Modelos y simulación - Prácticos

FAMAF - UNC

Severino Di Giovanni

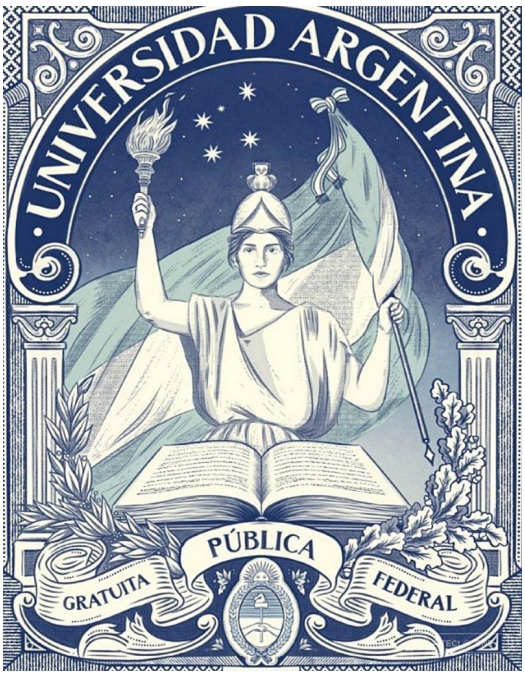

Figure 1: Severino Di Giovanni, el autor de este apunte. Un anarquista libertario, murió luchando por la libertad. Como él, otros miles han muerto para que nosotros gocemos de los derechos que tenemos. No te dejes engañar por los tristes pregoneros del egoísmo. Amá a tu prójimo y no olvides que si sus derechos se vulneran, los tuyos también. Ayudá a tu compañero de estudio, defendé tu universidad.

Contents

1 Práctico 4: Lenguaje imperativo simple

(**1**) Demostrar o refutar.
(c) (**if** b **then** c_0 **else** c_1); $c_2 \equiv$ **if** b **then** $c_0; c_2$ **else** $c_1; c_2$
(d) $c2$; (**if** b **then** $c0$ **else** $c1$) $\equiv$ **if** b **then** $c2; c0$ **else** $c_2; c_1$

(c) Sea $p =$ **if** b **then** c_0 **else** c_1 y

$$f = [\![\textbf{if } b \textbf{ then } c_0; c_2 \textbf{ else } c_1; c_2]\!] = \sigma \mapsto \begin{cases} [\![c_0; c_2]\!]\sigma & [\![b]\!]\sigma \\ [\![c_1; c_2]\!]\sigma & c.c. \end{cases}$$

Deseamos probar que $[\![p; c_2]\!] = f$. Por def.

$$\begin{aligned} [\![p; c_2]\!]\sigma &= [\![c_2]\!]_{\perp\!\!\perp}([\![p]\!]\sigma) \\ &= \begin{cases} [\![c_2]\!]_{\perp\!\!\perp}([\![c_0]\!]\sigma) & [\![b]\!]\,\sigma \\ [\![c_2]\!]_{\perp\!\!\perp}([\![c_1]\!]\sigma) & c.c. \end{cases} \\ &= \begin{cases} [\![c_2; c_0]\!]\sigma & [\![b]\!]\sigma \\ [\![c_2; c_1]\!]\sigma & c.c. \end{cases} \end{aligned}$$

$\therefore [\![p; c_2]\!] = f.$

(d) $[\![c_2;$ **if** b **then** c_0 **else** $c_1]\!] = [\![$**if** b **then** c_0 **else** $c_1]\!]_{\perp\!\!\perp}([\![c_2]\!]\sigma)$

$$\begin{aligned} &= \begin{cases} [\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!]([\![c_2]\!]\sigma) & [\![c_2]\!]\sigma \neq \perp \\ \perp & c.c. \end{cases} \\ &= \begin{cases} [\![c_0]\!]([\![c_2]\!]\sigma) & [\![c_2]\!]\sigma \neq \perp \wedge [\![b]\!]\sigma \\ [\![c_1]\!]([\![c_2]\!]\sigma) & [\![c_2]\!]\sigma \neq \perp \wedge \neg[\![b]\!]\sigma \\ \perp & [\![c_2]\!]\sigma = \perp \end{cases} \\ &= \begin{cases} [\![c_2; c_0]\!]\sigma & [\![c_2]\!]\sigma \neq \perp \wedge [\![b]\!]\sigma \\ [\![c_2; c_1]\!]\sigma & [\![c_2]\!]\sigma \neq \perp \wedge \neg[\![b]\!]\sigma \\ \perp & [\![c_2]\!]\sigma = \perp \end{cases} \\ &= \begin{cases} [\![c_2; c_0]\!]\sigma & [\![b]\!]\sigma \\ [\![c_2; c_1]\!]\sigma & \neg[\![b]\!]\sigma \end{cases} \\ &= [\![\textbf{if } b \textbf{ then } c_2; c_0 \textbf{ else } c_2; c_1]\!]\sigma \end{aligned}$$

(5) (a) Dar la semántica de **while** $x < 2$ **do if** $x < 0$ **then** $x := 0$ **else** $x := x+1$.

