\equiv$ **if** b **then fail else skip**.
(e) x := 0; **catch** x := 1 **in while** x < 1 **do fail** $\equiv$
  x := 0; **while** x < 1 **do catch** x := 1 **in fail**

Considere el punto (a) para el lenguaje sin fallas: ¿la respuesta es la misma que para el lenguaje con fallas?

(a) $[\![c ;$ while true do skip $]\!]\sigma$
$= [\![$while true do skip$]\!]_* ([\![c]\!]\sigma)$
si $[\![c]\!]\sigma = [\![$fail$]\!]\sigma = (abort, \sigma)$

y entonces

$[\![\text{while true do skip}]\!]_* \langle abort, \sigma\rangle$

$= \langle abort, \sigma\rangle$

$\neq \bot$

$= [\![\text{while true do skip}]\!]\sigma$

(b) $c; fail \equiv fail$

falso con $c = \text{while true do skip}$

$[\![c; fail]\!]\sigma = [\![fail]\!]_*([\![c]\!]\sigma)$

$= [\![fail]\!]_* \bot$

$= \bot$

$\neq \langle abort, \sigma\rangle = [\![fail]\!]\sigma$

(c) $[\![\text{newvar } v := e \text{ in } v := v+1; fail]\!]\sigma$

$= [\![fail]\!]_*([\![\text{newvar } v := e \text{ in } v := v+1]\!]\sigma)$

$= [\![fail]\!]_*((\lambda\sigma' \in \Sigma : [\sigma' \mid v : \sigma v])_*([\![v := v+1]\!][\sigma \mid v : [\![e]\!]\sigma]))$

$= [\![fail]\!]_*((\lambda\sigma' \in \Sigma : [\sigma' \mid v : \sigma v])[\underbrace{\sigma \mid v : [\![e]\!]\sigma \mid v : [\![e]\!]\sigma + 1}_{\sigma'}])$

$= [\![fail]\!][\sigma' \mid v : \sigma v]$

$= \langle abort, \sigma\rangle$

$[\![\text{newvar } w := e \text{ in } w := w+1; fail]\!]\sigma$

$= [\![fail]\!]_*([\![\text{newvar } w := e \text{ in } w := w+1]\!]\sigma)$

$= [\![fail]\!]_*((\lambda\sigma' \in \Sigma : [\sigma' \mid w : \sigma w])_*([\![w := w+1]\!][\sigma \mid w : [\![e]\!]\sigma]))$

$= [\![fail]\!]_*((\lambda\sigma' \in \Sigma : [\sigma' \mid w : \sigma w])[\underbrace{\sigma \mid w : [\![e]\!]\sigma \mid w : [\![e]\!]\sigma + 1}_{\sigma'}])$

$= [\![fail]\!][\sigma' \mid w : \sigma w]$

$= \langle abort, \sigma\rangle$

$[\![\text{newvar } v := e \text{ in } v := v+1; fail]\!] \equiv [\![\text{newvar } w := e \text{ in } w := w+1; fail]\!]$

(d) $[\![\text{while } b \text{ do } fail]\!]\sigma = \bigsqcup_{i=0}^{\infty} F^i \bot_{\Sigma \to \Sigma_\bot} \sigma$

$$F^0_{\bot_\Sigma \to \Sigma \bot} = \bot_{\Sigma \to \Sigma \bot}$$

$$F_{\bot_\Sigma \to \Sigma \bot} = \begin{cases} (\bot_{\Sigma \to \Sigma \bot}) \circ [\![fail]\!] \sigma & [\![b]\!] \sigma \\ \sigma & \neg [\![b]\!] \sigma \end{cases}$$

$$= \begin{cases} \sigma & \neg [\![b]\!] \sigma \\ \langle abort, \sigma \rangle & [\![b]\!] \sigma \end{cases}$$

$$F^2_{\bot_{\Sigma \to \Sigma \bot}} = \begin{cases} (F_{\bot_{\Sigma \to \Sigma \bot}}) \circ [\![fail]\!] \sigma & [\![b]\!] \sigma \\ \sigma & \neg [\![b]\!] \sigma \end{cases}$$

$$= \begin{cases} \langle abort, \sigma \rangle & [\![b]\!] \sigma \\ \sigma & \neg [\![b]\!] \sigma \end{cases}$$

$$F^i_{\bot_{\Sigma \to \Sigma \bot}} = \begin{cases} \langle abort, \sigma \rangle & [\![b]\!] \sigma \\ \sigma & \neg [\![b]\!] \sigma \end{cases}$$

$$g = \bigsqcup^i \bot_{\Sigma \to \Sigma \bot} = \begin{cases} \langle abort, \sigma \rangle & [\![b]\!] \sigma \\ \sigma & \neg [\![b]\!] \sigma \end{cases}$$

