

Trabajo práctico N° 5

Mayo 2025

Estudiante Emanuel Nicolás Herrador

Ejercicio 1

Dado $P \equiv \mathbf{newvar} \ x := \mathbf{y} + \mathbf{x} \text{ in } (\mathbf{while} \ x > 0 \text{ do } (\mathbf{if} \ x > 0 \text{ then skip else fail}))$, se pretende caracterizar (sin usar la semántica) los estados σ en los cuales P se comporta como **skip**. Es decir, se pretende ver cuáles son aquellos estados en los cuales se termina la ejecución de este comando sin abortar y sin cambiar los valores del estado.

Es sencillo notar que como en ningún comando se cambia el valor de alguna variable de forma global (es decir, sin contar la definición local con **newvar**), entonces solo nos interesa diferenciar los casos donde la ejecución termina o no, y si esta termina con falla o no. El otro punto a notar es que como la guarda del while es $x > 0$ y la del if dentro de este es la misma, entonces este último resulta irrelevante. Es decir, el while en realidad es equivalente a **while** $x > 0$ **do skip**. Luego, los únicos casos donde la ejecución termina y sin fallas es cuando no entra dentro del while, es decir, cuando $x \leq 0$. Ahora, como la definición local es $x := x + y$, entonces la respuesta son aquellos estados $\sigma \in \Sigma : \sigma x + \sigma y \leq 0$.

Ejercicio 2

Se pretende ahora analizar las siguientes equivalencias usando la semántica denotacional de LIS con fallas. Veamos cada uno de los casos de forma separada.

Item A

Queremos analizar la equivalencia dada por $c; \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \equiv \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip}$. No son equivalentes y esto lo vamos a mostrar con un contraejemplo. Sea $c \equiv \mathbf{fail}$, queremos analizar ahora

$$\mathbf{fail}; \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \equiv \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip}$$

Veamos que para el lado izquierdo:

$$\begin{aligned} \llbracket \mathbf{fail}; \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket \sigma &= \llbracket \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket_* (\llbracket \mathbf{fail} \rrbracket \sigma) \\ &= \llbracket \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket_* \langle \mathbf{abort}, \sigma \rangle \\ &= \langle \mathbf{abort}, \sigma \rangle \end{aligned}$$

Mientras que para el lado derecho tenemos que:

$$\begin{aligned} \llbracket \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket \sigma &= \begin{cases} \llbracket \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket_* (\llbracket \mathbf{skip} \rrbracket \sigma) & \text{si } \llbracket \mathbf{true} \rrbracket \sigma \\ \sigma & \text{si } \neg \llbracket \mathbf{true} \rrbracket \sigma \end{cases} \\ &= \llbracket \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket \sigma \end{aligned}$$

por lo que para $F \in (\Sigma \rightarrow \Sigma'_\perp) \rightarrow (\Sigma \rightarrow \Sigma'_\perp)$ tenemos que $Fw\sigma = w\sigma$ y la semántica del while por TMPF es $w = \bigsqcup_{i \in \mathbb{N}} F^i \perp_{\Sigma \rightarrow \Sigma_\perp}$. Claramente, entonces, $\llbracket \mathbf{while} \ \mathbf{true} \ \mathbf{do} \ \mathbf{skip} \rrbracket = \perp_{\Sigma \rightarrow \Sigma_\perp}$ por lo que se refuta la equivalencia al mostrar este contraejemplo.

Item B

Queremos analizar la equivalencia dada por $c; \mathbf{fail} \equiv \mathbf{fail}$. Claramente, si bien en ambos casos se llega a una falla, no necesariamente se llega al mismo estado por lo que no es cierta. Motivo de esto, se va a dar un contraejemplo. Sea $c \equiv x := x + 1$, se pretende ver $x := x + 1; \mathbf{fail} \equiv \mathbf{fail}$. Veamos primero el lado izquierdo:

$$\begin{aligned} \llbracket x := x + 1; \mathbf{fail} \rrbracket \sigma &= \llbracket \mathbf{fail} \rrbracket_* (\llbracket x := x + 1 \rrbracket \sigma) \\ &= \llbracket \mathbf{fail} \rrbracket [\sigma \mid x : \sigma x + 1] \\ &= \langle \mathbf{abort}, [\sigma \mid x : \sigma x + 1] \rangle \end{aligned}$$

Mientras que para el lado derecho tenemos $\llbracket \mathbf{fail} \rrbracket \sigma = \langle \mathbf{abort}, \sigma \rangle$. Luego, entonces es claro que no se cumple la equivalencia gracias a este contraejemplo mostrado.