Razonamiento previo. Si $\sigma\ x < 2$, el **while** incrementa x hasta alcanzar el valor 2, por lo que su semántica converge a:

$$\sigma \mapsto \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x < 2 \end{cases}$$

En el peor caso ($\sigma\ x < 0$), el bucle realiza a lo sumo 3 iteraciones: una para corregir $x < 0$, y dos más para alcanzar 2.

De esto se sigue que: (1) el bucle siempre termina en a lo sumo 3 pasos, y (2) sólo $F^1\ \bot$ a $F^4\ \bot$ aportan información; luego, la cadena se vuelve no interesante.

Por simplicidad, hagamos $p :=$ **if** $x < 0$ **then** $x := 0$ **else** $x := x + 1$ y observemos que

$$[\![p]\!]\sigma = \begin{cases} [\sigma \mid x : 0] & \sigma\ x < 0 \\ [\sigma \mid x : \sigma\ x + 1] & \sigma\ x \geq 0 \end{cases} \tag{1}$$

Definamos $F : (\Sigma \mapsto \Sigma_\bot) \mapsto (\Sigma \mapsto \Sigma_\bot)$ como

$$F\ f\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ f_{\perp\!\!\!\perp}\ [\![p]\!]\ \sigma & \sigma\ x < 2 \end{cases}$$

Aplicando (1), obtenemos

$$F\ f\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ f\ ([\sigma \mid x : \sigma\ x + 1]) & \sigma\ x \in \{0, 1\} \\ f\ ([\sigma \mid x : 0]) & \sigma\ x < 0 \end{cases}$$

Es trivial observar que

$$F\ \bot\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ \bot & \sigma\ x < 2 \end{cases}$$

Ahora bien,

$$F^2\ \bot\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ (F\ \bot)\ ([\sigma \mid x : \sigma\ x + 1]) & \sigma\ x \in \{0, 1\} \\ (F\ \bot)\ ([\sigma \mid x : 0]) & \sigma\ x < 0 \end{cases}$$

En el caso $\sigma\ x \in \{0, 1\}$, tenemos

$$(F\ \bot)\ ([\sigma \mid x : \sigma\ x + 1]) = \begin{cases} F([\sigma \mid x : 2]) & \sigma\ x = 1 \\ F([\sigma \mid x : 1]) & \sigma\ x = 0 \end{cases} = \begin{cases} [\sigma \mid x : 2] & \sigma\ x = 1 \\ \bot & \sigma\ x = 0 \end{cases}$$

En el caso $\sigma\ x < 0$, claramente $F([\sigma \mid x < 0]) = \bot$. Con lo cual

$$F^2\ \bot\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x = 1 \\ \bot & \sigma\ x < 1 \end{cases}$$

De manera análoga se demuestra que

$$F^3\ \bot\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x \in \{0, 1\} \\ \bot & \sigma\ x < 1 \end{cases}$$

Entonces

$$\begin{aligned} F^4\ \bot\ \sigma &= \begin{cases} \sigma & \sigma\ x \geq 2 \\ (F^3\ \bot)\ ([\sigma \mid x : \sigma\ x + 1]) & \sigma\ x \in \{0, 1\} \\ (F^3\ \bot)\ ([\sigma \mid x : 0]) & \sigma\ x < 0 \end{cases} \\ &= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x < 2 \end{cases} \end{aligned}$$

Es obvio entonces que a partir de $k \geq 4$, $F^{k+1}\bot = F^k\bot$, con lo cual $F^1\ \bot, F_2\ \bot, \ldots$ es una cadena no interesante con supremo $F^4\ \bot$.

$$\therefore \bigsqcup_{i \in \mathbb{N}} F^i\ \bot = \lambda\sigma. \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x < 2 \end{cases} = [\![\textbf{if}\ \sigma\ x \geq 2\ \textbf{then skip else}\ \sigma\ x := 2]\!]$$

(5) (b) Dar la semántica de

while $x < 2$ **do if** $y = 0$ **then** $x := x + 1$ **else skip**

Debería ser claro que si $y \neq 0$ el ciclo no termina, pues se ejecuta **skip** indefinidamente.