$[\![\text{while } b \text{ do fail}]\!] \sigma = g$

$[\![\text{if } b \text{ then fail else skip}]\!] \sigma$

$$= \text{if } [\![b]\!] \sigma \text{ then } [\![fail]\!] \sigma \text{ else } [\![skip]\!] \sigma$$

$$= \begin{cases} \langle abort, \sigma \rangle & [\![b]\!] \sigma \\ \sigma & \neg [\![b]\!] \sigma \end{cases}$$

$[\![\text{while } b \text{ do fail}]\!] \sigma = [\![\text{if } b \text{ then fail else skip}]\!] \sigma$

(e) $[\![x := 0 ; \text{catch } x := 1 \text{ in while } x < 1 \text{ do fail}]\!] \sigma$

$= [\![\text{catch } x := 1 \text{ in while } x < 1 \text{ do fail}]\!] * ([\![x := 0]\!] \sigma)$

$= [\![\text{catch } x := 1 \text{ in while } x < 1 \text{ do fail}]\!] [\sigma | x : 0]$

$= [\![x := 1]\!]_\dagger [\![\text{while } x < 1 \text{ do fail}]\!] [\sigma | x : 0]$

(tomando g del ejercicio anterior)

$= [\![x := 1]\!]_\dagger \langle abort, [\sigma | x : 0] \rangle$

$= [\![x := 1]\!] [\sigma | x : 0]$

$= [\sigma \mid x : 1]$

$[\![x := 0 ; \text{while } x < 1 \text{ do catch } x := 1 \text{ in fail}]\!]\sigma =$
$[\![\text{while } x < 1 \text{ do catch } x := 1 \text{ in fail}]\!]_* ([\![x := 0]\!]\sigma) =$
$[\![\text{while } x < 1 \text{ do catch } x := 1 \text{ in fail}]\!] [\sigma \mid x : 0)$

$[\![\text{while } x < 1 \text{ do catch } x := 1 \text{ in fail}]\!]_\sigma = \bigsqcup_{i=0}^{\infty} F^i_{\Sigma \to \Sigma_\perp} \sigma$

$$F\omega\sigma = \begin{cases} \omega_* [\![\text{catch } x := 1 \text{ in fail}]\!]\sigma & [\![x < 1]\!]\sigma \\ \sigma & c.c \end{cases}$$
$$= \begin{cases} \omega_* ([\![x := 1]\!]_+ [\![\text{fail}]\!]\sigma) & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$
$$= \begin{cases} \omega_* ([\![x := 1]\!] (\text{abort}, \sigma)) & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$
$$= \begin{cases} \omega_* [\sigma \mid x : 1] & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$

$$F^0_{\Sigma \to \Sigma_\perp} \sigma = \perp_{\Sigma \to \Sigma_\perp}$$
$$F \perp_{\Sigma \to \Sigma_\perp} \sigma = \begin{cases} \perp & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$
$$= \begin{cases} \sigma & \sigma x \geq 1 \\ \perp & \sigma x < 1 \end{cases}$$
$$F^2 \perp_{\Sigma \to \Sigma_\perp} \sigma = \begin{cases} (F \perp_{\Sigma \to \Sigma_\perp})_* [\sigma \mid x : 1] & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$
$$= \begin{cases} [\sigma \mid x : 1] & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$
$$F^3 \perp_{\Sigma \to \Sigma_\perp} \sigma = \begin{cases} (F^2 \perp_{\Sigma \to \Sigma_\perp})_* [\sigma \mid x : 1] & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$

$$= \begin{cases} [\sigma \mid x:1] & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$

$$\bigcup_{i=0}^{\infty} F^i \perp_{\{x<1\}} \sigma = \begin{cases} [\sigma \mid x:1] & \sigma x < 1 \\ \sigma & \sigma x \geq 1 \end{cases}$$

$[\![\text{while } x<1 \text{ do catch } x:=1 \text{ in fail}]\!]\sigma\,[\sigma \mid x:0]$
$= [\sigma \mid x:1]$

$[\![x:=0 \text{; while } x<1 \text{ do catch } x:=1 \text{ in fail}]\!]\sigma \equiv$
$[\![x:=0 \text{; catch } x:=1 \text{ in while } x<1 \text{ do fail}]\!]\sigma$

En el lenguaje sin fallas el a si es equivalente ya que
$[\![\text{while true do skip}]\!]\sigma = \perp$

(3) Computar la semántica operacional en estados σ tales que $\sigma x = 2, \sigma y = 1$. de los siguientes programas

(1) $y := x + y$; **if** $y > 0$ **then** $x := x - 1$ **else skip**

(2) **while** $x > 0$ **do**
($y := x + y$; **if** $y > 0$ **then** $x := x - 1$ **else skip**)

