Lenguajes y compiladores
1er parcial - 2023/05/03

1. La siguiente gramática abstracta corresponde a un lenguaje para dar órdenes a un robot.

$\langle ord \rangle$ ::= **mover** $\langle int \rangle$	ver ejemplo
| **girar**	intercambia la dirección actual
| **si pos** = $\langle coord \rangle$ **hacer** $\langle ord \rangle$	ejecuta la orden si la posición actual es la dada
| **si dir** = $\langle dir \rangle$ **hacer** $\langle ord \rangle$	ejecuta la orden si la dirección actual es la dada
| $\langle ord \rangle$; $\langle ord \rangle$	ejecuta la primera orden y luego la segunda
$\langle int \rangle$ ::= ... | −2 | −1 | 0 | 1 | 2 | ...	
$\langle dir \rangle$ ::= **EO** | **NS**	
$\langle coord \rangle$ ::= ($\langle int \rangle$, $\langle int \rangle$)	

Sea $D = \{NS, EO\}$ el conjunto de direcciones y sea $\Sigma = (\mathbb{Z} \times \mathbb{Z}) \times D$ el conjunto de estados. Escribí las ecuaciones semánticas con el siguente tipo: $[\![\_]\!] \colon \langle ord \rangle \to \Sigma \to \Sigma$. Por ejemplo,

$$[\![\textbf{mover } k]\!]((x,y),d) = \begin{cases} ((x+k,y),d) & \text{si } d = EO \\ ((x,y+k),d) & \text{si } d = NS \end{cases}$$

$$[\![\text{mover } k]\!]((x,y),d) = \begin{cases} ((x+k,y),d) & \text{si } d = EO \\ ((x,y+k),d) & \text{si } d = NS \end{cases}$$

$$[\![\text{girar}]\!]((x,y),d) = \begin{cases} ((x,y),NS) & \text{si } d = EO \\ ((x,y),EO) & \text{si } d = NS \end{cases}$$

$$[\![\text{si pos} = c \text{ hacer } p]\!]((x,y),d) = \begin{cases} [\![p]\!]((x,y),d) & \text{si } c = (x,y) \\ ((x,y),d) & \text{si no} \end{cases}$$

$$[\![\text{si dir} = d \text{ hacer } p]\!]((x,y),d') = \begin{cases} [\![p]\!]((x,y),d') & \text{si } d = d' \\ ((x,y),d') & \text{si no} \end{cases}$$

$$[\![p;q]\!]\sigma = [\![q]\!]([\![p]\!]\sigma)$$

2. Considerá la siguiente ecuación recursiva:

$$f(x) = \begin{cases} 0 & \text{si } x = 0 \\ 8 - f(x-2) & \text{si } x \neq 0 \end{cases}$$

Sea $F \colon (\mathbb{Z} \to \mathbb{Z}_\perp) \to (\mathbb{Z} \to \mathbb{Z}_\perp)$ el funcional asociado a esa ecuación. ¿Existe $x \in \mathbb{Z}$ tal que $F^3(\perp)(x) = 10$?

$$F\,f\,x = \begin{cases} 0 & \text{si } x = 0 \\ 8 - f(x-2) & \text{si } x \neq 0 \end{cases}$$

$F^0 \perp x = \perp$

$$F \perp x = \begin{cases} 0 & si\ x = 0 \\ 8 - \perp(x-2) & si\ x \neq 0 \end{cases}$$

$$F \perp x = \begin{cases} 0 & si\ x = 0 \\ \perp & si\ x \neq 0 \end{cases}$$

$$F^2 \perp x = \begin{cases} 0 & si\ x = 0 \\ 8 - F \perp x(x-2) & si\ x \neq 0 \end{cases}$$

$$F^2 \perp x = \begin{cases} 0 & si\ x = 0 \\ 8 - 0 & si\ x \neq 0 \wedge x-2 = 0 \\ \perp & si\ x \neq 0 \wedge x-2 \neq 0 \end{cases}$$

$$F^2 \perp x = \begin{cases} 0 & si\ x = 0 \\ 8 & si\ x = 2 \\ \perp & si\ x \neq 0 \wedge x \neq 2 \end{cases}$$

$$F^3 \perp x = \begin{cases} 0 & si\ x = 0 \\ 8 - F^2 \perp x(x-2) & si\ x \neq 0 \end{cases}$$

$$F^3 \perp x = \begin{cases} 0 & x = 0 \\ 8 & x \neq 0 \wedge x-2 = 0 \\ 0 & x \neq 0 \wedge x-2 = 2 \\ \perp & x \neq 0 \wedge x-2 \neq 0 \wedge x-2 \neq 2 \end{cases}$$

$$F^3 \perp x = \begin{cases} 0 & x = 0 \vee x = 4 \\ 8 & x = 2 \\ \perp & x \notin \{0, 2, 4\} \end{cases}$$

No existe x tal que $F^3 \perp x = 10$.

3. Decidí si cada una de las siguientes afirmaciones es verdadera o falsa. Justificá tu respuesta.

(a) En cualquier predominio infinito siempre hay cadenas interesantes.

(b) Sea $f \colon P \to P'$ una función continua entre los predominios P y P', entonces $f(\sqcup_i x_i) \leqslant \sqcup_i (f(x_i))$, asumí que x_i es una cadena interesante.

