Más sobre Semántica Operacional

David de Frutos Escrig

versión original elaborada por

Yolanda Ortega Mallén

Dpto. de Sistemas Informáticos y Computación

Universidad Complutense de Madrid

Sumario

- Más allá de lo puramente secuencial determinista.
- Bloques y procedimientos: necesidad de establecer *contextos*.

Bibliografía

- Hanne Riis Nielson & Flemming Nielson,
  Semantics with Applications. An Appetizer, Springer, 2007.
  Capítulo 3.

Abortando cómputos

Añadimos una nueva instrucción simple

$$S ::= x := a \mid \texttt{skip} \mid S_1 \,;\, S_2 \mid \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \mid \texttt{while}\ b\ \texttt{do}\ S \mid \texttt{abort}$$

Tratamiento semántico y consecuencias

- No se añaden reglas semánticas: reflejando la idea de indefinición.
- Paso corto: `abort` no es semánticamente equivalente a `skip` ni a `while true do skip`.
- Paso largo: `abort` no es semánticamente equivalente a `skip`, pero sí a `while true do skip`.

Abortando cómputos

Semántica de paso largo vs. semántica de paso corto

1. La semántica de paso largo no distingue entre ciclar y terminación anormal.
2. En paso corto, ciclar se refleja en las secuencias infinitas de derivación, y la terminación anormal en secuencias de derivación finitas, pero bloqueadas.

Ejercicio 3.2

Extender el lenguaje **While** con la sentencia `assert` b `before` S.

- Comprobamos que b se cumple antes de ir a ejecutar S
- Definir adecuadamentr su semántica de paso corto.
- Demostrar que `assert true before` S es semánticamente equivalente a S, pero
  `assert false before` S no es semánticamente equivalente ni a `skip` ni a `while true do skip`
- ¿A quién sí sería equivalente?.

Indeterminación

Añadimos una nueva instrucción compuesta

$$S ::= x := a \mid \texttt{skip} \mid S_1 \,;\, S_2 \mid \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2 \mid \texttt{while } b \texttt{ do } S \mid S_1 \texttt{ or } S_2$$

Semántica de paso largo

$$[\text{or}^1_{\text{bs}}] \quad \frac{\langle S_1, s\rangle \to s'}{\langle S_1 \texttt{ or } S_2, s\rangle \to s'} \qquad [\text{or}^2_{\text{bs}}] \quad \frac{\langle S_2, s\rangle \to s'}{\langle S_1 \texttt{ or } S_2, s\rangle \to s'}$$

Semántica de paso corto

$$[\text{or}^1_{\text{ss}}] \quad \langle S_1 \texttt{ or } S_2, s\rangle \Rightarrow \langle S_1, s\rangle$$

$$[\text{or}^1_{\text{ss}}] \quad \langle S_1 \texttt{ or } S_2, s\rangle \Rightarrow \langle S_2, s\rangle$$

Semántica de paso largo vs. semántica de paso corto

1. En paso largo, la indeterminación elimina los ciclos, cuando es posible.
2. En paso corto, mantiene siempre su presencia.

Paralelismo

Añadimos una nueva instrucción compuesta

$$S ::= x := a \mid \texttt{skip} \mid S_1 ; S_2 \mid \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2 \mid \texttt{while } b \texttt{ do } S \mid S_1 \texttt{ par } S_2$$

Semántica de paso corto

$$[\text{par}^1_{\text{ss}}] \quad \frac{\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle}{\langle S_1 \texttt{ par } S_2, s\rangle \Rightarrow \langle S_1' \texttt{ par } S_2, s'\rangle}$$

$$[\text{par}^3_{\text{ss}}] \quad \frac{\langle S_2, s\rangle \Rightarrow \langle S_2', s'\rangle}{\langle S_1 \texttt{ par } S_2, s\rangle \Rightarrow \langle S_1 \texttt{ par } S_2', s'\rangle}$$

$$[\text{par}^{2n}_{\text{ss}}] \quad \frac{\langle S_n, s\rangle \Rightarrow s'}{\langle S_1 \texttt{ par } S_2, s\rangle \Rightarrow \langle S_{3-n}, s'\rangle}$$

Semántica de paso largo vs. semántica de paso corto

1. En paso largo la ejecución de las componentes es atómica; no se puede expresar la intercalación.
2. En paso corto la intercalación se expresa fácilmente.