(1) $\langle y:=x+y; \text{if } y>0 \text{ then } x:=x-1 \text{ else skip}, [\sigma \mid x:2 \mid y:1] \rangle \rightarrow$
$\langle \text{if } y>0 \text{ then } x:=x-1 \text{ else skip}; [\sigma \mid x:2 \mid y:3] \rangle \rightarrow$
$\langle [\sigma \mid x:2 \mid y:3 \mid x:1] \rangle \rightarrow \langle [\sigma \mid y:3 \mid x:1] \rangle$

$[\![\{\{ y:=x+y; \text{if } y>0 \text{ then } x:=x-1 \text{ else skip} \}\}]\!]\sigma = [\sigma \mid y:3 \mid x:1]$
con $\sigma x=2, \sigma y=1$

(2) $\langle \text{while } x>0 \text{ do } y:=x+y; \text{if } y>0 \text{ then } x:=x-1 \text{ else skip}, [\sigma \mid x:2 \mid y:1] \rangle \rightarrow$
$\langle y:=x+y; \text{if } y>0 \text{ then } x:=x-1 \text{ else skip}; \text{while } x>0 \text{ do } \ldots, [\sigma \mid x:2 \mid y:1] \rangle \rightarrow$
$\langle \text{if } y>0 \text{ then } x:=x-1 \text{ else skip}; \text{while } x>0 \text{ do } \ldots, [\sigma \mid x:2 \mid y:3] \rangle \rightarrow$

$\langle x := x-1;\ \text{while } x>0 \text{ do } \ldots,\ [\sigma \mid x:2 \mid y:3] \rangle \rightarrow$

$\langle \text{while } x>0 \text{ do } y := x+y;\ \text{if } y>0 \text{ then } x := x-1 \text{ else skip},\ [\sigma \mid x:1 \mid y:3] \rangle \rightarrow$

$\langle y := x+y;\ \text{if } y>0 \text{ then } x := x-1 \text{ else skip};\ \text{while } \ldots,\ [\sigma \mid x:1 \mid y:3] \rangle \rightarrow$

$\langle \text{if } y>0 \text{ then } x := x-1 \text{ else skip};\ \text{while } \ldots,\ [\sigma \mid x:1 \mid y:4] \rangle \rightarrow$

$\langle x := x-1;\ \text{while } \ldots,\ [\sigma \mid x:1 \mid y:4] \rangle \rightarrow$

$\langle \text{while } x>0 \text{ do } \ldots,\ [\sigma \mid x:0 \mid y:4] \rangle \rightarrow$

$\langle [\sigma \mid x:0 \mid y:4] \rangle$

$\{\{\text{while } x>0 \text{ do } y := y+x;\ \text{if } y>0 \text{ then } x := x-1 \text{ else skip}\}\}\ [\sigma \mid x:2 \mid y:1] = [\sigma \mid x:0 \mid y:4]$

(4) Computar la semántica operacional del siguiente programa en un estado arbitrario

newvar x := 2 **in**
  while x > 0 **do**
    y := x + y; **if** y > 0 **then** x := x − 1 **else skip**

$\langle \text{newvar } x := 2 \text{ in while } x>0 \text{ do } y := x+y;\ \text{if } y>0 \text{ then } x := x-1 \text{ else skip}, \sigma \rangle \rightarrow$

$\langle \text{newvar } x := 2 \text{ in } y := x+y\ \text{if } y>0 \text{ then } x := x-1 \text{ else skip}, \text{while } \ldots, [\sigma' \mid x:\sigma x] \rangle \rightarrow$

$\langle \text{newvar } x := 2 \text{ in if } y>0 \text{ then } x := x-1 \text{ else skip}; \text{while } \ldots, [\sigma' \mid x:\sigma x \mid y:\sigma y+2] \rangle \rightarrow$

sup. $\sigma y > 0$ $\quad \langle \text{newvar } x := 1 \text{ in while } x>0 \text{ do } \ldots, [\sigma' \mid x:\sigma x \mid y:\sigma y+2] \rangle \rightarrow$

$\langle \text{newvar } x := 1 \text{ in } y := x+y;\ \text{if } x>0 \text{ then } x := x-1 \text{ else skip}; \text{while } \ldots, [\sigma' \mid x:\sigma x \mid y:\sigma y+2] \rangle \rightarrow$

$\langle \text{newvar } x := 1 \text{ in if } x>0 \text{ then } x := x-1 \text{ else skip}; \text{while } x>0 \ldots, [\sigma \mid x:\sigma x \mid y:\sigma y+2+1] \rangle \rightarrow$

$\langle \text{newvar } x := 0 \text{ while } x>0 \text{ do } \ldots, [\sigma' \mid x:\sigma x \mid y:\sigma y+2+1] \rangle \rightarrow$