Sea p el comando **if** ejecutado dentro del **while**. Si definimos $F : (\Sigma \mapsto \Sigma_\perp) \mapsto (\Sigma \mapsto \Sigma_\perp)$ como

$$F\ f\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ f_{\perp\!\!\perp}\ (\ [\![p]\!]\sigma) & \sigma\ x < 2 \end{cases}$$

entonces, desarrollando la semántica de p, tenemos

$$F\ f\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ f_{\perp\!\!\perp}\ ([\sigma \mid x : \sigma\ x + 1]) & \sigma\ x < 2 \wedge \sigma\ y = 0 \\ f_{\perp\!\!\perp}\sigma & \sigma\ x < 2 \wedge \sigma\ y \neq 0 \end{cases}$$

Ahora daremos el menor punto fijo de F, que será la semántica del comando. Claramente,

$$F\ \perp\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ \perp & \text{c. c.} \end{cases}$$

Continuando,

$$F^2\ \perp\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ (F\ \perp)_{\perp\!\!\perp}\ ([\sigma \mid x : \sigma\ x + 1]) & \sigma\ x < 2 \wedge \sigma\ y = 0 \\ (F\ \perp)_{\perp\!\!\perp}\sigma & \sigma\ x < 2 \wedge \sigma\ y \neq 0 \end{cases}$$

$$= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : \sigma\ x + 1] & \sigma\ x = 1 \wedge y = 0 \\ \perp & \sigma\ x < 1 \wedge y = 0 \\ \perp & \sigma\ x < 2 \wedge \sigma\ y \neq 0 \end{cases}$$

$$= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x = 1 \wedge y = 0 \\ \perp & \sigma\ x < 1 \wedge y = 0 \\ \perp & \sigma\ x < 2 \wedge \sigma\ y \neq 0 \end{cases}$$

Solo para ser explícitos, veamos que

$$F^3 \perp \sigma = \begin{cases} \sigma & \sigma\, x \geq 2 \\ (F \perp)^2_{\perp\!\!\!\perp} ([\sigma \mid x : \sigma\, x + 1]) & \sigma\, x < 2 \wedge \sigma\, y = 0 \\ (F \perp)^2_{\perp\!\!\!\perp} \sigma & \sigma\, x < 2 \wedge \sigma\, y \neq 0 \end{cases}$$

$$= \begin{cases} \sigma & \sigma\, x \geq 2 \\ [\sigma \mid x : 2] & \sigma\, x \in \{0, 1\} \wedge y = 0 \\ \perp & \sigma\, x < 0 \wedge y = 0 \\ \perp & \sigma\, x < 2 \wedge \sigma\, y \neq 0 \end{cases}$$

Planteamos como hipótesis inductiva que

$$F^k \perp \sigma = \begin{cases} \sigma & \sigma\, x \geq 2 \\ [\sigma \mid x : 2] & 3 - k \leq \sigma\, x \leq 1 \wedge y = 0 \\ \perp & c.c. \end{cases}$$

Entonces

$$F^{k+1}\ \bot\ \sigma = \begin{cases} \sigma & \sigma\ x \geq 2 \\ (F\ \bot)^k_{\Vdash}\ ([\sigma \mid x : \sigma\ x + 1]) & \sigma\ x < 2 \wedge \sigma\ y = 0 \\ (F\ \bot)^k_{\Vdash} \sigma & \sigma\ x < 2 \wedge \sigma\ y \neq 0 \end{cases}$$

$$= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : \sigma\ x + 1] & \sigma\ x < 2 \wedge \sigma\ x + 1 \geq 2 \wedge y = 0 \\ [\sigma \mid x : 2] & \sigma\ x < 2 \wedge 3 - k \leq \sigma\ x + 1 \leq 1 \wedge y = 0 \\ \bot & c.c. \end{cases}$$

$$= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x < 2 \wedge \sigma\ x \geq 1 \wedge y = 0 \\ [\sigma \mid x : 2] & \sigma\ x < 2 \wedge 3 - (k+1) \leq \sigma\ x \leq 0 \wedge y = 0 \\ \bot & c.c. \end{cases}$$

