Lenguajes y compiladores
Práctico 6

Repaso.

(1) Decida si la función $H\colon (\Sigma \to \Sigma'_\perp) \to (\Sigma \to \Sigma'_\perp)$ es monótona y si es continua.

$$H\,f\,\sigma = \begin{cases} \sigma & \text{si } f = \perp \\ \perp & \text{en caso contrario} \end{cases}$$

(2) Considere la función $G\colon (\Sigma \to \Sigma'_\perp) \to (\Sigma \to \Sigma'_\perp)$ definida por

$$G\,f\,\sigma = \begin{cases} \perp & \text{si } f \text{ es parcial} \\ \sigma & \text{en caso contrario} \end{cases}$$

(a) Decida si G es continua o no.

(b) Decida si existe un comando **while** b **do** c, tal que $G = F$, donde F es el funcional asociado al **while**.

(1) Sean $g, f \in (\Sigma \to \Sigma'_\perp)$ tales que $f \leq g$

Si $g = \perp_{\Sigma \to \Sigma'_\perp}$ entonces claramente $f = \perp_{\Sigma \to \Sigma'_\perp}$

luego
$$f \leq g$$
$$H f \sigma \leq H g \sigma$$
$$\sigma \leq \sigma$$

Si $g \neq \perp_{\Sigma \to \Sigma'_\perp}$ y $f = \perp_{\Sigma \to \Sigma'_\perp}$
$$f \leq g$$
pero $H f \sigma = \sigma \not\leq \perp = g$

Entonces la función no es monótona.

Si H fuese continua también fuese monótona, por ende no es continua.

(2) Sean $g, f \in (\Sigma \to \Sigma'_\perp)$ tales que $f \leq g$

Si g y f son parciales

luego
$$f \leq g$$
$$G f \sigma \leq G g \sigma \qquad (f \text{ y } g \text{ son parciales})$$
$$\perp \leq \perp$$

Si f es parcial y g no
$$f \leq g$$

$$G f \sigma \leq G g \sigma$$
$$\perp \leq \sigma$$

Si $f \leq g$ no se puede dar que g sea parcial y f total ya que o $g \leq f$ o no habría comparación entonces sólo queda el caso en que f y g son totales

$$f \leq g$$
$$G f \sigma \leq G g \sigma$$
$$\sigma \leq \sigma$$

G es monótona.

Como $(\Sigma \to \Sigma'_\perp) \to (\Sigma \to \Sigma'_\perp)$ es un predominio sólo hay que ve si
$G(\sup\{f_i \mid i \in \mathbb{N}\}) \leq \sup\{G f_i \mid i \in \mathbb{N}\}$

sea $f_1 \leq \ldots \leq f_n \leq \ldots$ una cadena interesante

$$G(\sup\{f_i \mid i \in \mathbb{N}\})\sigma = \begin{cases} \perp & \text{si } \sup\{f_i \mid i \in \mathbb{N}\} \text{ es parcial} \\ \sigma & \text{c.c} \end{cases}$$

$$G f_i \sigma = \begin{cases} \perp & \text{si } f_i \text{ es parcial} \\ \sigma & \text{c.c} \end{cases}$$

si $\sup\{f_i \mid i \in \mathbb{N}\}$ es parcial entonces f_i es parcial $\forall i \in \mathbb{N}$ y

$$\begin{aligned} G(\sup\{f_i \mid i \in \mathbb{N}\})\sigma &= \perp \\ &= \sup\{\perp, \perp, \perp, \ldots\} \\ &= \sup\{G f_i \mid i \in \mathbb{N}\} \end{aligned}$$

si $\sup\{f_i \mid i \in \mathbb{N}\}$ no es parcial entonces o f_i es parcial $\forall i \in \mathbb{N}$ y

$$G(\sup\{f_i \mid i \in \mathbb{N}\})\sigma = \sigma$$

pero $\sup\{G f_i \mid i \in \mathbb{N}\}\sigma = \sup\{G f_i \sigma \mid i \in \mathbb{N}\} = \perp$