Bloques y declaración de variables

Nueva categoría sintáctica: Las declaraciones de variables $D_V \in \mathbf{Dec}_V$

Añadimos las declaraciones y una nueva instrucción compuesta

$$D_V ::= \texttt{var}\ x := a\,;\, D_V \mid \varepsilon$$

$$S ::= x := a \mid \texttt{skip} \mid S_1\,;\, S_2 \mid \texttt{if}\ldots \mid \texttt{while}\ b\ \texttt{do}\ S \mid \texttt{begin}\ D_V\ S\ \texttt{end}$$

Recuperando los valores de las variables declaradas en un bloque: $s[X \longmapsto s']$

Semántica de paso largo

$$[\text{var}_{\text{bs}}] \quad \frac{\langle D_V, s[x \mapsto \mathcal{A}[\![a]\!]s]\rangle \to_D s'}{\langle \texttt{var}\ x := a\,;\, D_V, s\rangle \to_D s'}$$

$$[\text{none}_{\text{bs}}] \quad \langle \varepsilon, s\rangle \to_D s \qquad \text{(El estado acumulado se devuelve)}$$

$$[\text{block}_{\text{bs}}] \quad \frac{\langle D_V, s\rangle \to_D s',\ \langle S, s'\rangle \to s''}{\langle \texttt{begin}\ D_V\ S\ \texttt{end}, s\rangle \to s''[\text{DV}(D_V) \longmapsto s]}$$

Procedimientos

Nueva categoría sintáctica: Las declaraciones de procedimientos $D_P \in \mathbf{Dec}_P$.

Añadimos las declaraciones, extendemos la definición de los bloques, y añadimos una nueva instrucción simple

$$D_P ::= \texttt{proc}\ p\ \texttt{is}\ S\ ;\ D_P \mid \varepsilon$$

$$S ::= x := a \mid \ldots \mid \texttt{while}\ b\ \texttt{do}\ S \mid \texttt{begin}\ D_V\ D_P\ S\ \texttt{end} \mid \texttt{call}\ p$$

Múltiples posibilidades a la hora de precisar los ámbitos

- Ámbito dinámico para variables y procedimientos.
- Ámbito dinámico para variables y estático para procedimientos.
- Ámbito estático para variables y procedimientos.

Ámbito dinámico (variables y procedimientos)

- Entorno de procedimientos: $env_P \in \mathbf{Env}_P = \mathbf{Pname} \hookrightarrow \mathbf{Stm}$

Acumulación de declaraciones de procedimientos

$$\begin{aligned} \mathsf{updP}(\texttt{proc}\ p\ \texttt{is}\ S\,;D_P, env_P) &= \mathsf{updP}(D_P, env_P[p \mapsto S]) \\ \mathsf{updP}(\varepsilon, env_P) &= env_P \end{aligned}$$

- Nuevo Sistema de transiciones: $env_P \vdash \langle S, s\rangle \to s'$

Las llamadas generan la ejecución de los cuerpos de los procedimientos

$$[\mathrm{block}_{\mathrm{bs}}] \quad \frac{\langle D_V, s\rangle \to_D s',\ \mathsf{updP}(D_P, env_P) \vdash \langle S, s'\rangle \to s''}{env_P \vdash \langle \texttt{begin}\ D_V\ D_P\ S\ \texttt{end}, s\rangle \to s''[\mathrm{DV}(D_V) \longmapsto s]}$$

$$[\mathrm{call}^{\mathrm{rec}}_{\mathrm{bs}}] \quad \frac{env_P \vdash \langle env_P\ p, s\rangle \to s'}{env_P \vdash \langle \texttt{call}\ p, s\rangle \to s'}$$

Ámbito estático para procedimientos

Nuevo Entorno de procedimientos: $\mathbf{Env}_P = \mathbf{Pname} \hookrightarrow \mathbf{Stm} \times \mathbf{Env}_P$

Los procedimientos recuerdan el entorno en que fueron declarados

$$\begin{aligned} \mathsf{updP}(\texttt{proc}\ p\ \texttt{is}\ S\,;\, D_P, env_P) &= \mathsf{updP}(D_P, env_P[p \mapsto (S, env_P)]) \\ \mathsf{updP}(\varepsilon, env_P) &= env_P \end{aligned}$$

Las rellamadas utilizan el entorno de los procedimientos (con o sin recursión)

$$[\mathrm{call}_{\mathrm{bs}}] \quad \frac{env'_P \vdash \langle S, s\rangle \to s'}{env_P \vdash \langle \texttt{call}\ p, s\rangle \to s'}$$

$$[\mathrm{call}^{\mathrm{rec}}_{\mathrm{bs}}] \quad \frac{env'_P[p \mapsto (S, env'_P)] \vdash \langle S, s\rangle \to s'}{env_P \vdash \langle \texttt{call}\ p, s\rangle \to s'}$$

siendo $env_P\ p = (S, env'_P)$

Ámbito estático (variables y procedimientos)