$$= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x = 1 \wedge y = 0 \\ [\sigma \mid x : 2] & 3 - (k+1) \leq \sigma\ x \leq 0 \wedge y = 0 \\ \bot & c.c. \end{cases}$$

$$= \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & 3 - (k+1) \leq \sigma\ x \leq 1 \wedge y = 0 \\ \bot & c.c. \end{cases}$$

quod erat demonstrandum. Se sigue entonces que

$$\bigsqcup_{i \in \mathbb{N}} F^i\ \bot = \lambda\sigma. \begin{cases} \sigma & \sigma\ x \geq 2 \\ [\sigma \mid x : 2] & \sigma\ x \leq 1 \wedge y = 0 \\ \bot & \sigma\ y \neq 0 \end{cases}$$

Aclaración. Para no escribir tanto, agrupamos $\bot$ en un solo caso durante el desarrollo de $F^1\ \bot, F^2\ \bot$, etc. Pero debería ser claro que en uno de los casos damos $\bot$ porque la cantidad de iteraciones es limitada, mientras que en otro caso damos $\bot$ porque $\sigma\ y \neq 0$. En el primer caso, a medida que se aumentan las iteraciones, se añade más y más información y, en el límite, la indefinición desaparece. En el segundo caso, la indefinición no desaparece: siempre que $\sigma\ y \neq 0$, se da $\bot$.

(6) Asuma que $[\![\textbf{while } b \textbf{ do } c]\!]\sigma \neq \bot$. Demuestre (a) que existe $n \geq 0$ tal que $F^n \perp \sigma \neq \bot$. Demuestre (b) que si $\sigma' = [\![\textbf{while } b \textbf{ do } c]\!]\sigma$, entonces $\neg[\![b]\!]\sigma'$.

(a) Sabemos que

$$[\![\textbf{while } b \textbf{ do } c]\!] = \bigsqcup_{i \in \mathbb{N}} F^i \perp$$

para

$$F\ f\ \sigma = \begin{cases} \sigma & \neg[\![b]\!]\sigma \\ f_{\perp\!\!\perp}[\![c]\!]\sigma & c.c. \end{cases}$$

Asuma que no existe $n \geq 0$ tal que $F^n \perp \sigma \neq \bot$. Se sigue que la cadena $\left\{F^i \perp\right\}_{i \in \mathbb{N}}$ es simplemente la cadena $\perp \sqsubseteq \perp \sqsubseteq \perp \sqsubseteq \ldots$. El supremo de esta cadena es $\perp$. Por lo tanto,

$$[\![\textbf{while } b \textbf{ do } c]\!] = \bigsqcup_{i \in \mathbb{N}} F^i \perp = \perp$$

lo cual contradice la hipótesis. La contradicción viene de asumir que no existe $n \geq 0$ tal que $F^n \perp \sigma \neq \bot$.

$\therefore$ Existe $n \geq 0$ tal que $F^n \perp \sigma \neq \bot$. ■

(b) Dado que la semántica de **while** b **do** c es un punto fijo de F, si usamos $\varphi := [\![\textbf{while } b \textbf{ do } c]\!]$, entonces

$$\varphi\ \sigma = F\ \varphi\ \sigma$$

Si σ es tal que $\neg[\![b]\!]\sigma$, entonces se sigue inmediatamente de la definición de F que en el estado $\varphi\sigma$ no se cumple b. Veamos el caso en que se cumple $[\![b]\!]\sigma$. Por la definición de F,

$$\varphi\ \sigma = \varphi_{\perp\!\!\perp}\ (\varphi_{\perp\!\!\perp} \ldots (\varphi_{\perp\!\!\perp}[\![c]\!]\sigma)) = \varphi^k_{\perp\!\!\perp}[\![c]\!]\sigma \tag{2}$$

donde la hipótesis de que el ciclo nunca es $\perp$ nos permite garantizar que existe tal $k \in \mathbb{N}$. Ahora bien, por la definición de F, k es definido estrictamente por el hecho de que

$$\neg[\![b]\!](\varphi^k_{\perp\!\!\perp}[\![c]\!]\sigma)$$

Por la ecuación (2), resulta entonces

$$\neg[\![b]\!]\ (\varphi\sigma) \quad ■$$

(7) Demostrar o refutar:
(a) **while false do** $c \equiv$ **skip**
(b) **while** b **do** $c \equiv$ **while** b **do** $(c; c)$
(c) (**while** b **do** c); **if** b **then** c_0 **else** $c_1 \equiv$ (**while** b **do** c); c_1

(a) Es trivial.