todas las f_i son parciales

Pero recordemos que $\Sigma \to \Sigma'_\perp$ tiene orden llano, entonces no hay cadenas interesantes y el supremo de cada cadena está en la cadena, entonces si $\sup\{f_i \mid i \in \mathbb{N}\}$ no es parcial algún f_i $i \in \mathbb{N}$ no es parcial y

$$G(\sup\{f_i \mid i \in \mathbb{N}\})\sigma = \sigma = \sup\{\perp, \perp, \perp, \ldots, \sigma, \sigma\} = \sup\{G f_i \sigma \mid i \in \mathbb{N}\}.$$

Entonces G es continua.

(1) Considerando el lenguaje con fallas y output, de un programa para cada posible comportamiento:
   (a) cantidad finita de output y luego divergencia,
   (b) cantidad finita de output y luego falla,
   (c) cantidad finita de output y luego terminación,
   (d) cantidad infinita de output.

(a) while True if $x<0$ do $(x!;\ x:=x+1)$ else skip

Para $\sigma x = -2$ $\langle -2, -1 \rangle$

(b) while True if $x<0$ do $(x!;\ x:=x+1)$ else fail

Para $\sigma x = -2$ $\langle -2, -1, \langle \text{abort}, [\sigma | x : 0] \rangle \rangle$

(c) while $x<0$ do $(x!;\ x:=x+1)$

Para $\sigma x = -2$ $\langle -2, -1, [\sigma | x : 0] \rangle$

(d) while True do $x!$

Para $\sigma x = 2$ $\langle 2, 2, 2, \dots \rangle$

(2) Dado el programa **while** x $>$ 0 **do** !x; c, calcule su semántica denotacional, considerando los casos
   (a) $c \equiv$ **if** x $>$ 0 **then skip else fail**
   (b) $c \equiv$ **if** x $>$ 0 **then fail else skip**

(a) $[\![\text{while } x>0 \text{ do } !x; \text{if } x>0 \text{ then skip else fail}]\!]\sigma = \bigsqcup_{i=0}^{\infty} F^i \perp_{\Sigma \to \Omega}$

$$F w \sigma = \begin{cases} w_*([\![!x; \text{if } x>0 \text{ then skip else fail}]\!]\sigma) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_*([\![\text{if } x>0 \text{ then skip else fail}]\!]_* [\![!x]\!]\sigma) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_*([\![\text{if } x>0 \text{ then skip else fail}]\!] \langle \sigma x, \sigma \rangle) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_*(\text{if } \sigma x>0 \text{ then } \langle \sigma x, \sigma \rangle \text{ else } \langle \sigma x, \langle \text{abort}, \sigma \rangle \rangle) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \omega_\alpha \langle \sigma x, \sigma \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$F^0 \perp_{\Sigma \to \Omega} \sigma = \perp_{\Sigma \to \Omega}$$

$$F \perp_{\Sigma \to \Omega} \sigma = \begin{cases} (\perp_{\Sigma \to \Omega})_\alpha \langle \sigma x, \sigma \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \langle \sigma x \rangle ++ (\perp_{\Sigma \to \Omega})_\alpha \langle \sigma \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \langle \sigma x, \perp \rangle = \langle \sigma x \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$F^2 \perp_{\Sigma \to \Omega} \sigma = \begin{cases} (F \perp_{\Sigma \to \Omega})_\alpha \langle \sigma x, \sigma \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \langle \sigma x \rangle ++ (F \perp_{\Sigma \to \Omega})_\alpha \langle \sigma \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \langle \sigma x, \sigma x \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

Claramente $\bigsqcup_{i=0}^{\infty} F^i \perp_{\Sigma \to \Omega} \sigma = \begin{cases} \langle \sigma x, \sigma x, \dots \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$