$\langle [\sigma' \mid x:\sigma x \mid y:\sigma y+2+1] \rangle$

$\{\{\text{newvar } x := 2 \text{ in while } x>0 \text{ do } y := x+y;\ \text{if } y>0 \text{ then } x := x-1 \text{ else skip}\}\}\sigma$

si $\sigma y > 0$ $\quad = [\sigma' \mid x:\sigma x \mid y:\sigma y+2+1]$

si $\sigma y < 0 \ \wedge \ \sigma y$ impar $\quad = [\sigma' \mid x:\sigma x \mid y:4]$

si $\sigma y < 0 \ \wedge \ \sigma y$ par $\quad = [\sigma' \mid x:\sigma x \mid y:5]$

(5) Pruebe:

(a) Si $\langle c_0, \sigma\rangle \to^* \sigma'$, entonces $\langle c_0; c_1, \sigma\rangle \to^* \langle c_1, \sigma'\rangle$

(b) Si $\langle c, [\sigma|\mathsf{x} : [\![e]\!]\sigma]\rangle \to^* \sigma'$, entonces

$$\langle \textbf{newvar } \mathsf{x} := e \textbf{ in } c, \sigma\rangle \to^* [\sigma'|\mathsf{x} : \sigma\mathsf{x}].$$

(c) Si $\langle c, [\sigma|\mathsf{x} : [\![e]\!]\sigma]\rangle \to^* \langle c', \sigma'\rangle$, entonces

$$\langle \textbf{newvar } \mathsf{x} := e \textbf{ in } c, \sigma\rangle \to^* \langle \textbf{newvar } \mathsf{x} := \sigma' x \textbf{ in } c', [\sigma'|\mathsf{x} : \sigma x]\rangle$$

Suponemos $\langle c_0, \sigma\rangle$.

(a) $\langle c_0, \sigma\rangle \to^* \sigma' = \langle c_0^0, \sigma^0\rangle \to \langle c_0^1, \sigma'\rangle \to \dots \to \langle c_0^{n-1}, \sigma^{n-1}\rangle \to \sigma'$ con $n \geq 1$

Por ind.

Caso $n = 1$

$$\langle c_0, \sigma\rangle \to \sigma'$$

entonces $$\frac{\langle c_0, \sigma\rangle \to \sigma'}{\langle c_0; c_1, \sigma\rangle \to \langle c_1, \sigma'\rangle}$$

Por H.I como $\langle c_0^1, \sigma'\rangle \to^* \sigma'$ entonces $\langle c_0; c_1, \sigma^1\rangle \to^* \langle c_1, \sigma'\rangle$

$$\frac{\langle c_0, \sigma\rangle \to \langle c_0^1, \sigma^1\rangle}{\langle c_0; c_1, \sigma\rangle \to \langle c_0^1; c_1, \sigma^1\rangle}$$

y por H.I $\langle c_0^1; c_1, \sigma^1\rangle \to^* \langle c_1, \sigma'\rangle$ luego $\langle c_0; c_1, \sigma\rangle \to^* \langle c_1, \sigma'\rangle$

(b) Supongamos $\langle c, [\sigma|x : [\![e]\!]\sigma]\rangle \to^* \sigma'$

entonces $\langle c, [\sigma|x : [\![e]\!]\sigma]\rangle = \langle c_0, \sigma^0\rangle \to \langle c_1, \sigma^1\rangle \to \dots \to \langle c_n, \sigma^{n-1}\rangle \to \sigma'$ $n \geq 1$

Caso $n = 1$

$$\langle c_0, \sigma^0\rangle = \langle c, [\sigma|x : [\![e]\!]\sigma]\rangle \to \sigma'$$

$$\frac{}{\langle \text{newvar } x := e \text{ in } c, \sigma\rangle \to [\sigma'|x : \sigma x]}$$

H.I

$$\langle \text{newvar } x := e \text{ in } c, \sigma\rangle \to \langle \text{newvar } x := \sigma^1 x \text{ in } c_1, [\sigma'|x : \sigma x]\rangle$$

$\sigma^0 = [\sigma^1|x : \sigma x]$

$\sigma^1 = [\sigma^0|x : \sigma^1 x]$

$\langle c_1, [\sigma^0|x : \sigma^1 x]\rangle \to^* \sigma'$

$\langle \text{newvar } x := \sigma' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle \to^* [\sigma' | x : \sigma x]$

Por H.I $\langle c_1, \sigma' \rangle \to^* \sigma'$ entonces $\langle \text{newvar } x := \sigma'' x \text{ in } c_1, \sigma_0 \rangle \to^* [\sigma' | x : \sigma x]$

$$\frac{\langle c, \sigma \rangle \to \langle c_1, \sigma' \rangle}{\langle \text{newvar } x := e \text{ in } c, \sigma \rangle \to \langle \text{newvar } x := \sigma' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle}$$

y por H.I $\langle \text{newvar } x := \sigma'' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle \to^* [\sigma' | x : \sigma x]$

luego $\langle \text{newvar } x := e \text{ in } c, \sigma \rangle \to^* [\sigma' | x : \sigma x]$