(b) Falso. Basta dar un contraejemplo. Sea σ un estado con $\sigma\, x = 0$ y considere

$$w_1 := \textbf{while}\ x \leq 0\ \textbf{do}\ x := x + 1, \qquad w_2 := \textbf{while}\ x \leq 0\ \textbf{do}\ (x := x + 1); (x := x + 1)$$

Claramente, $[\![w_1]\!]\sigma\, x = 1$ y $[\![w_2]\!]\sigma\, x = 2$. Sin embargo, dados comandos c_1, c_2,

$$c_1 \equiv c_2 \iff \forall \sigma \in \Sigma : [\![c_1]\!]\sigma = [\![c_2]\!]\sigma$$

$\therefore\ w_1 \not\equiv w_2$.

(c) Es verdadero. En el ejercicio anterior, demostramos que si un ciclo termina, entonces la guarda no puede cumplirse en el estado resultante del ciclo. Es decir que si

$$[\![\textbf{while}\ b\ \textbf{do}\ c]\!]\sigma = \sigma' \neq \bot$$

entonces $[\![b]\!]\sigma' \equiv \textbf{False}$. Por lo tanto, asumiendo que $[\![\textbf{while}\ b\ \textbf{do}\ c]\!]\sigma$ termina y no es $\bot$,

$$\begin{aligned}
&[\![(\textbf{while}\ b\ \textbf{do}\ c); \textbf{if}\ b\ \textbf{then}\ c_0\ \textbf{else}\ c_1]\!]\sigma \\
&= [\![\textbf{if}\ b\ \textbf{then}\ c_0\ \textbf{else}\ c_1]\!]\,([\![\textbf{while}\ b\ \textbf{do}\ c]\!]\sigma) \\
&= [\![c_1]\!]\,([\![\textbf{while}\ b\ \textbf{do}\ c]\!]\sigma) \\
&= [\![(\textbf{while}\ b\ \textbf{do}\ c); c_1]\!]\sigma \qquad \blacksquare
\end{aligned}$$

Ahora bien, si el ciclo no termina (es decir, si devuelve $\bot$), es trivial demostrar que la equivalencia también se cumple.

(8) Considerar las siguientes definiciones como syntactic sugar del comando

$$\textbf{for } v := e_0 \textbf{ to } e_1 \textbf{ do } c$$

(a) $v := e_0$; **while** $v \leq e_1$ **do** c; $v := v + 1$
(b) **newvar** $v := e_0$ **in while** $v \leq e_1$ **do** c; $v := v + 1$
(c) **newvar** $w := e_1$ **in newvar** $v := e_0$ **in while** $v \leq w$ **do** c; $v := v + 1$
¿Es alguna satisfactoria? Justificar.

Recordemos que, al menos de acuerdo con Reynolds,

$$\begin{aligned} &\textbf{for } v := e_0 \textbf{ to } e_1 \textbf{ do } c \\ :=\ &\textbf{newvar } w := e_1 \textbf{ in newvar } v := e_0 \textbf{ in while } v \leq w \textbf{ do } (c; v := v + 1) \end{aligned}$$

que es la expresión (c). Para no ser tramposos, igual justificaremos por qué dicha definición es satisfactoria, llegado el momento.

(a) La definición es satisfactoria en el sentido de que, si se la llama en un estado σ, ejecutará el comando c en los sucesivos estados

$$\begin{aligned} &[\sigma \mid v : [\![e_0]\!]\sigma] \\ &[\sigma \mid v : [\![e_0]\!]\sigma + 1] \\ &\vdots \\ &[\sigma \mid v : [\![e_1]\!]\sigma] \end{aligned}$$

Sin embargo, debemos notar que no se restaura el valor de v, i.e. v no es local al ciclo.

(b) Esta definición es funcional y restaura el valor de v. Sin embargo, es ineficiente, porque en cada llamada del while debe volver a computarse el valor de e_1 bajo el estado dado. Es concebible que e_1 sea una expresión compleja, e.g. una productoria de $k > 10.000$ variables, o cualquier locura que se nos ocurra. Por lo tanto, lo ideal sería computar la cota superior e_1 una sola vez y alocar dicho valor en otra variable.

(c) La definición (c) resuelve el problema de la (b), porque aloca en la variable local w el valor de la cota superior, que por lo tanto se computa una única vez. Una vez dicho valor es asignado a w, procede igual que en la def. (b): asigna a una variable local v el valor de e_0 e itera adecuadamente.