Tratamiento mucho más sofisticado del estado

- Direcciones de memoria: $\mathbf{Loc}$ (es *como* una memoria dinámica (heap))
- new : $\mathbf{Loc} \to \mathbf{Loc}$ (siguiente posición libre)
- Entorno de variables: $env_V \in \mathbf{Env}_V = \mathbf{Var} \to \mathbf{Loc}$
- Memoria: $sto \in \mathbf{Store} = (\mathbf{Loc} \to \mathbb{Z}) * \mathbf{Loc}$

  Notación: Denotaremos $sto(1)$ simplemente con sto, por abuso de notación; y tomaremos $\text{next} = sto(2)$.

Actualización del entorno y de la memoria tras declarar variables

$$[\text{var}_{\text{bs}}] \quad \frac{\langle D_V,\ (env_V[x \mapsto l], sto[l \mapsto v][\text{next} \mapsto \text{new}\, l])\rangle \to_D (env'_V, sto')}{\langle \texttt{var}\ x := a\,;\ D_V,\ (env_V, sto)\rangle \to_D (env'_V, sto')}$$

donde $l = \text{next}$ y $v = \mathcal{A}[\![a]\!](sto \circ env_V)$

$$[\text{none}_{\text{bs}}] \quad \langle \varepsilon,\ (env_V, sto)\rangle \to_D (env_V, sto)$$

Ámbito estático (variables y procedimientos)

Nuevo Entorno de procedimientos: $\mathbf{Env}_P = \mathbf{Pname} \hookrightarrow \mathbf{Stm} \times \mathbf{Env}_V \times \mathbf{Env}_P$

Los procedimientos recuerdan todo el entorno en que fueron declarados

$$\mathsf{updP}(\texttt{proc } p \texttt{ is } S\,;D_P, env_V, env_P) = \mathsf{updP}(D_P, env_V, env_P[p \mapsto (S, env_V, env_P)])$$

$$\mathsf{updP}(\varepsilon, env_V, env_P) = env_P$$

Un Nuevo Sistema de transiciones: $env_V, env_P \vdash \langle S, sto\rangle \rightarrow sto'$

Ámbito estático (variables y procedimientos)

Reglas de los elementos más afectados por las novedades

$$[\text{ass}_{\text{bs}}] \qquad env_V, env_P \vdash \langle x := a, sto\rangle \to sto[env_V\ x \mapsto \mathcal{A}[\![a]\!](sto \circ env_V)]$$

$$[\text{block}_{\text{bs}}] \qquad \frac{\begin{array}{c}\langle D_V, (env_V, sto)\rangle \to_D (env'_V, sto'),\\ env'_V, \mathsf{updP}(D_P, env'_V, env_P) \vdash \langle S, sto'\rangle \to sto''\end{array}}{env_V, env_P \vdash \langle \texttt{begin}\ D_V\ D_P\ S\ \texttt{end}, sto\rangle \to sto''}$$

$$[\text{call}_{\text{bs}}] \qquad \frac{env'_V, env'_P \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{call}\ p, sto\rangle \to sto'}$$

$$[\text{call}^{\text{rec}}_{\text{bs}}] \qquad \frac{env'_V, env'_P[p \mapsto (S, env'_V, env'_P)] \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{call}\ p, sto\rangle \to sto'}$$

siendo $\quad env_P\ p = (S, env'_V, env'_P)$

Ejercicios

Ejercicio 3.14

En particular, la semántica con ámbito estático de la extensión del lenguaje **While** que acabamos de definir, es una semántica alternativa del lenguaje **While** original.

Formular y demostrar la equivalencia entre esta nueva semántica y la de paso largo definida originalmente.

Ejercicio 3.15

Modificar la sintaxis de la declaración de procedimientos para que reciban ahora siempre dos parámetros con paso por valor:

$$D_P ::= \texttt{proc}\ p(x_1, x_2)\ \texttt{is}\ S\,;\, D_P \mid \varepsilon$$
$$S ::= x := a \mid \ldots \mid \texttt{call}\ p(a_1, a_2)$$

Los entornos de procedimiento serán ahora elementos de:

$$\mathbf{Env}_P = \mathbf{Pname} \hookrightarrow \mathbf{Var} \times \mathbf{Var} \times \mathbf{Stm} \times \mathbf{Env}_V \times \mathbf{Env}_P.$$

Modificar la semántica convenientemente, dando en particular nuevas reglas para las llamadas a procedimientos (con o sin recursión).