(b) $[\![\text{while } x>0 \text{ do } !x ; \text{if } x>0 \text{ then fail else skip}]\!]\sigma = \bigsqcup_{i=0}^{\infty} F \perp_{\Sigma \to \Omega}$

$$F \omega \sigma = \begin{cases} \omega_\alpha([\![!x ; \text{if } x>0 \text{ then fail else skip}]\!]\sigma) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \omega_\alpha([\![\text{if } x>0 \text{ then fail else skip}]\!]_\alpha [\![!x]\!]\sigma) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \omega_\alpha([\![\text{if } x>0 \text{ then fail else skip}]\!] \langle \sigma x, \sigma \rangle) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \omega_\alpha(\text{if } \sigma x > 0 \text{ then } \langle \sigma x, \langle \text{abort}, \sigma \rangle \rangle \text{ else } \langle \sigma x ; \sigma \rangle) & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_* \, \langle \sigma x, \langle abort, \sigma \rangle \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \langle \sigma x, \langle abort, \sigma \rangle \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$$

Claramente $\bigsqcup_{i=0}^{\infty} F^i \perp \to \Omega \, \sigma = \begin{cases} \langle \sigma x, \langle abort, \sigma \rangle \rangle & \sigma x > 0 \\ \langle \sigma \rangle & \sigma x \leq 0 \end{cases}$

(3) Demostrar o refutar las siguientes equivalencias usando semántica denotacional:
- (a) $?x; ?y \ \equiv ?y; ?x$.
- (b) $?x; z := x \ \equiv ?z$.
- (c) **newvar** $x := e$ **in** $(?x; z := x) \ \equiv ?z$.

(a) $[\![?x; ?y]\!] \sigma =$

$[\![?y]\!]_* ([\![?x]\!] \sigma) =$

$[\![?y]\!]_* (\iota_{in} (\lambda n \in \mathbb{Z} \ \iota_{term} [\sigma | x : n])) =$

$\iota_{in} ([\![?y]\!]_* (\lambda n \in \mathbb{Z} \ \iota_{term} [\sigma | x : n])) =$

$\iota_{in} (\lambda n \in \mathbb{Z} \ [\![?y]\!]_* \iota_{term} [\sigma | x : n]) =$

$\iota_{in} (\lambda n \in \mathbb{Z} \ [\![?y]\!] [\sigma | x : n]) =$

$\iota_{in} (\lambda n \in \mathbb{Z} \ \iota_{in} (\lambda m \in \mathbb{Z} \ \iota_{term} [[\sigma | x : n] | y : m]))$

$[\![?y; ?x]\!] \sigma =$

$[\![?x]\!]_* ([\![?y]\!] \sigma) =$

$[\![?x]\!]_* (\iota_{in} (\lambda n \in \mathbb{Z} \ \iota_{term} [\sigma | y : n])) =$

$\iota_{in} ([\![?x]\!]_* (\lambda n \in \mathbb{Z} \ \iota_{term} [\sigma | y : n])) =$

$\iota_{in} (\lambda n \in \mathbb{Z} \ [\![?x]\!]_* \iota_{term} [\sigma | y : n]) =$

$\iota_{in} (\lambda n \in \mathbb{Z} \ \iota_{in} (\lambda m \in \mathbb{Z} \ \iota_{term} [[\sigma | y : n] | x : m]))$

No son equivalentes

$\iota_{in} f \neq \iota_{in} g \iff f \neq g$

$f_1 = \iota_{in} (\lambda m \in \mathbb{Z} \ \underbrace{\iota_{term} [[\sigma | x : n] | y : m]}_{h})$

$$g_1 = \mathrm{lin}\,(\lambda m \in \mathbb{Z}\ \ \overbrace{\mathrm{lterm}\,[\,[\sigma \mid y:1] \mid x:m]}^{h'})$$

$$\mathrm{lin}\,h \neq \mathrm{lin}\,h' \iff h \neq h'$$

$$h_2 = \mathrm{lterm}\,[\,[\sigma \mid x:1] \mid y:2]$$

$$h_2' = \mathrm{lterm}\,[\,[\sigma \mid y:1] \mid x:2]$$

(b) $[\![?x ; z := x]\!]\sigma =$

$$[\![z := x]\!]_* \left([\![?x]\!]\sigma \right) =$$

$$[\![z := x]\!]_*\, \mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \mathrm{lterm}\,[\sigma \mid x:n]) =$$