(c) Supongamos $\langle c, [\sigma | x : [\![e]\!]\sigma] \rangle \to^* \sigma'$

entonces $\langle c, [\sigma | x : [\![e]\!]\sigma] \rangle = \langle c_0, \sigma^0 \rangle \to \langle c_1, \sigma^1 \rangle \to \dots \to \langle c_n, \sigma^{n-1} \rangle \to (c', \sigma') \quad n \geq 1$

Caso $n = 1$

$$\langle c_0, \sigma^0 \rangle = \langle c, [\sigma | x : [\![e]\!]\sigma] \rangle \to \langle c', \sigma' \rangle$$

$$\langle \text{newvar } x := e \text{ in } c, \sigma \rangle \to \langle \text{newvar } x := \sigma' x \text{ in } c', [\sigma' | x : \sigma x] \rangle$$

H.I

$$\langle \text{newvar } x := e \text{ in } c, \sigma \rangle \to \langle \text{newvar } x := \sigma' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle$$

$\sigma^0 = [\sigma' | x : \sigma x]$

$\sigma^1 = [\sigma^0 | x : \sigma' x]$

$\langle c_1, [\sigma^0 | x : \sigma' x] \rangle \to^* \langle c', \sigma' \rangle$

$\langle \text{newvar } x := \sigma' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle \to^* \langle \text{newvar } x := \sigma' x \text{ in } c', [\sigma' | x : \sigma x] \rangle$

Por H.I

$$\frac{\langle c, \sigma \rangle \to \langle c_1, \sigma' \rangle}{\langle \text{newvar } x := e \text{ in } c, \sigma \rangle \to \langle \text{newvar } x := \sigma' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle}$$

y por H.I $\langle \text{newvar } x := \sigma'' x \text{ in } c_1, [\sigma' | x : \sigma x] \rangle \to^* \langle \text{newvar } x := \sigma' x \text{ in } c', [\sigma' | x : \sigma x] \rangle$

luego $\langle \text{newvar } x := e \text{ in } c, \sigma \rangle \to^* \langle \text{newvar } x := \sigma' x \text{ in } c', [\sigma' | x : \sigma x] \rangle$

(6) Demostrar la corrección de la semántica operacional one-step (incluyendo **fail**) respecto de la semántica denotacional, demostrando simultáneamente

(a) $\langle c, \sigma \rangle \to \sigma' \Longrightarrow [\![c]\!]\sigma = \sigma'$.

(b) $\langle c, \sigma \rangle \to \langle \mathbf{abort}, \sigma' \rangle \Longrightarrow [\![c]\!]\sigma = \langle \mathbf{abort}, \sigma' \rangle$.

(c) $\langle c, \sigma \rangle \to \langle c', \sigma' \rangle \Longrightarrow [\![c]\!]\sigma = [\![c']\!]\sigma'$.

Use inducción en la derivación.

(a) Supongamos $\langle c, \sigma \rangle \to \sigma'$

$c = skip$ — caso base

$[\![skip]\!]\sigma = \sigma$ y $\langle skip, \sigma \rangle \to \sigma$

$c = v := e$

$\langle v := e, \sigma \rangle \to [\sigma \mid v : [\![e]\!]\sigma]$

$[\![v := e]\!]\sigma = [\sigma \mid v : [\![e]\!]\sigma$

$c =$ while b do c', $[\![b]\!]\sigma = F$

$[\![$while b do $c']\!]\sigma = \sigma$ $\quad [\![b]\!]\sigma = F$

$\dfrac{([\![b]\!]\sigma = F)}{\langle$while b do $c', \sigma\rangle \to \sigma}$

$c =$ newvar $v := e$ in c — caso inductivo

$$\frac{\langle c', [\sigma \mid v : [\![e]\!]\sigma] \rangle \to \sigma_1}{\langle \text{newvar } v := e \text{ in } c', \sigma \rangle \to \underbrace{[\sigma_1 \mid v : \sigma v]}_{\sigma'}}$$

$[\![c']\!][\sigma \mid v : [\![e]\!]\sigma] = \sigma_1$

$[\![$newvar $v := e$ in $c]\!]\sigma = (\lambda\sigma'.[\sigma' \mid v : \sigma v]) \# [\![c]\!][\sigma \mid v : [\![e]\!]\sigma]$

$= (\lambda\sigma'.[\sigma' \mid v : \sigma v])\,\sigma_1$

$= \underbrace{[\sigma_1 \mid v : \sigma v]}_{\sigma'}$

(b) $\langle c, \sigma \rangle \to \langle abort, \sigma \rangle$

$c = fail$ — caso base

$\langle fail, \sigma \rangle \to \langle abort, \sigma \rangle$

$[\![fail]\!]\sigma = \langle abort, \sigma \rangle$

$c = c_0; c_1$ con $\dfrac{\langle c_0, \sigma \rangle \to \langle abort, \sigma' \rangle}{\langle c_0; c_1, \sigma \rangle \to \langle abort, \sigma' \rangle}$ — casos inductivos

