

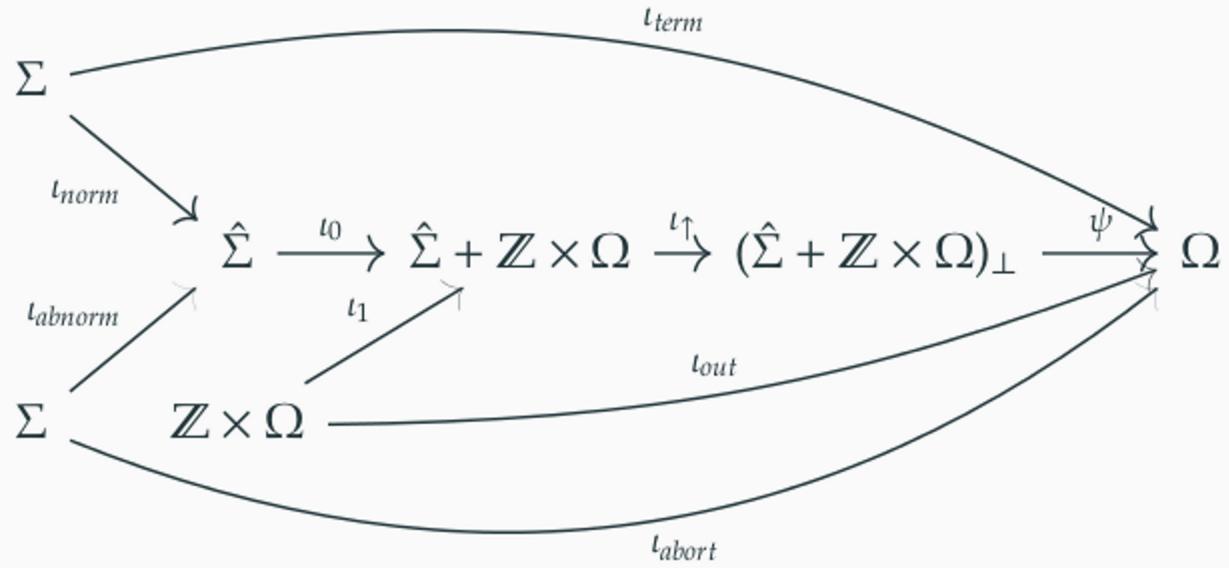
# Definiciones

viernes, 19 de abril de 2024 9:36

## Imperativo con input:

$\langle \text{comm} \rangle ::= \dots | ! \langle \text{intexp} \rangle$

$$\Omega = \hat{\Sigma} (+ \mathbb{Z} \times \Omega_{\perp})$$



$\square_* : (\Sigma \rightarrow \Omega) \rightarrow \Omega \rightarrow \Omega$

$$f_* \omega = \begin{cases} \omega = \iota_{term}(\sigma) & \rightarrow f \sigma \\ \omega = \iota_{out}(n, \sigma) & \rightarrow \iota_{out}(f_*(\sigma)) \\ \text{si no} & \rightarrow \omega \end{cases}$$

$\square_+ : (\Sigma \rightarrow \Sigma) \rightarrow \Omega \rightarrow \Omega$

$$f_\dagger \omega = \begin{cases} \omega = \iota_{term}(\sigma) & \rightarrow \quad \iota_{term} f((\sigma)) \\ \omega = \iota_{out}(n, \sigma) & \rightarrow \quad \iota_{out} n, f_*(\sigma)) \\ \omega = \iota_{abort}(\sigma) & \rightarrow \quad \iota_{abort} f((\sigma)) \\ \omega = \iota_{out}(n, \sigma) & \rightarrow \quad \iota_{out} (n, f_\dagger(\sigma)) \\ \text{si no} & \rightarrow \quad \perp \end{cases}$$

$\boxed{\text{ }} \langle \text{comm} \rangle \rightarrow \Sigma \rightarrow \Omega$

$! \boxed{p} = \iota_{out}(\sigma x, \sigma)$

**Agregando input:**

$\langle \text{comm} \rangle ::= \dots \mid ? \langle \text{var} \rangle$

$$\Omega = \hat{\Sigma} (+ \ \mathbb{Z} \times \Omega + \mathbb{Z} \rightarrow \Omega_\perp)$$

