Tema 2. Semántica operacional del lenguaje WHILE

Semántica de paso largo

Transiciones

Configuraciones

- La *sentencia* S se ejecuta desde el estado s: $\langle S, s\rangle$.
- Estado s' que se alcanza al terminar su ejecución.

Transiciones ($\langle S, s\rangle \to s'$) :

- Asignación:

$$[\mathrm{ass_{ns}}] := \langle x := a, s\rangle \to s\left[x \mapsto \mathcal{A}[\![a]\!]s\right]$$

- *Skip*:

$$[\mathrm{skip_{ns}}] := \langle \mathtt{skip}, s\rangle \to s$$

- Composición:

$$[\mathrm{comp_{ns}}] := \frac{\langle S_1, s\rangle \to s',\ \langle S_2, s'\rangle \to s''}{\langle S_1; S_2, s\rangle \to s''}$$

- Condicional:
  - Si se cumple:

$$\left[\mathrm{if}_{\mathrm{ns}}^{\mathrm{tt}}\right] := \frac{\langle S_1, s\rangle \to s'}{\langle \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2, s\rangle \to s'},\ \ \mathrm{si}\ \mathcal{B}[\![b]\!] = \mathbf{tt}$$

  - Si no se cumple:

$$\left[\mathrm{if}_{\mathrm{ns}}^{\mathrm{ff}}\right] := \frac{\langle S_2, s\rangle \to s'}{\langle \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2, s\rangle \to s'},\ \ \mathrm{si}\ \mathcal{B}[\![b]\!] = \mathbf{ff}$$

- Bucle:
  - Si se cumple:

$$\left[\text{while}_{\text{ns}}^{\text{tt}}\right] := \frac{\langle S, s\rangle \to s', \langle \texttt{while}\ b\ \texttt{do}\ S, s''\rangle \to s''}{\langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s''},\ \ \text{si}\ \mathcal{B}[\![b]\!] = \mathbf{tt}$$

  - Si no se cumple:

$$\left[\text{while}_{\text{ns}}^{\text{ff}}\right] := \langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s,\ \ \text{si}\ \mathcal{B}[\![b]\!] = \mathbf{ff}$$

Una sentencia S es un estado s tiene dos posibilidades:

- **Terminar** si, y sólo si, $\exists s' \in \mathbf{State}. \langle S, s\rangle \to s'$.
- **Ciclar** si, y sólo si, $\nexists s' \in \mathbf{State}. \langle S, s\rangle \to s'$.

Árboles de derivación

Para todo programa escrito en el lenguaje **WHILE** podemos escribir un árbol de derivación. Este árbol tendrá como raíz el programa entero y sus hijos serán aplicaciones de las reglas de la anterior sección. Las hojas serán *axiomas* (aquellas reglas que no tienen «conclusión»).

Si el programa «cicla» tendrá un árbol infinito, mientras que si termina será finito. En esta versión del **WHILE** para toda transición habrá una sola regla que se pueda aplicar.

Inducción sobre el árbol de derivación

1. Probar que la propiedad se cumple para los *axiomas*.

2. Probar para cada regla compuesta: asumiendo que se cumple para las hipótesis, ver que también es cierta para las conclusiones.

Propiedades

Definición (Equivalencia entre sentencias)
Decimos que dos sentencias S_1 y S_2 son semánticamente equivalentes si, y sólo si, $\forall s \in \mathbf{State}$ se cumple que:

$$\langle S_1, s\rangle \to s' \Leftrightarrow \langle S_2, s\rangle \to s'$$

Definición y proposición (Determinismo)
Decimos que una semántica es *determinista* si dada una sentencia S y cualquier

estado s se cumple que:

$$\langle S, s\rangle \to s' \wedge \langle S, s\rangle \to s'' \Rightarrow s' = s''$$

Función semántica

Definición (Significado de una sentencia)
Definimos la semántica operacional de paso largo de las sentencias a través de la siguiente función:

$$\mathcal{S}_{ns} : \mathbf{Stm} \to (\mathbf{State} \hookrightarrow \mathbf{State})$$
$$\mathcal{S}_{ns}[\![S]\!]s = \begin{cases} s', & \text{si } \langle S, s\rangle \to s' \\ \texttt{indefinido}, & \text{c.c} \end{cases}$$

Semántica de paso corto

Transición
Primer paso de ejecución de S: $\langle S, s\rangle \Rightarrow \gamma$.