Item C

Queremos analizar la equivalencia **newvar** $v := e$ **in** $v := v + 1$; **fail** \equiv **newvar** $w := e$ **in** $w := w + 1$; **fail**. Estas dos son equivalentes, por lo que me concentraré en demostrarlo aquí. Notar que ambas semánticas son similares, por lo que solo me concentraré en calcularlo para el lazo izquierdo:

$$\begin{aligned}
\llbracket \text{newvar } v := e \text{ in } v := v + 1; \text{fail} \rrbracket \sigma &= (\lambda \sigma' \in \Sigma. [\sigma' \mid v : \sigma v])_{\dagger} (\llbracket v := v + 1; \text{fail} \rrbracket [\sigma \mid v : \llbracket e \rrbracket \sigma]) \\
&= (\lambda \sigma' \in \Sigma. [\sigma' \mid v : \sigma v])_{\dagger} (\llbracket \text{fail} \rrbracket_* (\llbracket v := v + 1 \rrbracket [\sigma \mid v : \llbracket e \rrbracket \sigma])) \\
&= (\lambda \sigma' \in \Sigma. [\sigma' \mid v : \sigma v])_{\dagger} (\llbracket \text{fail} \rrbracket [\sigma \mid v : \llbracket e \rrbracket \sigma + 1]) \\
&= (\lambda \sigma' \in \Sigma. [\sigma' \mid v : \sigma v])_{\dagger} \langle \text{abort}, [\sigma \mid v : \llbracket e \rrbracket \sigma + 1] \rangle \\
&= \langle \text{abort}, \sigma \rangle
\end{aligned}$$

Como en el lado derecho también se llega, de forma similar, a $\langle \text{abort}, \sigma \rangle$, entonces se demuestra la equivalencia. \square

Item D

Queremos analizar la equivalencia dada por **while** b **do** **fail** \equiv **if** b **then** **fail** **else** **skip**. Claramente son equivalentes, por lo que me centraré en demostrarlo. Primero, veamos el lado izquierdo:

$$\begin{aligned}
\llbracket \text{while } b \text{ do fail} \rrbracket \sigma &= \begin{cases} \llbracket \text{while } b \text{ do fail} \rrbracket_* (\llbracket \text{fail} \rrbracket \sigma) & \text{si } \llbracket b \rrbracket \sigma \\ \sigma & \text{si } \neg \llbracket b \rrbracket \sigma \end{cases} \\
&= \begin{cases} \llbracket \text{while } b \text{ do fail} \rrbracket_* \langle \text{abort}, \sigma \rangle & \text{si } \llbracket b \rrbracket \sigma \\ \sigma & \text{si } \neg \llbracket b \rrbracket \sigma \end{cases} \\
&= \begin{cases} \langle \text{abort}, \sigma \rangle & \text{si } \llbracket b \rrbracket \sigma \\ \sigma & \text{si } \neg \llbracket b \rrbracket \sigma \end{cases}
\end{aligned}$$

Ahora, respecto al lado derecho tenemos que:

$$\begin{aligned}
\llbracket \text{if } b \text{ then fail else skip} \rrbracket \sigma &= \begin{cases} \llbracket \text{fail} \rrbracket \sigma & \text{si } \llbracket b \rrbracket \sigma \\ \llbracket \text{skip} \rrbracket \sigma & \text{si } \neg \llbracket b \rrbracket \sigma \end{cases} \\
&= \begin{cases} \langle \text{abort}, \sigma \rangle & \text{si } \llbracket b \rrbracket \sigma \\ \sigma & \text{si } \neg \llbracket b \rrbracket \sigma \end{cases}
\end{aligned}$$

Por ello, entonces, se demuestra la equivalencia entre ambos comandos debido a que presentan la misma semántica para todo estado. \square

Item E

Queremos analizar la equivalencia dada por:

$$x := 0; \text{catchin } (\text{while } x < 1 \text{ do fail}) \text{ with } x := 1 \equiv x := 0; \text{while } x < 1 \text{ do } (\text{catchin fail with } x := 1)$$

