LENGUAJES Y COMPILADORES

Repaso. La clase pasada presentamos el lenguaje imperativo simple y dimos las ecuaciones para casi todas las construcciones del lenguaje. En efecto, teníamos que

$$\begin{array}{rcl} [\![\ ]\!] & \in & \langle\text{intexp}\rangle \to \Sigma \to \mathbb{Z} \\ [\![\ ]\!] & \in & \langle\text{boolexp}\rangle \to \Sigma \to \mathbb{B} \\ [\![\ ]\!] & \in & \langle\text{comm}\rangle \to \Sigma \to \Sigma_\perp \end{array}$$

y las ecuaciones para las expresiones enteras y booleanas son idénticas a las vistas para la lógica de predicados, salvo que ahora se omiten los casos de los cuantificadores por no formar parte del lenguaje imperativo simple.

También alcanzamos a presentar las ecuaciones de los comandos:

$$\begin{array}{rcl} [\![\ ]\!] & \in & \langle\text{comm}\rangle \to \Sigma \to \Sigma_\perp \\ [\![\mathbf{skip}]\!]\sigma & = & \sigma \\ [\![v := e]\!]\sigma & = & [\sigma|v : [\![e]\!]\sigma] \\ [\![\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1]\!]\sigma & = & \begin{cases} [\![c_0]\!]\sigma & \text{si } [\![b]\!]\sigma \\ [\![c_1]\!]\sigma & \text{si no} \end{cases} \\ [\![c_0; c_1]\!]\sigma & = & [\![c_1]\!]_{\perp\!\!\!\perp}([\![c_0]\!]\sigma) \\ [\![\mathbf{newvar}\ \text{v:= e}\ \mathbf{in}\ \text{c}]\!]\sigma & = & (\lambda\sigma' \in \Sigma.\ [\sigma'|v : \sigma\ v])_{\perp\!\!\!\perp}([\![c]\!][\sigma|v : [\![e]\!]\sigma]) \end{array}$$

donde si $f \in P \to D$, donde P es un predominio y D un dominio, entonces $f_{\perp\!\!\!\perp}$ denota la única extensión estricta de f a $P_\perp$:

$$f_{\perp\!\!\!\perp}\ x = \begin{cases} \perp & \text{si } x = \perp \\ f\ x & \text{si no} \end{cases}$$

La ecuación del while. Intuitivamente, podríamos escribir

$$\begin{array}{rcl} [\![\mathbf{while}\ b\ \mathbf{do}\ c]\!]\sigma & = & [\![\mathbf{if}\ b\ \mathbf{then}\ c; \mathbf{while}\ b\ \mathbf{do}\ c\ \mathbf{else}\ \mathbf{skip}]\!]\sigma \\ & = & \begin{cases} [\![c; \mathbf{while}\ b\ \mathbf{do}\ c]\!]\sigma & \text{si } [\![b]\!]\sigma \\ [\![\mathbf{skip}]\!]\sigma & \text{si no} \end{cases} \\ & = & \begin{cases} [\![\mathbf{while}\ b\ \mathbf{do}\ c]\!]_{\perp\!\!\!\perp}([\![c]\!]\sigma) & \text{si } [\![b]\!]\sigma \\ \sigma & \text{si no} \end{cases} \end{array}$$

La ecuación resultante

$$(1) \qquad [\![\mathbf{while}\ b\ \mathbf{do}\ c]\!]\sigma = \begin{cases} [\![\mathbf{while}\ b\ \mathbf{do}\ c]\!]_{\perp\!\!\!\perp}([\![c]\!]\sigma) & \text{si } [\![b]\!]\sigma \\ \sigma & \text{si no} \end{cases}$$

no es dirigida por sintaxis. Esto significa que no sabemos si tiene solución y en ese caso, si es única.