$[\![c_0; c_1]\!]\sigma = [\![c_1]\!]([\![c_0]\!]\sigma) = [\![c_1]\!]\langle abort, \sigma' \rangle$

$= \langle abort, \sigma' \rangle$

$c =$ newvar $v := e$ in c' con $\dfrac{\langle c', [\sigma \mid v : [\![e]\!]\sigma] \rangle \to \langle abort, \sigma' \rangle}{\langle \text{newvar } v := e \text{ in } c' \rangle \to \langle abort, \underbrace{[\sigma' \mid v : \sigma v]}_{\sigma''} \rangle}$.

$$[\![\text{newvar } v:=e \text{ in } c']\!]\sigma = (\lambda\sigma'.[\sigma'|v:\sigma v])_\perp([\![c']\!][\sigma|v:[\![e]\!]\sigma])$$
$$= (\lambda\sigma'.[\sigma'|v:\sigma v]) \to (\text{abort}, \sigma)$$
$$= \langle \text{abort}, [\sigma'|v:\sigma v]\rangle$$

(c) Suponganos $\langle c,\sigma\rangle \to \langle c',\sigma'\rangle$

$c = \text{if } b \text{ then } c_0 \text{ else } c_1$ — casos base

$$\frac{[\![b]\!]\sigma = T}{\langle c,\sigma\rangle \to \langle c_0,\sigma\rangle}$$

$$[\![c]\!]\sigma = [\![c_0]\!]\sigma$$

$$\frac{[\![b]\!]\sigma = F}{\langle c,\sigma\rangle \to \langle c_1,\sigma\rangle}$$

$$[\![c]\!]\sigma = [\![c_1]\!]\sigma$$

$c = \text{while } b \text{ do } c'$

$$\frac{[\![b]\!]\sigma = T}{\langle c,\sigma\rangle \to \langle c'; \text{while } b \text{ do } c', \sigma\rangle}$$

$$[\![\text{while } b \text{ do } c']\!]\sigma = [\![c'; \text{while } b \text{ do } c']\!]\sigma$$
$$= [\![\text{while } b \text{ do } c]\!]_\perp([\![c']\!]\sigma)$$
$$= F\,[\![\text{while } b \text{ do } c]\!]\sigma$$
$$= [\![\text{while } b \text{ do } c']\!]\sigma$$

$w = [\![\text{while } b \text{ do } c']\!]$, $Fw = w$

$c = c_0; c_1$ — casos inductivos

$$\frac{\langle c_0,\sigma\rangle \to \langle c_0',\sigma'\rangle}{\langle c_0; c_1,\sigma\rangle \to \langle c_0'; c_1,\sigma'\rangle}$$

$$[\![c_0; c_1]\!]\sigma = [\![c_1]\!]_*([\![c_0]\!]\sigma)$$
$$= [\![c_1]\!]_*[\![c_0']\!]\sigma'$$
$$= [\![c_0'; c_1]\!]\sigma'$$

$c = \text{newvar } v:=e \text{ in } c'$

$$\frac{\langle c', [\sigma|v:[\![e]\!]\sigma]\rangle \to \langle c'',\sigma'\rangle}{\langle \text{newvar } v:=e \text{ in } c',\sigma\rangle \to \langle \text{newvar } v:=\sigma' v \text{ in } c'', [\sigma'|v:\sigma v]\rangle}$$

$$[\![\text{newvar } v:=e \text{ in } c']\!]\sigma' = (\lambda\sigma'.[\sigma'|v:\sigma v])_*([\![c']\!][\sigma|v:[\![e]\!]\sigma])$$
$$= (\lambda\sigma'.[\sigma'|v:\sigma v])_*([\![c'']\!]\sigma')$$
$$= [\![\text{newvar } v:=\sigma' v \text{ in } c'']\!][\sigma'|v:\sigma v]$$

(7) Demostrar simultáneamente:
(a) $\langle c, \sigma\rangle \to^* \sigma' \Longleftarrow [\![c]\!]\sigma = \sigma'$.
(b) $\langle c, \sigma\rangle \to^* \langle \mathbf{abort}, \sigma'\rangle \Longleftarrow [\![c]\!]\sigma = \langle \mathbf{abort}, \sigma'\rangle$.
Use inducción en la estructura de c. Tendrá que usar los resultados probados en el ejercicio 5. El caso **while** requiere a su vez una inducción en el mínimo n tal que $F^n(\bot)\sigma \neq \bot$.

