Análisis de Lenguajes de Programación

Semántica Operacional

15 de Septiembre de 2025

Lenguaje imperativo simple

Veremos la semántica operacional de un lenguaje imperativo simple, donde modelaremos los efectos laterales de:

- estado
- errores
- E/S

Sintaxis

$$
\begin{aligned}
ie \;:=\;& nv \\
&\mid var \\
&\mid -ie \\
&\mid ie + ie \\
&\mid ie -_b ie \\
&\mid ie \times ie \\
&\mid ie \div ie
\end{aligned}
\qquad\qquad
\begin{aligned}
be \;:=\;& \mathtt{true} \mid \mathtt{false} \\
&\mid ie = ie \\
&\mid ie < ie \\
&\mid ie > ie \\
&\mid be \wedge be \\
&\mid be \vee be \\
&\mid \neg be
\end{aligned}
$$

$$
\begin{aligned}
comm \;:=\;& \mathtt{skip} \mid var := ie \\
&\mid comm \;; comm \\
&\mid \mathtt{if}\ be\ \mathtt{then}\ comm\ \mathtt{else}\ comm \\
&\mid \mathtt{while}\ be\ comm
\end{aligned}
$$

Estado

- El lenguaje solo contiene variables enteras. La noción de estado se puede modelar con la siguiente definición:

  $$\Sigma = Var \rightarrow nv$$

  Es decir, un estado es una función total entre identificadores y enteros.

- Para modificar el estado utilizaremos la siguiente notación:

  $[\sigma \mid v : n]$ que representa un estado que coincide con σ en todas las variables salvo posiblemente en v, donde tiene asignado n.

Relaciones de evaluación de paso grande

- Para expresiones enteras la relación tiene este tipo:

$$\Downarrow_i \in \ (ie \times \Sigma) \times nv$$

- Regla para variables:

$$\frac{}{(v, \sigma) \ \Downarrow_i \ \sigma(v)} \qquad (\textsc{Var})$$

- Las demás reglas se extienden con un estado, por ejemplo:

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}} n_0 \qquad (e_1, \sigma) \Downarrow_{\mathrm{i}} n_1}{(e_0 + e_1, \sigma) \Downarrow_{\mathrm{i}} n_0 + n_1} \qquad (\textsc{Add})$$

Evaluación para expresiones booleanas

- La relación de evaluación tiene este tipo:

$$\Downarrow_b \in \ (be \times \Sigma) \times bv$$

- Las reglas se extienden con un estado, por ejemplo:

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{b}} b_0 \qquad (e_1, \sigma) \Downarrow_{\mathrm{b}} b_1}{(e_0 \wedge e_1, \sigma) \Downarrow_{\mathrm{b}} b_0 \wedge b_1} \qquad (\textsc{And})$$

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}} n_0 \qquad (e_1, \sigma) \Downarrow_{\mathrm{i}} n_1}{(e_0 = e_1, \sigma) \Downarrow_{\mathrm{b}} n_0 = n_1} \qquad (\textsc{Eq})$$

Evaluación para comandos

La relación de evaluación tiene este tipo:

$$\Rightarrow \in (comm \times \Sigma) \times \Sigma$$

$$\frac{(e, \sigma) \Downarrow_{\mathrm{i}} n}{(x := e, \sigma) \Rightarrow [\sigma \mid x : n]} \ (\textsc{Ass})$$

$$\frac{}{(\texttt{skip}\ , \sigma) \Rightarrow \sigma} \ (\textsc{Skip})$$

$$\frac{(c_0, \sigma) \Rightarrow \sigma' \qquad (c_1, \sigma') \Rightarrow \sigma''}{(c_0 \,; c_1, \sigma) \Rightarrow \sigma''} \ (\textsc{Seq})$$

$$\frac{(e, \sigma) \Downarrow_{\mathrm{b}} \texttt{true} \qquad (c_1, \sigma) \Rightarrow \sigma'}{(\texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2, \sigma) \Rightarrow \sigma'} \ (\textsc{If-T})$$

$$\frac{(e, \sigma) \Downarrow_{\mathrm{b}} \texttt{false} \qquad (c_2, \sigma) \Rightarrow \sigma'}{(\texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2, \sigma) \Rightarrow \sigma'} \ (\textsc{If-F})$$

Evaluación para comandos

$$\frac{(e,\sigma) \Downarrow_{\mathrm{b}} \texttt{true} \qquad (c,\sigma) \Rightarrow \sigma' \qquad (\texttt{while}\ \ e\ c,\sigma') \Rightarrow \sigma''}{(\texttt{while}\ \ e\ c,\sigma) \Rightarrow \sigma''} \quad (\textsc{While-T})$$

$$\frac{(e,\sigma) \Downarrow_{\mathrm{b}} \texttt{false}}{(\texttt{while}\ \ e\ c,\sigma) \Rightarrow \sigma} \quad (\textsc{While-F})$$

Evaluación de paso chico para comandos

- La relación tiene el siguiente tipo:

$$\leadsto \in \ (comm \times \Sigma) \times (comm \times \Sigma)$$

- Si la ejecución termina lo hace en una configuración de la forma $(\texttt{skip}\ , \sigma)$, para algún σ.

- Las ejecuciones de la asignación y `skip` terminan en un paso, son similares a la de paso grande.