- Si $\gamma = \langle S', s'\rangle$, ejecución *no terminada*.
- Si $\gamma = s'$, ejecución *terminada*.

Si para un $\langle S, s\rangle$ *no existe* ningún $\gamma.\langle S, s\rangle \Rightarrow \gamma$, la configuración estará *bloqueada*.

- Asignación:

$$[\text{ass}_{\text{sos}}] := \langle x := a, s\rangle \Rightarrow s\,[x \mapsto \mathcal{A}[\![a]\!]s]$$

- *Skip*:

$$[\text{skip}_{\text{sos}}] := \langle \texttt{skip}, s\rangle \Rightarrow s$$

- Composición:

$$\left[\text{comp}^1_{\text{sos}}\right] := \frac{\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_1'; S_2, s'\rangle}$$
$$\left[\text{comp}^2_{\text{sos}}\right] := \frac{\langle S_1, s\rangle \Rightarrow s'}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_2, s'\rangle}$$

- Condicional:
  - Si se cumple:

$$\left[\text{if}^{\text{tt}}_{\text{sos}}\right] := \langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, s\rangle \Rightarrow \langle S_1, s\rangle,\ \ \text{si}\ \mathcal{B}[\![b]\!] = \mathbf{tt}$$

- Si no se cumple:

$$\left[\text{if}^{\text{tt}}_{\text{sos}}\right] := \langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, s\rangle \Rightarrow \langle S_2, s\rangle,\ \ \text{si}\ \mathcal{B}[\![b]\!] = \mathbf{ff}$$

- Bucle:

$$[\text{while}_{\text{sos}}] := \langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \Rightarrow \\ \langle \texttt{if}\ b\ \texttt{then}\ (S,\ \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else}\ \texttt{skip}, s\rangle$$

Las secuencias de derivación pueden *terminar* (con éxito, alcanza un s', o no) o *ciclar*.

Inducción sobre la longitud de secuencia

1. Demostrar la propiedad para las secuencias de longitud 0.

2. Asumiendo que se cumple para las secuencias de longitud k, probarla para secuencias de longitud $k+1$.

Propiedades

Lema
Si tenemos $\langle S_1; S_2, s\rangle \Rightarrow^k s''$, entonces existen $s' \in \mathbf{State}$ y $k_1, k_2 \in \mathbb{N}$ tales que:

$$\langle S_1, s\rangle \Rightarrow^{k_1} s' \wedge \langle S_2, s\rangle \Rightarrow^{k_2} s'',\ \text{con}\ k = k_1 + k_2.$$

Definición (Equivalencia semántica)
Decimos que dos sentencias S_1 y S_2 son semánticamente equivalentes si para todo estado s se cumple que:

- $\langle S_1, s\rangle \Rightarrow^* \gamma \Leftrightarrow \langle S_2, s\rangle \Rightarrow^* \gamma$, para cada γ terminal o bloqueada.
- La secuencia que inicia con $\langle S_1, s\rangle$ es *infinita* si, y sólo si, lo es la que inicia con $\langle S_2, s\rangle$.

Función semántica

Definición (Significado de una sentencia)
Definimos la semántica operacional de paso corto de las sentencias a través de la siguiente función:

$$\mathcal{S}_{sos} : \mathbf{Stm} \to (\mathbf{State} \hookrightarrow \mathbf{State})$$

$$\mathcal{S}_{sos}[\![S]\!]s = \begin{cases} s', & \text{si } \langle S, s\rangle \Rightarrow^* s' \\ \texttt{indefinido}, & \text{c.c} \end{cases}$$

Equivalencia

Teorema

Para toda sentencia $S \in \mathbf{Stm}$ se cumple la siguiente relación:

$$\mathcal{S}_{ns}[\![S]\!] = \mathcal{S}_{sos}[\![S]\!]$$

Es decir, la semántica operacional de paso corto y paso largo con *equivalentes*.