\* y  $\dagger$  se modifican adecuadamente

$? \boxed{p} = \iota_{in}(\lambda n . [\sigma \mid x: n])$

1)

jueves, 18 de abril de 2024 15:08

## Ejercicios.

(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 do skip

b)

abort

c)

skip

d)

while true do ! 0

2) 

viernes, 19 de abril de 2024 9:33

(2) Dado el programa **while**  $x > 0$  **do**  $\lambda x;c$ , calcule su semántica denotacional, considerando los casos

- (a)  $c \equiv \text{if } x > 0 \text{ then skip else fail}$
- (b)  $c \equiv \text{if } x > 0 \text{ then fail else skip}$

3)

viernes, 19 de abril de 2024 9:33

(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)

Falso cuando  $x \neq y$ :

Tenemos:

$$\begin{aligned} ?[\![x; ?y]\!] &=?[\![x]\!][\![?y]\!]) \\ &=?[\![x]\!](\lambda n . \iota_{term}[\sigma | x: n])) \\ &= \iota_{in}(\lambda n ?[\![x]\!]\iota_{term}[\sigma | x: n])) \\ &= \iota_{in}(\lambda n ?[\![x]\! | x: n]) \\ &= \iota_{in}(\lambda n . \iota_{in}(\lambda m . \iota_{term}[\sigma | x: n, y: m])) \end{aligned}$$

Analogamente:

$$?[\![y; ?x]\!] = \iota_{in}(\lambda n . \iota_{in}(\lambda m . \iota_{term}[\sigma | x: m, y: n]))$$

b)

Falso cuando  $x \neq z$

Tenemos:

$$\begin{aligned} ?[\![x; z := x]\!] &=?[\![z := x]\!][\![?x]\!]\sigma) \\ &=?[\![z := x]\!](\lambda n . \iota_{term}[\sigma | x: n])) \\ &= \iota_{in}(\lambda n . \iota_{term}[\![z := x]\! | x: n])) \\ &= \iota_{in}(\lambda n . \iota_{term}[\sigma | x: n, z: [\sigma | x: n] x]) \\ &= \iota_{in}(\lambda n . \iota_{term}[\sigma | x: n, z: n]) \end{aligned}$$

Y:

$$\begin{aligned} ?[\![?z]\!]\sigma &= \iota_{in}(\lambda n . \iota_{term}[\sigma | z: n]) \end{aligned}$$

c)

Verdadero, demostración:

$$\begin{aligned} n[\![\text{newvar } x := e \text{ in } ?x; z := x]\!] &= \text{rest}_{x, \sigma} x_+ (\ell[\![x; z := x]\!]) \\ &= \text{rest}_{x, \sigma} x_+ (\ell[\![x; z := x]\!]) \\ &= \text{rest}_{x, \sigma} x_+ (\ell[\!=\! x]\!)\! (\ell[\![x]\!]) \\ &= \text{rest}_{x, \sigma} x_+ (z[\!:=\! x]_n (\lambda n . \iota_{term} [\sigma \mid x : n])) \\ &= \text{rest}_{x, \sigma} x_+ (\lambda n z[\!:=\! x]_n (\lambda n . \iota_{term} [\sigma \mid x : n])) \\ &= \text{rest}_{x, \sigma} x_+ (\lambda n . \iota_{term} [\sigma \mid x : n, z : n]) \\ &= \iota_{in} (\lambda n . \iota_{term} [\text{rest}_{x, \sigma} x [\sigma \mid x : n, z : n]]) \\ &= \iota_{in} (\lambda n . \iota_{term} [\sigma \mid z : n]) \\ &= [\![?z]\!] \sigma \end{aligned}$$

4)

viernes, 19 de abril de 2024 10:58

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

$$?x; c; !x \equiv ?y; c; !y$$

Falso, por ejemplo: cuando

$c = \text{skip}$

5)

viernes, 19 de abril de 2024 10:58

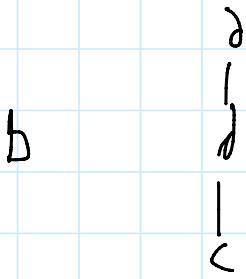
- (5) Describa mediante un diagrama de Hasse las relaciones de orden que se establecen entre los siguientes elementos de  $\Omega$ :

a •  $\iota_{in}(\lambda n. \iota_{out}(n, \perp))$

b •  $\iota_{out}(0, \perp)$ ,

c •  $\iota_{in}(\lambda n. \perp)$ ,

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



6)

viernes, 19 de abril de 2024

12:00

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

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

while true ? x; ! x

7)

viernes, 19 de abril de 2024

12:12

- (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  $\Omega$ ? Justifique su respuesta.

No es siempre interesante, por ejemplo cuando:

$c = \text{fail}$