(a) Falso $\mathbb{N}$ con el orden discreto es un predominio infinito donde todos los elementos son maximales.

(b) Verdadero por definición de continuidad $f(\bigsqcup_i x_i) = \bigsqcup_i (f(x_i))$

4. Considerá el lenguaje imperativo simple con fallas. Sea c el programa siguiente

$$\textbf{while } x \neq 0 \textbf{ do if } x > 0 \textbf{ then } d := 1 + d;\ x := x - 1 \textbf{ else fail}$$

(a) Escribí de la forma más sencilla posible la ecuación para $F(f)(\sigma)$ donde F es el funcional asociado al ciclo de ese programa.

(b) Proponé un estado σ (dando los valores de x y d) tal que $F^1(\perp)(\sigma) \neq \perp$.

(a)

$$F\,f\,\sigma = \begin{cases} f_* [\![\text{if } x>0 \text{ then } d:=1+d;\ x:=x-1 \text{ else fail}]\!]\sigma & \text{si } \sigma x \neq 0 \\ \sigma & \sigma x = 0 \end{cases}$$

$$F\,f\,\sigma = \begin{cases} f_* [\![d:=1+d;\ x:=x-1]\!]\sigma & \text{si } \sigma x \neq 0 \wedge \sigma x > 0 \\ f_* [\![\text{fail}]\!]\sigma & \text{si } \sigma x \neq 0 \wedge \sigma x \leq 0 \\ \sigma & \sigma x = 0 \end{cases}$$

$$F\,f\,\sigma = \begin{cases} f\,[\sigma | d = 1+\sigma d \,|\, x = \sigma x - 1] & \text{si } \sigma x \neq 0 \wedge \sigma x > 0 \\ \langle \text{abort}, \sigma \rangle & \text{si } \sigma x \neq 0 \wedge \sigma x \leq 0 \\ \sigma & \text{si } \sigma x = 0 \end{cases}$$

(b) $F^0\, \perp_{\Sigma \to \Sigma'_\perp}\, \sigma = \perp_{\Sigma \to \Sigma'_\perp}$

$$F\,\perp_{\Sigma \to \Sigma'_\perp}\,\sigma = \begin{cases} \perp_{\Sigma \to \Sigma'_\perp} [\sigma | d = 1+\sigma d \,|\, x = \sigma x - 1] & \text{si } \sigma x \neq 0 \wedge \sigma x > 0 \\ \langle \text{abort}, \sigma \rangle & \text{si } \sigma x \neq 0 \wedge \sigma x \leq 0 \\ \sigma & \text{si } \sigma x = 0 \end{cases}$$

$$F\,\perp_{\Sigma \to \Sigma'_\perp}\,\sigma = \begin{cases} \perp & \text{si } \sigma x > 0 \\ \langle \text{abort}, \sigma \rangle & \text{si } \sigma x < 0 \\ \sigma & \text{si } \sigma x = 0 \end{cases}$$

$\sigma x = 0,\ \sigma d = 0.$

5. Decidí si las siguientes equivalencias son correctas. Si no lo son proponé contraejemplos concretos. Si lo son, hacé la demostración.

(a) **catchin** c **with** (**fail** ; c') $\equiv c; c'$.

(b) **catchin** (c; **fail**) **with** $c' \equiv$ **catchin** c **with** c'.

Recordá que en **catchin** c **with** c' se ejecuta c y si se produce una falla, entonces se ejecuta c' en el estado donde se produjo la falla.

(a) falso sean $c = [\![x := 0]\!]$ $c' = [\![x := 1]\!]$

$[\![\text{catchin } c \text{ with } (\text{fail}; c')]\!]\sigma$

$= [\![\text{fail}; c']\!]_+ ([\![c]\!]\sigma)$

$= [\![\text{fail}; c']\!]_+ [\sigma | x : 0]$

$= [\sigma | x : 0]$

$[\![c; c']\!]\sigma$

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

$= [\![x := 1]\!]_* ([\![x := 0]\!]\sigma)$

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

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

$= [\sigma | x : 1]$

(b) Falso sea $c = \text{skip}$ $c' = x := x+1$

$[\![\text{catching } (c; \text{fail}) \text{ with } c']\!]\sigma$

$= [\![c']\!]_+ ([\![c; \text{fail}]\!]\sigma)$

$= [\![c']\!]_+ ([\![\text{fail}]\!]_* [\![c]\!]\sigma)$

$= [\![x := x+1]\!]_+ ([\![\text{fail}]\!]_* [\![\text{skip}]\!]\sigma)$

$= [\![x := x+1]\!]_+ ([\![\text{fail}]\!]\sigma)$

$= [\![x := x+1]\!]_+ \langle \text{abort}, \sigma \rangle$

$= [\sigma | x : \sigma x + 1]$

$[\![\text{catching } c \text{ with } c']\!]\sigma$

$= [\![\text{catching skip with } x := x+1]\!]\sigma$

$$= [\![x := x+1]\!]_+ \left([\![\mathrm{skip}]\!] \sigma \right)$$

$$= [\![x := x+1]\!]_+ \, \sigma$$

$$= \sigma .$$