$$\mathrm{lin}\,(\,[\![z := x]\!]_* (\lambda n \in \mathbb{Z}\ \mathrm{lterm}\,[\sigma \mid x:n])) =$$

$$\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \ [\![z := x]\!]_*\, \mathrm{lterm}\,[\sigma \mid x:n]) =$$

$$\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \ \mathrm{lterm}\,[\,[\sigma \mid x:n] \mid z:\sigma x])$$

$$[\![?z]\!]\sigma =$$

$$\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \ \mathrm{lterm}\,[\sigma \mid z:n])$$

No son equivalentes

(c) $[\![\mathbf{newvar}\ x := e\ \mathbf{in}\ (?x ; z := x)]\!]\sigma =$

$$(\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger \left([\![?x ; z := x]\!]\sigma \right) =$$

$$(\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger \left([\![z := x]\!]_*\ [\![?x]\!]\sigma \right) =$$

$$(\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger \left([\![z := x]\!]_*\, \mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \mathrm{lterm}\,[\sigma \mid x:n]) \right) =$$

$$(\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger \left(\mathrm{lin}\,(\,[\![z := x]\!]_*\ (\lambda n \in \mathbb{Z}\ \mathrm{lterm}\,[\sigma \mid x:n])) \right) =$$

$$(\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger \left(\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ [\![z := x]\!]_*\ \mathrm{lterm}\,[\sigma \mid x:n]) \right) =$$

$$(\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger \left(\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \mathrm{lterm}\,[\,[\sigma \mid x:n] \mid z:\sigma x]) \right) =$$

$$\mathrm{lin}\,\left((\lambda \sigma' \in \Sigma\, [\sigma' \mid x:\sigma x])_\dagger\, (\lambda n \in \mathbb{Z}\ \mathrm{lterm}\,[\,[\sigma \mid x:n] \mid z:\sigma x]) \right) =$$

$$\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \ (\lambda \sigma' \in \Sigma .\, [\sigma' \mid x:\sigma x])_\dagger\ \mathrm{lterm}\,[\,[\sigma \mid x:n] \mid z:\sigma x]) =$$

$$\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \ \mathrm{lterm}\,[\,[\,[\sigma \mid x:n] \mid z:\sigma x] \mid x:\sigma x]) =$$

$$\mathrm{lin}\,(\lambda n \in \mathbb{Z}\ \ \mathrm{lterm}\,[\sigma \mid z:n]).$$

$$= [\![?z]\!]\sigma$$

(4) Sea c un progama que no incluya fallas, outputs, ni inputs tal que $\{\mathsf{x},\mathsf{y}\} \cap FV(c) = \emptyset$. Determine si es válida la siguiente igualdad:

$$?\mathsf{x}\,;\, c\,;\, !\mathsf{x} \equiv ?\mathsf{y}\,;\, c\,;\, !\mathsf{y}$$

$$[\![?x; c; !x]\!]\sigma =$$

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

$$[\![c; !x]\!]_* (\iota_{in} (\lambda n \in \mathbb{Z}\ \iota_{term} [\sigma | x : n])) =$$

$$[\![!x]\!]_* [\![c]\!]_* (\iota_{in} (\lambda n \in \mathbb{Z}\ \iota_{term} [\sigma | x : n])) =$$

$$[\![!x]\!]_* \iota_{in} (\lambda n \in \mathbb{Z}\ [\![c]\!]_* \iota_{term} [\sigma | x : n]))$$

No son equivalentes, ya que en los estados finales de ambos casos (si $[\![c]\!]\sigma = \perp$) se va a obtene valores distintos en x, e y.