(10) Enunciar el teorema de coincidencia y demostra el caso **while**.

Teorema de coincidencia.
(a) Sean σ, σ' estados tales que $\sigma\ w = \sigma'\ w$ para toda $w \in FV(c)$. Entonces o bien $[\![c]\!]\sigma = [\![c]\!]\sigma' = \bot$ o bien $[\![c]\!]\sigma\ w = [\![c]\!]\sigma'\ w$ para toda $w \in FV(c)$.

(b) Si $[\![c]\!]\sigma \neq \bot$, entonces $[\![c]\!]\sigma\ w = \sigma\ w$ para toda $w \in FA(c)$.

(a) Sean σ_1, σ_2 tales que $\sigma_1\ w = \sigma_2\ w$ para todo $w \in FV(c)$, donde

$$c := \textbf{while}\ b\ \textbf{do}\ d$$

Por def. de FV, tenemos que $\sigma_1\ w = \sigma_2\ w$ para toda $w \in FV(b) \cup FV(d)$. Asumamos como hipótesis inductiva que el teorema vale para b y d, y definamos

$$\gamma_1 := [\![d]\!]\sigma_1, \qquad \gamma_{i+1} := [\![d]\!]\gamma_i$$
$$\beta_1 := [\![d]\!]\sigma_2, \qquad \beta_{i+1} := [\![d]\!]\beta_i$$

Es decir, $\{\gamma_i\}$ y $\{\beta_i\}$ son los estados correspondientes a las sucesivas iteraciones del **while**. Observemos que por HI resulta que $\gamma_1 = [\![d]\!]\sigma_1, \beta_1 = [\![d]\!]\sigma_2$ coinciden en las variables libres de d. Es fácil ver por inducción que entonces γ_i, β_i coinciden en las variables libres de d para toda i.

Observemos que b no tiene variables ligadas y por lo tanto, o bien es constante, o bien posee sólo variables libres. Si b es constante el problema es trivial porque o bien el while se ejecuta indefinidamente desde cualquier estado, o no se ejecuta desde ningún estado. Así que veamos el caso en que b tiene al menos una variable libre.

Observemos que si b y d no tienen variables libres en común, entonces las sucesivas ejecuciones de d no afectarán el valor de b y, por lo tanto, o bien el **while** se ejecuta sin terminar desde ambos estados, o no se ejecuta desde ninguno de los estados. Así que en este caso el problema es trivial. Por lo tanto, veamos el caso $FV(b) \cap FV(d) \neq \emptyset$.

Demostremos que $(\star)$ $[\![b]\!]\gamma_i = [\![b]\!]\beta_i$. Claramente, una ejecución de d sólo afecta la semántica de b a través de modificaciones de las variables en $FV(b) \cap FV(d) \subseteq FV(d)$. Pero ya establecimos que $\gamma_i\ w = \beta_i\ w$ para toda $w \in FV(d)$, y entonces esto es verdadero para el caso particular $w \in FV(b) \cap FV(d)$. Por lo tanto, γ_i, β_i coinciden en los valores de las variables libres de b que también son variables libres de d. Respecto a las variables libres de b que no son variables libres de d, las mismas no son modificadas por las sucesivas ejecuciones de d. Por lo tanto tienen el mismo valor que en σ_1 y σ_2, respectivamente. Pero σ_1, σ_2 coinciden en las variables libres de b. Por lo tanto, coinciden también en estas variables.

Asuma que $[\![c]\!]\sigma_1 = \bot$. Entonces, para toda i se cumple que $[\![b]\!]\gamma_i \equiv \textbf{True}$ (de otro modo el **while** terminaría). Por $(\star)$ se sigue que $[\![b]\!]\beta_i \equiv \textbf{True}$. Como esto vale para toda i, las sucesivas iteraciones de $\{\beta_i\}$ nunca hacen la guarda falsa. Por lo tanto, el **while** nunca termina partiendo desde σ_2. $\therefore [\![c]\!]\sigma_2 = \bot$

Asuma que $[\![c]\!]\sigma_1 \neq \perp$. Un razonamiento idéntico al anterior nos da que $[\![c]\!]\sigma_2 \neq \perp$, y no sólo eso sino que se da la misma cantidad k de iteraciones en ambos casos. Es decir que los estados finales de ambos casos son γ_k, β_k, respectivamente. Ya observamos antes que γ_k, β_k coinciden en las variables libres de d y de b, lo cual concluye la prueba.