- Secuenciamiento:

$$\frac{(c_0, \sigma) \leadsto (c_0', \sigma')}{(c_0\,; c_1, \sigma) \leadsto (c_0'\,; c_1, \sigma')} \qquad (\textsc{Seq}_1)$$

$$\frac{}{(\texttt{skip}\ ; c_1, \sigma) \leadsto (c_1, \sigma)} \qquad (\textsc{Seq}_2)$$

Evaluación de paso chico para comandos

- Condicionales (mantenemos evaluación de paso grande para expresiones):

$$\frac{e \Downarrow_{\mathrm{b}} \texttt{true}}{(\texttt{if } e \texttt{ then } c_0 \texttt{ else } c_1, \sigma) \rightsquigarrow (c_0, \sigma)} \qquad (\textsc{If-T})$$

$$\frac{e \Downarrow_{\mathrm{b}} \texttt{false}}{(\texttt{if } e \texttt{ then } c_0 \texttt{ else } c_1, \sigma) \rightsquigarrow (c_1, \sigma)} \qquad (\textsc{If-F})$$

Evaluación de paso chico para comandos

- Bucles:

$$\frac{e \Downarrow_{\mathrm{b}} \texttt{false}}{(\texttt{while}\ e\ c, \sigma) \rightsquigarrow (\texttt{skip}\ , \sigma)} \quad (\textsc{While-F})$$

$$\frac{e \Downarrow_{\mathrm{b}} \texttt{true}}{(\texttt{while}\ e\ c, \sigma) \rightsquigarrow (c\,;\texttt{while}\ e\ c, \sigma)} \quad (\textsc{While-T})$$

Agregamos errores

- Agregamos un valor que representa un error: $\mathbf{err_i}$
- Las expresiones enteras que fallen devolverán este valor.
- La relación de evaluación tendrá este tipo:

$$\Downarrow_i \in \ (ie \times \Sigma) \times (nv \cup \{\ \mathbf{err_i}\})$$

- Agregamos reglas que generan errores:

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}}\ n_0 \qquad (e_1, \sigma) \Downarrow_{\mathrm{i}}\ 0}{(e_0 \div e_1, \sigma) \Downarrow_{\mathrm{i}}\ \mathbf{err_i}} \qquad (\textsc{Div0})$$

Agregamos errores

- Distinguimos las divisiones que no generan error.

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}} n_0 \qquad (e_1, \sigma) \Downarrow_{\mathrm{i}} n_1 \qquad n_1 \neq 0}{(e_0 \div e_1, \sigma) \Downarrow_{\mathrm{i}} n_0 \div n_1} \quad (\textsc{Div-1})$$

- Agregamos reglas para propagar el error:

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}} \mathbf{err_i}}{(e_0 \div e_1, \sigma) \Downarrow_{\mathrm{i}} \mathbf{err_i}} \quad (\textsc{Div-2})$$

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}} n_0 \qquad (e_1, \sigma) \Downarrow_{\mathrm{i}} \mathbf{err_i}}{(e_0 \div e_1, \sigma) \Downarrow_{\mathrm{i}} \mathbf{err_i}} \quad (\textsc{Div-3})$$

Agregamos errores

- Agregamos un valor que representa un error boolano: $\mathbf{err_b}$
- La relación de evaluación tendrá este tipo:

$$\Downarrow_b \in \; (be \times \Sigma) \times (bv \cup \{\, \mathbf{err_b} \})$$

- Modificamos las reglas para generar y propagar errores booleanos.

$$\frac{(e_0, \sigma) \Downarrow_{\mathrm{i}} \mathbf{err_i}}{(e_0 = e_1, \sigma) \Downarrow_{\mathrm{b}} \mathbf{err_b}} \qquad (\textsc{Eq}\text{-}0)$$

Agregamos errores

- Modificamos la relación para comandos:

$$\leadsto \in (comm \times \Sigma) \times ((comm \cup \{\mathbf{err_c}\}) \times \Sigma)$$

- Ejemplos de algunas reglas:

$$\frac{(e, \sigma) \Downarrow_{\mathrm{i}} \mathbf{err_i}}{(x := e, \sigma) \leadsto (\mathbf{err_c}, \sigma)} \qquad (\textsc{Ass-0})$$

$$\frac{(c_0, \sigma) \leadsto (\mathbf{err_c}, \sigma')}{(c_0 \,;\, c_1, \sigma) \leadsto (\mathbf{err_c}, \sigma')} \qquad (\textsc{Seq-0})$$

Agregamos E/S

- Agregamos etiquetas a las transiciones entre configuraciones, donde:
  - $n?$ indicará la entrada de un entero n
  - $n!$ indicará la salida de un entero n
- La relación de evaluación tendrá este tipo:

$$\rightsquigarrow \in \ (comm \times \Sigma) \times l \times ((comm \cup \{\mathbf{err_c}\}) \times \Sigma)$$

  donde $l := nv? \mid nv! \mid \tau$

- La etiqueta τ representa la transición silenciosa, escribimos $x \rightsquigarrow y$ en lugar de $x \overset{\tau}{\rightsquigarrow} y$

Agregamos E/S

- Extendemos la sintaxis de los comandos:

$$comm \ := \ \ldots \mid \texttt{input} \ \ var \mid \texttt{print} \ \ ie$$

- Agregamos reglas semánticas para estos comandos:

$$\frac{}{(\texttt{input} \ \ v, \sigma) \overset{n?}{\rightsquigarrow} (\texttt{skip} \ \ , [\sigma | v : n])} \qquad (\textsc{Input})$$

$$\frac{(e, \sigma) \Downarrow_{\mathrm{i}} n}{(\texttt{print} \ \ e, \sigma) \overset{n!}{\rightsquigarrow} (\texttt{skip} \ \ , \sigma)} \qquad (\textsc{Print})$$

- Reescribimos las transiciones de las reglas con una etiqueta l.