(5) Describa mediante un diagrama de Hasse las relaciones de orden que se establecen entre los siguientes elementos de Ω:

- $\iota_{in}(\lambda n.\ \iota_{out}(n, \perp))$
- $\iota_{out}(0, \perp)$,
- $\iota_{in}(\lambda n.\ \ \perp)$,
- $\iota_{in}(f)$, donde $f\,n = \begin{cases} \perp & \text{si } n < 0 \\ \iota_{out}(n, \perp) & \text{caso contrario} \end{cases}$

$$\begin{array}{ccc} & \iota_{in} (\lambda n.\ \iota_{out} (n, \perp)) & \\ & | & \\ \iota_{out} (0, \perp) & \iota_{in}(f) & f_n = \begin{cases} \perp \\ \iota_{out} (n, \perp) \end{cases} \\ & | & \\ & \iota_{in} (\lambda n.\ \perp) & \end{array}$$

(6) Dé un programa, y justifique su elección, cuya semántica sea el supremo de la cadena:

$$w_0 = \perp, \qquad w_{i+1} = \iota_{in}(\lambda n.\ \iota_{out}(n, w_i))$$

$$[\![\text{while true do } ?x; !x]\!]\sigma = \bigsqcup_{i=0}^{\infty} F \perp_{\Sigma \to \Omega} \sigma$$

$$F\,w\,\sigma = \begin{cases} w_* \llbracket ?x;\,!x \rrbracket \sigma & \llbracket true \rrbracket \sigma \\ \iota_{term}\, \sigma & \llbracket false \rrbracket \sigma \end{cases}$$

$$= w_* (\llbracket !x \rrbracket_* (\iota_{in} (\lambda n.\ \iota_{term} [\sigma | x : n])))$$

$$= w_* \iota_{in} (\lambda n.\ \llbracket !x \rrbracket_* \iota_{term} [\sigma | x : n])$$

$$= w_* \iota_{in} (\lambda n\ \iota_{out} ([\sigma | x : n] x,\ \iota_{term} [\sigma | x : n])$$

$$= w_* \iota_{in} (\lambda n\ \iota_{out} (n,\ \iota_{term} [\sigma | x : n]))$$

$$F^0 \perp_{\Sigma \to \Omega}\, \sigma = \perp_{\Sigma \to \Omega}$$

$$F \perp_{\Sigma \to \Omega}\, \sigma = (\perp_{\Sigma \to \Omega})_* \iota_{in} (\lambda n\ \iota_{out} (n, \iota_{term} [\sigma | x : n]))$$

$$= \iota_{in} (\lambda n\ (\perp_{\Sigma \to \Omega})_* \iota_{out} (n, \iota_{term} [\sigma | x : n]))$$

$$= \iota_{in} (\lambda n\ \iota_{out} (n, (\perp_{\Sigma \to \Omega})_* \iota_{term} [\sigma | x : n]))$$

$$= \iota_{in} (\lambda n\ \iota_{out} (n, \perp_\Omega))$$

$$F^2 \perp_{\Sigma \to \Omega}\, \sigma = (F \perp_{\Sigma \to \Omega})_* \iota_{in} (\lambda n\ \iota_{out} (n, \iota_{term} [\sigma | x : n]))$$

$$= \iota_{in} (\lambda n\ (F \perp_{\Sigma \to \Omega})_* \iota_{out} (n, \iota_{term} [\sigma | x : n]))$$

$$= \iota_{in} (\lambda n\ \iota_{out} (n, (F \perp_{\Sigma \to \Omega})_* \iota_{term} [\sigma | x : n]))$$

$$= \iota_{in} (\lambda n\ \iota_{out} (n, \underbrace{\iota_{in} (\lambda n\ \iota_{out} (n, \perp_\Omega))}_{(F \perp_{\Sigma \to \Omega})_*}))$$

si $F^i \perp_{\Sigma \to \Omega} = w_i$

$w_0 = \perp_{\Sigma \to \Omega}$

$w_{i+1} = \iota_{in} (\lambda n\ \iota_{out} (n, w_i))$

(7) Considere los programas de la forma **while true do** $(?x; c)$. La cadena $F^i \perp \sigma$ de la semántica del **while**, ¿será siempre una cadena interesante en Ω? Justifique su respuesta.