Pero ya estuvimos en esta situación. Definamos

$$F\ w\ \sigma = \begin{cases} w_{\perp\!\!\!\perp}([\![c]\!]\sigma) & \text{si } [\![b]\!]\sigma \\ \sigma & \text{si no} \end{cases}$$

con esta definición, la ecuación (1) queda

$$[\![\textbf{while } b \textbf{ do } c]\!]\sigma = F \ [\![\textbf{while } b \textbf{ do } c]\!] \ \sigma$$

Tratándose de una igualdad entre funciones (comprobar que $[\![\textbf{while } b \textbf{ do } c]\!] \in \Sigma \to \Sigma_{\perp\!\!\!\perp}$ y $F \in (\Sigma \to \Sigma_{\perp\!\!\!\perp}) \to (\Sigma \to \Sigma_{\perp\!\!\!\perp})$ y por ello $F \ [\![\textbf{while } b \textbf{ do } c]\!] \in \Sigma \to \Sigma_{\perp\!\!\!\perp}$ la ecuación también puede escribirse

$$[\![\textbf{while } b \textbf{ do } c]\!] = F \ [\![\textbf{while } b \textbf{ do } c]\!]$$

que simplemente dice que $[\![\textbf{while } b \textbf{ do } c]\!]$ es punto fijo de F. Recordemos entonces el teorema del menor punto fijo.

Teorema. Sea D un dominio, y $F \in D \to D$ continua. Entonces $\sup(F^i \ \perp)$ existe y es el menor punto fijo de F.

Recordemos que nos interesa el menor punto fijo de F porque otros puntos fijos contienen información que no se encuentra en la ecuación a resolver.

Sabemos que $F \in (\Sigma \to \Sigma_\perp) \to (\Sigma \to \Sigma_\perp)$, es decir $F \in D \to D$ con $D = \Sigma \to \Sigma_\perp$, que es un dominio porque $\Sigma_\perp$ lo es.

Para poder utilizar el teorema, deberíamos asegurarnos de que F es continua. En caso de serlo, la semántica de **while** b **do** c será

$$[\![\textbf{while } b \textbf{ do } c]\!] = \bigsqcup_{i=0}^{\infty} F^i \ \perp_{\Sigma \to \Sigma_\perp}$$

para

$$F \ w \ \sigma = \begin{cases} w_{\perp\!\!\!\perp}([\![c]\!]\sigma) & \text{si } [\![b]\!]\sigma \\ \sigma & \text{si no} \end{cases}$$

Esta definición sí es dirigida por sintaxis.

Veamos que F es continua. Para ello, primero vemos que es monótona: sean $w \sqsubseteq w' \in \Sigma \to \Sigma_\perp$. Si $[\![b]\!]\sigma$ y $[\![c]\!]\sigma = \perp$, entonces

$$\begin{array}{rcll} F \ w \ \sigma & = & w_{\perp\!\!\!\perp}([\![c]\!]\sigma) & \text{definición de } F \\ & = & \perp & [\![c]\!]\sigma = \perp \\ & \leq & F \ w' \ \sigma & \perp \text{ es el mínimo} \end{array}$$

En cambio, si $[\![b]\!]\sigma$ pero $[\![c]\!]\sigma \neq \perp$

$$\begin{array}{rcll} F \ w \ \sigma & = & w_{\perp\!\!\!\perp}([\![c]\!]\sigma) & \text{definición de } F \\ & = & w \ ([\![c]\!]\sigma) & [\![c]\!]\sigma \neq \perp \\ & \leq & w' \ ([\![c]\!]\sigma) & w \sqsubseteq w' \\ & = & w'_{\perp\!\!\!\perp}([\![c]\!]\sigma) & [\![c]\!]\sigma \neq \perp \\ & = & F \ w' \ \sigma & \text{definición de } F \end{array}$$

Por último, si $\neg[\![b]\!]\sigma$

$$\begin{array}{rcl} F \ w \ \sigma & = & \sigma \\ & = & F \ w' \ \sigma \end{array}$$

Por lo tanto, F es monótona. Por las propiedades ya demostradas, para comprobar continuidad, alcanza con demostrar que para toda cadena $w_0 \sqsubseteq w_1 \sqsubseteq w_2 \sqsubseteq \dots$ de funciones de $\Sigma \to \Sigma_\perp$, $F \ (\bigsqcup_{i=0}^{\infty} w_i) \sqsubseteq \bigsqcup_{i=0}^{\infty} F \ w_i$.