Inducción en c

Casos base

c = skip

$[\![c]\!]\sigma = \sigma$

$\langle c, \sigma\rangle \to \sigma$ por ende $\langle c, \sigma\rangle \to^* \sigma$

c = v := e

$[\![c]\!]\sigma = [\sigma \mid v : [\![e]\!]\sigma]$

$\langle c, \sigma\rangle \to [\sigma \mid v : [\![e]\!]\sigma]$

c = fail

$[\![\text{fail}]\!]\sigma = \langle \text{abort}, \sigma\rangle$

$\langle \text{fail}, \sigma\rangle \to \langle \text{abort}, \sigma\rangle$

Casos inductivos

$c = c_0 ; c_1$ con $[\![c_0]\!]\sigma = \sigma_0$

$[\![c]\!]\sigma = [\![c_1]\!]_\sharp([\![c_0]\!]\sigma)$

$= [\![c_1]\!]\sigma_0$

$= \sigma'$

H.I $\langle c_0, \sigma\rangle \to^* \sigma_0$

$\langle c_1, \sigma_0\rangle \to^* \sigma'$

Por ej. 5 y H.I

$\langle c_0 ; c_1, \sigma\rangle \to^* \langle c_1, \sigma'\rangle \to^* \sigma'$

con $[\![c_0]\!]\sigma = \langle \text{abort}, \sigma_0\rangle$

$[\![c]\!]\sigma = [\![c_1]\!]_\sharp([\![c_0]\!]\sigma)$

$= [\![c_1]\!]_\sharp \langle \text{abort}, \sigma_0\rangle$

$= \langle \text{abort}, \sigma_0\rangle$

H.I $\langle c_0, \sigma\rangle \to^* \langle abort, \sigma_0\rangle$

$\langle c_1, \langle abort, \sigma_0\rangle\rangle \to^* \langle abort, \sigma\rangle$

Por ej 5 y H.I

$\langle c_0; c_1, \sigma\rangle \to \langle c_1, \langle abort, \sigma_0\rangle\rangle \to^* \langle abort, \sigma\rangle$

Los casos con $[\![c_1]\!]\sigma_0 = \langle abort, \sigma_1\rangle$ son totalmente análogos.

$c =$ if b then c_0 else c_1 con $[\![c_0]\!]\sigma = \sigma_0$

Solo probamos caso $[\![b]\!]_\sigma = T$ ya que son análogos

$[\![c]\!]\sigma = [\![c_0]\!]\sigma$

$= \sigma_0$

H.I $\langle c_0, \sigma\rangle \to^* \sigma_0$

$$\frac{[\![b]\!]_\sigma = T}{\langle c, \sigma\rangle \to \langle c_0, \sigma\rangle}$$

Y por H.I $\langle c_0, \sigma\rangle \to^* \sigma_0$, luego $\langle c, \sigma\rangle \to^* \sigma_0$.

con $[\![c_0]\!]\sigma = \langle abort, \sigma_0\rangle$

$[\![c]\!]\sigma = [\![c_0]\!]\sigma$

$= \langle abort, \sigma_0\rangle$

H.I $\langle c_0, \sigma\rangle \to^* \langle abort, \sigma_0\rangle$

$$\frac{[\![b]\!]\sigma = T}{\langle c, \sigma\rangle \to \langle c_0, \sigma\rangle}$$

Y por H.I $\langle c_0, \sigma\rangle \to^* \langle abort, \sigma_0\rangle$ luego $\langle c, \sigma\rangle \to^* \langle abort, \sigma_0\rangle$

$c =$ newvar $v := e$ in c' con $[\![c']\!][\sigma | v : [\![e]\!]\sigma] = \sigma'$

$[\![c]\!]\sigma = (\lambda\sigma'.[\sigma' | v : \sigma v]) * ([\![c']\!][\sigma | v : [\![e]\!]\sigma])$

$= (\lambda\sigma'.[\sigma' | v : \sigma v])\,\sigma'$

$= [\sigma' | v : \sigma v]$

H.I $\langle c', [\sigma | v : [\![e]\!]\sigma]\rangle \to^* \sigma'$

por ej 5 y H.I

$\langle$newvar $v := e$ in $c', \sigma\rangle \to^* [\sigma' | v : \sigma v]$

$c =$ newvar $v := e$ in c' con $[\![c']\!][\sigma | v : [\![e]\!]\sigma] = \langle abort, \sigma' \rangle$

$$[\![c]\!]\sigma = (\lambda \sigma'. [\sigma' | v : \sigma v]) * ([\![c']\!][\sigma | v : [\![e]\!]\sigma])$$
$$= (\lambda \sigma', [\sigma' | v : \sigma v]) \langle abort, \sigma \rangle$$
$$= \langle abort, [\sigma' | v : \sigma v] \rangle$$

H.I $\quad \langle c', [\sigma | v : [\![e]\!]\sigma] \rangle \to \langle abort, \sigma' \rangle$

por ej 5 y H.I

$$\frac{\langle c', [\sigma | v : [\![e]\!]\sigma] \rangle \to \langle abort, \sigma' \rangle}{\langle \text{newvar } v \text{ in } c', \sigma \rangle \to \langle abort, [\sigma' | v : \sigma v] \rangle}$$