Falso, por ejemplo si $\llbracket c \rrbracket \sigma = \iota_{abort}\, \sigma$

$$\llbracket \text{while true do } (?x; c) \rrbracket \sigma = \bigsqcup_{i=0}^{\infty} F^i \perp_{\Sigma \to \Omega}\, \sigma$$

$$F\,w\,\sigma = \begin{cases} w_* \llbracket ?x; c \rrbracket \sigma & \llbracket true \rrbracket \sigma \\ \iota_{term}\, \sigma & \llbracket false \rrbracket \sigma \end{cases}$$

$$= w_* (\llbracket c \rrbracket_* \llbracket ?x \rrbracket \sigma)$$

$= w_\alpha \left([\![c]\!]_* \, \iota_{in}(\lambda n.\ \iota_{term}[\sigma|x:n]) \right)$

$= w_\alpha \ \iota_{in}(\lambda n\ [\![c]\!]_* \iota_{term}[\sigma|x:n]))$

$= w_\alpha \ \iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:n])$

$= \iota_{in}(\lambda n\ w_\alpha \iota_{abort}[\sigma|x:n])$

$= \iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:n])$

Ni siquiera depende de w entonces la cadena es igual a

$\perp_{\Sigma\to\Omega} \leq \iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:n]) \leq \iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:n]) \leq \dots$

(8) Dado el programa $P \equiv$

newvar x := x + 1 **in**
while x > 0 **do** ?x; **if** y > 0 **then fail else** !x

(a) Calcular la semántica denotacional de P, en un estado σ tal que $\sigma y > 0$.
(b) Considere el caso $\sigma y \leq 0$. Calcule $F\perp$ y $F^2\perp$. Puede dar una expresión general para $\mathbf{Y}_{\Sigma\to\Sigma_\perp} F$.

(a) $[\![\text{newvar } x := x+1 \text{ in while } x>0 \text{ do } ?x; \text{ if } y>0 \text{ then fail else } !x]\!]\sigma$

$= (\lambda\sigma'.\ [\sigma'|x:\sigma x])_\dagger ([\![\text{while } x>0 \text{ do } ?x; \text{if } y>0 \text{ then fail else } !x]\!]_* [\sigma|x:\sigma x+1])$

$[\![\text{while } x>0 \text{ do } ?x; \text{if } y>0 \text{ then fail else } !x]\!]\sigma = \bigsqcup_{i=0}^{\infty} F^i \perp_{\Sigma\to\Omega}\sigma$

$$F\,w\,\sigma = \begin{cases} w_* [\![?x; \text{if } y>0 \text{ then fail else } !x]\!] & \sigma x>0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

Veamos el caso en que $\sigma y > 0$

$$F w \sigma = \begin{cases} w_* ([\![\text{fail}]\!]_* \iota_{in}(\lambda n.\ \iota_{term}[\sigma|x:n]) & \sigma x>0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_* \ \iota_{in}(\lambda n\ [\![\text{fail}]\!]_* \iota_{term}[\sigma|x:n]) & \sigma x>0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_* \ \iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:n]) & \sigma x>0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ w_* \iota_{abort}[\sigma|x:n]) & \sigma x>0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:n]) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

si $\sigma x + 1 > 0$

$(\lambda\sigma'.\ [\sigma'|x:\sigma x])_\dagger (⟦\text{while } x>0 \text{ do } ?x; \text{if } y>0 \text{ then fail else } !x⟧_* [\sigma|x:\sigma x+1]) =$

$(\lambda\sigma'.\ [\sigma'|x:\sigma x])_\dagger (\iota_{in}(\lambda n\ \iota_{abort}[\sigma|x:\sigma x+1])) =$

$\iota_{in}(\lambda n\ (\lambda\sigma'.\ [\sigma'|x:\sigma x])_\dagger\ \iota_{abort}\ [[\sigma|x:\sigma x+1]|x:n]) =$