Claramente son equivalentes, por lo que me concentraré en demostrarlo. Sean $w_1 = \llbracket \text{while } x < 1 \text{ do fail} \rrbracket$ y $w_2 = \llbracket \text{while } x < 1 \text{ do } (\text{catchin fail with } x := 1) \rrbracket$. Veamos primero el lado izquierdo:

$$\begin{aligned}
&\llbracket x := 0; \text{catchin } (\text{while } x < 1 \text{ do fail}) \text{ with } x := 1 \rrbracket \sigma \\
&= \llbracket \text{catchin } (\text{while } x < 1 \text{ do fail}) \text{ with } x := 1 \rrbracket_* (\llbracket x := 0 \rrbracket \sigma) \\
&= \llbracket \text{catchin } (\text{while } x < 1 \text{ do fail}) \text{ with } x := 1 \rrbracket [\sigma \mid x : 0] \\
&= \llbracket x := 1 \rrbracket_+ (w_1 [\sigma \mid x : 0])
\end{aligned}$$

Notemos que:

$$\begin{aligned}
w_1 [\sigma \mid x : 0] &= \begin{cases} w_{1*} (\llbracket \text{fail} \rrbracket [\sigma \mid x : 0]) & \text{si } \llbracket x < 1 \rrbracket [\sigma \mid x : 0] \\ [\sigma \mid x : 0] & \text{si } \neg \llbracket x < 1 \rrbracket [\sigma \mid x : 0] \end{cases} \\
&= w_{1*} \langle \text{abort}, [\sigma \mid x : 0] \rangle \\
&= \langle \text{abort}, [\sigma \mid x : 0] \rangle
\end{aligned}$$

Entonces, si seguimos con el lado izquierdo:

$$\begin{aligned} \llbracket \mathbf{x} := 1 \rrbracket_+ (w_1[\sigma \mid \mathbf{x} : 0]) &= \llbracket \mathbf{x} := 1 \rrbracket_+ \langle \mathbf{abort}, [\sigma \mid \mathbf{x} : 0] \rangle \\ &= [\sigma \mid \mathbf{x} : 1] \end{aligned}$$

Ahora, veamos el lado derecho:

$$\begin{aligned} &\llbracket \mathbf{x} := 0; \mathbf{while} \ \mathbf{x} < 1 \ \mathbf{do} \ (\mathbf{catchin} \ \mathbf{fail} \ \mathbf{with} \ \mathbf{x} := 1) \rrbracket \sigma \\ &= w_{2*}(\llbracket \mathbf{x} := 0 \rrbracket \sigma) \\ &= w_2[\sigma \mid \mathbf{x} : 0] \\ &= \begin{cases} w_{2*}(\llbracket \mathbf{catchin} \ \mathbf{fail} \ \mathbf{with} \ \mathbf{x} := 1 \rrbracket [\sigma \mid \mathbf{x} : 0]) & \text{si } \llbracket \mathbf{x} < 1 \rrbracket [\sigma \mid \mathbf{x} : 0] \\ [\sigma \mid \mathbf{x} : 0] & \text{si } \neg \llbracket \mathbf{x} < 1 \rrbracket [\sigma \mid \mathbf{x} : 0] \end{cases} \\ &= w_{2*}(\llbracket \mathbf{x} := 1 \rrbracket_+ (\llbracket \mathbf{fail} \rrbracket [\sigma \mid \mathbf{x} : 0])) \\ &= w_{2*}(\llbracket \mathbf{x} := 1 \rrbracket_+ \langle \mathbf{abort}, [\sigma \mid \mathbf{x} : 0] \rangle) \\ &= w_{2*} \llbracket \mathbf{x} := 1 \rrbracket [\sigma \mid \mathbf{x} : 0] \\ &= w_2[\sigma \mid \mathbf{x} : 1] \\ &= \begin{cases} w_{2*}(\llbracket \mathbf{catchin} \ \mathbf{fail} \ \mathbf{with} \ \mathbf{x} := 1 \rrbracket [\sigma \mid \mathbf{x} : 1]) & \text{si } \llbracket \mathbf{x} < 1 \rrbracket [\sigma \mid \mathbf{x} : 1] \\ [\sigma \mid \mathbf{x} : 1] & \text{si } \neg \llbracket \mathbf{x} < 1 \rrbracket [\sigma \mid \mathbf{x} : 1] \end{cases} \\ &= [\sigma \mid \mathbf{x} : 1] \end{aligned}$$

Y, con esto, entonces se demuestra que la equivalencia es correcta. \square

Siguientes ejercicios

Los siguientes ejercicios propuestos en esta guía corresponden a semántica operacional que no es un tema que va al primer parcial. Motivo de ello, aún no los realizaré para concentrarme en los demás temas que sí entran. Luego los haré.