Nuevamente consideramos tres casos. Si $[\![b]\!]\sigma$ y $[\![c]\!]\sigma = \bot$, entonces

$$\begin{array}{rcll} F\ (\bigsqcup_{i=0}^{\infty} w_i)\ \sigma & = & (\bigsqcup_{i=0}^{\infty} w_i)_{\bot\!\!\!\bot}([\![c]\!]\sigma) & \text{definición de } F \\ & = & \bot & [\![c]\!]\sigma = \bot \\ & \leq & \bigsqcup_{i=0}^{\infty} F\ w_i\ \sigma & \bot \text{ es el mínimo} \end{array}$$

Ahora, si $[\![b]\!]\sigma$ pero $[\![c]\!]\sigma \neq \bot$

$$\begin{array}{rcll} F\ (\bigsqcup_{i=0}^{\infty} w_i)\ \sigma & = & (\bigsqcup_{i=0}^{\infty} w_i)_{\bot\!\!\!\bot}([\![c]\!]\sigma) & \text{definición de } F \\ & = & (\bigsqcup_{i=0}^{\infty} w_i)\ ([\![c]\!]\sigma) & [\![c]\!]\sigma \neq \bot \\ & = & \bigcup_{i=0}^{\infty} w_i\ ([\![c]\!]\sigma) & \text{supremo de cadena de funciones} \\ & = & \bigcup_{i=0}^{\infty} w_{i\bot\!\!\!\bot}\ ([\![c]\!]\sigma) & [\![c]\!]\sigma \neq \bot \\ & = & \bigcup_{i=0}^{\infty} F\ w_i\ \sigma & \text{definición de } F \\ & = & (\bigsqcup_{i=0}^{\infty} F\ w_i)\ \sigma & \text{supremo de cadena de funciones} \end{array}$$

Finalmente, si $\neg[\![b]\!]\sigma$,

$$\begin{array}{rcll} F\ (\bigsqcup_{i=0}^{\infty} w_i)\ \sigma & = & \sigma & \text{definición de } F \\ & = & \bigcup_{i=0}^{\infty}\ \sigma & \text{supremo de cadena constante} \\ & = & \bigcup_{i=0}^{\infty}\ F\ w_i\ \sigma & \text{definición de } F \\ & = & (\bigsqcup_{i=0}^{\infty} F\ w_i)\ \sigma & \text{supremo de cadena de funciones} \end{array}$$

Por lo tanto F es continua y la definición dada para la semántica del **while** está bien definida.

Ejemplos. Podemos considerar algunos ejemplos de cálculo de semántica de programas con **while**. El primero de ellos, **while true do skip** que es el caso más sencillo de un programa que no termina en ningún estado, y por ello, su semántica debe ser $\bot_{\Sigma\to\Sigma_\bot}$.

Sea $F \in (\Sigma \to \Sigma_\bot) \to (\Sigma \to \Sigma_\bot)$ definida por

$$\begin{array}{rcl} F\ w\ \sigma & = & \begin{cases} w_{\bot\!\!\!\bot}([\![\mathbf{skip}]\!]\sigma) & \text{si } [\![\mathbf{true}]\!]\sigma \\ \sigma & \text{si no} \end{cases} \\ & = & w_{\bot\!\!\!\bot}([\![\mathbf{skip}]\!]\sigma) \\ & = & w_{\bot\!\!\!\bot}\ \sigma \\ & = & w\ \sigma \end{array}$$

es decir, $F\ w = w$. Calculemos $F^i\ \bot_{\Sigma\to\Sigma_\bot}$ para algunos $i \in \mathbb{N}$:

$$\begin{array}{rcl} F^0\ \bot_{\Sigma\to\Sigma_\bot} & = & \bot_{\Sigma\to\Sigma_\bot} \\ F^1\ \bot_{\Sigma\to\Sigma_\bot} & = & F\ (F^0\ \bot_{\Sigma\to\Sigma_\bot}) \\ & = & F\ \bot_{\Sigma\to\Sigma_\bot} \\ & = & \bot_{\Sigma\to\Sigma_\bot} \end{array}$$