$\iota_{in}(\lambda n\ \iota_{abort}\ [[[\sigma|x:\sigma x+1]|x:n]|x:\sigma x])$

$\iota_{in}(\lambda n\ \iota_{abort}\ \sigma)$

si $\sigma x + 1 < 0$

$(\lambda\sigma'.\ [\sigma'|x:\sigma x])_\dagger (⟦\text{while } x>0 \text{ do } ?x; \text{if } y>0 \text{ then fail else } !x⟧_* [\sigma|x:\sigma x+1]) =$

$(\lambda\sigma'.\ [\sigma'|x:\sigma x])_\dagger (\iota_{term}[\sigma|x:\sigma x+1]) =$

$\iota_{term}\ [[\sigma|x:\sigma x+1]|x:\sigma x] =$

$\iota_{term}\ \sigma$

(b) $$F\,w\,\sigma = \begin{cases} w_* ⟦?x; \text{if } y>0 \text{ then fail else } !x⟧ & \sigma x > 0 \\ \iota_{term}\ \sigma & \sigma x \leq 0 \end{cases}$$

Veamos el caso en que $\sigma y \leq 0$

$$F w \sigma = \begin{cases} w_* (⟦!x⟧_*\ \iota_{in}(\lambda n.\ \iota_{term}[\sigma|x:n]) & \sigma x > 0 \\ \iota_{term}\ \sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_*\ \iota_{in}(\lambda n\ ⟦!x⟧\ \iota_{term}[\sigma|x:n]) & \sigma x > 0 \\ \iota_{term}\ \sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_*\ \iota_{in}(\lambda n\ \iota_{out}([\sigma|x:n]x,\ \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\ \sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} w_*\ \iota_{in}(\lambda n\ \iota_{out}(n,\ \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\ \sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ w_*\ \iota_{out}(n,\ \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\ \sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \omega_* \iota_{out}(n, \iota_{term}(\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$F^0 \bot_{\Sigma\to\Omega}\,\sigma = \bot_{\Sigma\to\Omega}$$

$$F \bot_{\Sigma\to\Omega}\,\sigma = \begin{cases} \iota_{in}(\lambda n\ (\bot_{\Sigma\to\Omega})_* \iota_{out}(n, \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \iota_{out}(n, (\bot_{\Sigma\to\Omega})_* \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \iota_{out}(n, \bot_\Omega)) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$F^2 \bot_{\Sigma\to\Omega}\,\sigma = \begin{cases} \iota_{in}(\lambda n\ (F\bot_{\Sigma\to\Omega})_* \iota_{out}(n, \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \iota_{out}(n, (F\bot_{\Sigma\to\Omega})_* \iota_{term}[\sigma|x:n])) & \sigma x > 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \iota_{out}(n, \iota_{in}(\lambda n'\ \iota_{out}(n', \bot_\Omega)))) & \sigma x > 0 \wedge [\sigma|x:n]x > 0 \\ \iota_{in}(\lambda n\ \iota_{out}(n, \iota_{term}[\sigma|x:n])) & \sigma x > 0 \wedge [\sigma|x:n]x \leq 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

$$= \begin{cases} \iota_{in}(\lambda n\ \iota_{out}(n, \iota_{in}(\lambda n'\ \iota_{out}(n', \bot_\Omega)))) & \sigma x > 0 \wedge n > 0 \\ \iota_{in}(\lambda n\ \iota_{out}(n, \iota_{term}[\sigma|x:n])) & \sigma x > 0 \wedge n \leq 0 \\ \iota_{term}\,\sigma & \sigma x \leq 0 \end{cases}$$

No se puede dar una expresión general, ya que el comportamiento del programa va a depender del valor que se le asigna a x en cada iteración.