$c =$ catchin c_0 with c_1 $\quad [\![c_0]\!]\sigma = \sigma_0$

$$[\![c]\!]\sigma = [\![c_1]\!]_\perp ([\![c_0]\!]\sigma)$$
$$= [\![c_1]\!]_\perp \sigma_0$$
$$= \sigma_0$$

H.I $\quad \langle c_0, \sigma \rangle \xrightarrow{*} \sigma_0$

Por ej. 5, H.I y def. de catchin

$$\langle c_0, \sigma \rangle \xrightarrow{*} \sigma_0 \Rightarrow \langle c, \sigma \rangle \xrightarrow{*} \sigma_0$$

$[\![c_0]\!]\sigma = \langle abort, \sigma_0 \rangle$

$$[\![c]\!]\sigma = [\![c_1]\!]_\perp ([\![c_0]\!]\sigma)$$
$$= [\![c_1]\!]_\perp \langle abort, \sigma_0 \rangle$$
$$= \sigma_1$$

H.I $\quad \langle c_0, \sigma \rangle \xrightarrow{*} \langle abort, \sigma_0 \rangle \quad \langle c_1, \sigma_0 \rangle \xrightarrow{*} \sigma_1$

Por ej. 5, H.I y def. de catchin

$$\langle c_0, \sigma \rangle \xrightarrow{*} \langle abort, \sigma_0 \rangle \Rightarrow \langle c, \sigma \rangle \xrightarrow{*} \langle c_1, \sigma_0 \rangle \xrightarrow{*} \sigma_1$$

$c =$ while b do c_0

$$[\![c]\!]\sigma = \bigcup_{i=0}^{\infty} F^i \perp \sigma = \sigma'$$

Inducción en k tal que $F^k \perp \sigma \neq \perp$

$$F^k \perp \sigma \overset{\forall}{=} \sigma'$$

$k=2$ $\quad F^2 \perp \sigma \neq \perp \iff [\![b]\!]\sigma = F$

entonces $\dfrac{[\![b]\!]\sigma = F}{\langle \text{while } b \text{ do } c_0, \sigma\rangle \to \sigma}$

Paso inductivo

$F^{k-1} \perp \sigma = \perp$ y $F^k \perp \sigma \neq \perp$ y $[\![b]\!]\sigma = T$

$F^k \perp \sigma = (F^{k-1}\perp)_\star (\underbrace{[\![c_0]\!]\sigma}_{\sigma_0}) = \sigma'$

Por H.I sobre k $\quad \langle \text{while } b \text{ do } c_0, \sigma_0\rangle \to^* \sigma'$

Por H.I (c_0) $\quad \langle c_0, \sigma\rangle \to^* \sigma_0$

ej.s $\quad \langle c_0; \text{while } b \text{ do } c_0, \sigma_0\rangle \to^* \sigma'$

$\langle \text{while } b \text{ do } c_0, \sigma\rangle \to^* \sigma'$

(8) Demostrar la equivalencia entre la semántica denotacional y la operacional del lenguaje imperativo simple incluyendo **fail**. Debe salir inmediatamente de los dos ejercicios anteriores.

$[\![c]\!] = \{\{c\}\}$

Si $[\![c]\!]\sigma = \sigma'$ entonces por ej. anterior $\langle c, \sigma\rangle \to^* \sigma'$

Si $[\![c]\!]\sigma = \langle \text{abort}, \sigma'\rangle$ entonces por ej. anterior $\langle c, \sigma\rangle \to^* \langle \text{abort}, \sigma'\rangle$

Si $[\![c]\!]\sigma = \perp$ Hay que ver que $\langle c, \sigma\rangle \uparrow$

Supongamos $\langle c, \sigma\rangle \to^* \sigma'$

entonces $\langle c, \sigma\rangle = \langle c_0, \sigma_0\rangle \to \langle c_1, \sigma_1\rangle \to \dots \to \langle c_{n-1}, \sigma_{n-1}\rangle \to \sigma' \quad n \geq 1$

por ej.6 esto nos dice que $[\![c_{n-1}]\!]\sigma_{n-1} = \sigma'$

pero esto significa que $c = c_0; c_1; \dots; c_{n-1}$ y

$[\![c]\!]\sigma = [\![c_1; \dots; c_{n-1}]\!]_\star ([\![c_0]\!]\sigma)$

$= [\![c_1; \dots; c_{n-1}]\!]\sigma_1$

$= [\![c_2; \dots; c_{n-1}]\!]_\star ([\![c_1]\!]\sigma_1)$

$\vdots$

$= [\![c_{n-1}]\!]_\star \sigma_{n-1}$

$= \sigma'$ Lo cual contradice la hipótesis