Tenemos entonces $F^1\ \bot_{\Sigma\to\Sigma_\bot} = F^0\ \bot_{\Sigma\to\Sigma_\bot}$. En general, si tenemos $F^{i+1}\ \bot_D = F^i\ \bot_D$ podemos concluir que $F^i\ \bot_D = F^{i+1}\ \bot_D = F^{i+2}\ \bot_D = \ldots$ y la cadena resulta no interesante y su supremo es $F^i\ \bot_D$.

En el caso de **while true do skip** obtenemos, $\bigsqcup_{i=0}^{\infty} F^i\ \bot_{\Sigma\to\Sigma_\bot} = \bot_{\Sigma\to\Sigma_\bot}$ que es lo que habíamos intuido.

Podíamos haber observado también, al obtener $F\ w = w$, que cualquier $w \in \Sigma \to \Sigma_\bot$ es punto fijo de F ya que F es la función identidad. El menor de ellos es el mínimo de $\Sigma \to \Sigma_\bot$, o sea, $\bot_{\Sigma\to\Sigma_\bot}$.

Un ejemplo no trivial es el de **while** $x \neq 0 \wedge x \neq 1$ **do** $x := x - 2$. Intuitivamente, este programa reemplaza el valor de x en el estado por el del módulo 2 de dicho valor si el mismo es no negativo. Si es negativo, no termina. Comparar con el caso considerado anteriormente de una función recursiva que tenía a $mod2$ como solución.

Tenemos

$$[\![\textbf{while } x \neq 0 \wedge x \neq 1 \textbf{ do } x := x - 2]\!] = \bigsqcup_{i=0}^{\infty} F^i \perp_{\Sigma \to \Sigma_\perp}$$

para

$$\begin{array}{rcl}
F\ w\ \sigma & = & \begin{cases} w_{\perp\!\!\perp}([\![x := x-2]\!]\sigma) & \text{si } [\![x \neq 0 \wedge x \neq 1]\!]\sigma \\ \sigma & \text{si no} \end{cases} \\
& = & \begin{cases} w_{\perp\!\!\perp}([\sigma|x : \sigma\ x - 2]) & \text{si } \sigma\ x \notin \{0,1\} \\ \sigma & \text{si no} \end{cases} \\
& = & \begin{cases} w\ ([\sigma|x : \sigma\ x - 2]) & \text{si } \sigma\ x \notin \{0,1\} \\ \sigma & \text{si no} \end{cases} \\
& = & \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ w\ [\sigma|x : \sigma\ x - 2] & \text{si no} \end{cases}
\end{array}$$

Calculemos $F^i \perp_{\Sigma \to \Sigma_\perp}$ para algunos $i \in \mathbb{N}$:

$$\begin{array}{rcl}
F^0 \perp_{\Sigma \to \Sigma_\perp} & = & \perp_{\Sigma \to \Sigma_\perp} \\
F^1 \perp_{\Sigma \to \Sigma_\perp} & = & F\ (F^0 \perp_{\Sigma \to \Sigma_\perp}) \\
& = & F \perp_{\Sigma \to \Sigma_\perp} \\
& = & \sigma \mapsto \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ \perp_{\Sigma \to \Sigma_\perp} [\sigma|x : \sigma\ x - 2] & \text{si no} \end{cases} \\
& = & \sigma \mapsto \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ \perp & \text{si no} \end{cases} \\
F^2 \perp_{\Sigma \to \Sigma_\perp} & = & F\ (F^1 \perp_{\Sigma \to \Sigma_\perp}) \\
& = & \sigma \mapsto \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ F^1 \perp_{\Sigma \to \Sigma_\perp} [\sigma|x : \sigma\ x - 2] & \text{si no} \end{cases} \\
& = & \sigma \mapsto \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ F^1 \perp_{\Sigma \to \Sigma_\perp} \overbrace{[\sigma|x : \sigma\ x - 2]}^{\sigma'} & \text{si no} \end{cases} \\
& = & \sigma \mapsto \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ \sigma' & \text{si } \sigma\ x \notin \{0,1\} \wedge \sigma'\ x \in \{0,1\} \\ \perp & \text{si } \sigma\ x \notin \{0,1\} \wedge \sigma'\ x \notin \{0,1\} \end{cases} \\
& = & \sigma \mapsto \begin{cases} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ [\sigma|x : \sigma\ x - 2] & \text{si } \sigma\ x \notin \{2,3\} \\ \perp & \text{si } \sigma\ x \notin \{0,1,2,3\} \end{cases} \\
& = & \sigma \mapsto \begin{cases} [\sigma|x : \sigma\ x \% 2] & \text{si } \sigma\ x \notin \{0,1,2,3\} \\ \perp & \text{si } \sigma\ x \notin \{0,1,2,3\} \end{cases}
\end{array}$$

$$\begin{array}{rcl}
F^3 \perp_{\Sigma\to\Sigma_\perp} & = & F\ (F^2 \perp_{\Sigma\to\Sigma_\perp}) \\
& = & \sigma \mapsto \left\{ \begin{array}{ll} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ F^2 \perp_{\Sigma\to\Sigma_\perp}\ \overbrace{[\sigma|x:\sigma\ x-2]}^{\sigma'} & \text{si no} \end{array} \right. \\
& = & \sigma \mapsto \left\{ \begin{array}{ll} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ [\sigma'|x:\sigma'\ x\,\%2] & \text{si } \sigma\ x \notin \{0,1\} \wedge \sigma'\ x \in \{0,1,2,3\} \\ \perp & \text{si } \sigma\ x \notin \{0,1\} \wedge \sigma'\ x \notin \{0,1,2,3\} \end{array} \right. \\
& = & \sigma \mapsto \left\{ \begin{array}{ll} \sigma & \text{si } \sigma\ x \in \{0,1\} \\ [\sigma'|x:\sigma'\ x\,\%2] & \text{si } \sigma\ x \in \{2,3,4,5\} \\ \perp & \text{si } \sigma\ x \notin \{0,1,2,3,4,5\} \end{array} \right. \\
& = & \sigma \mapsto \left\{ \begin{array}{ll} [\sigma|x:\sigma\ x\,\%2] & \text{si } \sigma\ x \in \{0,1,2,3,4,5\} \\ \perp & \text{si } \sigma\ x \notin \{0,1,2,3,4,5\} \end{array} \right.
\end{array}$$

Se puede comprobar que

$$F^i \perp_{\Sigma\to\Sigma_\perp} \quad = \quad \sigma \mapsto \left\{ \begin{array}{ll} [\sigma|x:\sigma\ x\,\%2] & \text{si } \sigma\ x \in \{0,1,\ldots,2*i-1\} \\ \perp & \text{si } \sigma\ x \notin \{0,1,\ldots,2*i-1\} \end{array} \right.$$

Observar que $F^i \perp_{\Sigma\to\Sigma_\perp}$ es la función que, aplicada a un estado para el que alcanzarían con (a lo sumo) $i-1$ iteraciones del ciclo para terminar, devuelve el estado final del **while**, y aplicada a un estado para el que no alcanzarían, devuelve $\perp$. En el límite esto daría la función que aplicada a un estado para el que alcanza con una cantidad finita de iteraciones, devuelve el estado final, y aplicada a un estado para el que no termina, devuelve $\perp$:

$$\bigsqcup_{i=0}^{\infty} F^i \perp_{\Sigma\to\Sigma_\perp} = \sigma \mapsto \left\{ \begin{array}{ll} [\sigma|x:\sigma\ x\,\%2] & \text{si } \sigma\ x \geq 0 \\ \perp & \text{si no} \end{array} \right.$$

que es lo que habíamos anticipado originariamente.