(9) Demostrar o refutar las siguientes equivalencias usando semántica denotacional
  (a) ?x; **while** b **do** !x; ?x **od**; !x $\equiv$ **while** b **do** ?x; !x **od**.
  (b) ?x; **while** b **do** !x; ?x **od**; !x $\equiv$?x; !x; **while** b **do** ?x; !x **od**.

(a) $[\![?x; \text{while } b \text{ do } !x; ?x \text{ od}; !x]\!]\sigma$

$= [\![\text{while } b \text{ do } !x; ?x \text{ od}; !x]\!]_* ([\![?x]\!]\sigma)$.

$$= [\![\text{while } b \text{ do } !x; ?x \text{ od}; !x]\!]_* (\iota_{in}(\lambda n.\ \iota_{term}[\sigma|x:n]))$$
$$= \iota_{in}(\lambda n.\ [\![\text{while } b \text{ do } !x; ?x \text{ od}; !x]\!]_* \iota_{term}[\sigma|x:n])$$
$$= \iota_{in}(\lambda n.\ [\![!x]\!]_* ([\![\text{while } b \text{ do } !x; ?x \text{ od}]\!] [\sigma|x:n]))$$

$$[\![\text{while } b \text{ do } !x; ?x \text{ od}]\!]\sigma = \bigsqcup_{i=0}^{\infty} F^i \bot_{\Sigma\to\Omega}\sigma$$

$$F w \sigma = \begin{cases} w_* [\![!x; ?x]\!]\sigma & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$F w \sigma = \begin{cases} w_* ([\![?x]\!]_* \iota_{out}(\sigma x, \iota_{term}\,\sigma)) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$= \begin{cases} w_* (\iota_{out}(\sigma x, [\![?x]\!]_* \iota_{term}\,\sigma)) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$= \begin{cases} w_* (\iota_{out}(\sigma x, \iota_{in}(\lambda n.\ \iota_{term}[\sigma|x:n]))) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$= \begin{cases} \iota_{out}(\sigma x, \iota_{in}(\lambda n\ w_*\ \iota_{term}[\sigma|x:n])) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$F^0 \bot_{\Sigma\to\Omega}\sigma = \bot_{\Sigma\to\Omega}$$

$$F \bot_{\Sigma\to\Omega}\sigma = \begin{cases} \iota_{out}(\sigma x, \iota_{in}(\lambda n (\bot_{\Sigma\to\Omega})_* \iota_{term}[\sigma|x:n])) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$= \begin{cases} \iota_{out}(\sigma x, \iota_{in}(\lambda n\ \bot_\Omega)) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$F^2 \bot_{\Sigma\to\Omega}\sigma = \begin{cases} \iota_{out}(\sigma x, \iota_{in}(\lambda n\ (F \bot_{\Sigma\to\Omega})_* \iota_{term}[\sigma|x:n])) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

$$= \begin{cases} \iota_{out}(\sigma x, \iota_{in}(\lambda n\ \iota_{out}(\sigma x, \iota_{in}(\lambda n \bot_\Omega)))) & [\![b]\!]\sigma \\ \iota_{term}\,\sigma & \neg[\![b]\!]\sigma \end{cases}$$

No es fácil encontrar el supremo pero supongamos $b = \text{fail}$

$$[\![?x ; \text{while } b \text{ do } !x ; ?x \text{ od} ; !x]\!] \sigma$$
$$= \iota_{in}(\lambda n.\ [\![!x]\!]_* ([\![\text{while } b \text{ do } !x ; ?x \text{ od}]\!] [\sigma | x : n]))$$
$$= \iota_{in}(\lambda n.\ [\![!x]\!]_* \ \iota_{term} [\sigma | x : n])$$
$$= \iota_{in}(\lambda n.\ \iota_{out}([\sigma | x : n] x , \iota_{term} [\sigma | x : n]))$$
$$= \iota_{in}(\lambda n.\ \iota_{out}(n , \iota_{term} [\sigma | x : n]))$$

mientras que
$$[\![\text{while } b \text{ do } !x ; ?x \text{ od}]\!] \sigma$$
$$= \iota_{term}\ \sigma$$

(b) Son equivalentes