Semántica Denotacional

David de Frutos Escrig

versión original elaborada por

Yolanda Ortega Mallén

Dpto. de Sistemas Informáticos y Computación

Universidad Complutense de Madrid

Sumario

- Función semántica; valores semnticos y definicin composicional.
- Teoría de puntos fijos. Dominios semánticos. Funciones continuas..
- Equivalencia entre la semántica operacional y la denotacional.

Bibliografía

- Hanne Riis Nielson & Flemming Nielson,
  Semantics with Applications. An Appetizer, Springer, 2007.
  Capítulo 5.
- Glynn Winskel,
  The Formal Semantics of Programming Languages: An Introduction, The Mit Press, 1993.
  Capítulos 5, 8 y 12. (profundización en la Teoría de Dominios)

Función semántica para las sentencias de **While**

$$\mathcal{S}_{\mathrm{d}} : \mathbf{Stm} \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$$

$$\begin{array}{rcl}
\mathcal{S}_{\mathrm{d}}[\![x := a]\!]s & = & s[x \mapsto \mathcal{A}[\![a]\!]s] \\
\mathcal{S}_{\mathrm{d}}[\![\texttt{skip}]\!] & = & \mathsf{id} \\
\mathcal{S}_{\mathrm{d}}[\![S_1; S_2]\!] & = & \mathcal{S}_{\mathrm{d}}[\![S_2]\!] \circ \mathcal{S}_{\mathrm{d}}[\![S_1]\!] \\
\mathcal{S}_{\mathrm{d}}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] & = & \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\mathrm{d}}[\![S_1]\!], \mathcal{S}_{\mathrm{d}}[\![S_2]\!]) \\
\mathcal{S}_{\mathrm{d}}[\![\texttt{while } b \texttt{ do } S]\!] & = & \mathsf{FIX}\ F \\
\text{donde } F\ g & = & \mathsf{cond}(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{\mathrm{d}}[\![S]\!], \mathsf{id})
\end{array}$$

$$\mathsf{cond} : (\mathbf{State} \longrightarrow \mathbf{T}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$$

$$\mathsf{cond}(p, g_1, g_2)\ s = \begin{cases} g_1\ s & \text{si } p\ s = \mathbf{tt} \\ g_2\ s & \text{si } p\ s = \mathbf{ff} \end{cases}$$

$$\mathsf{FIX} : (\mathbf{State} \hookrightarrow \mathbf{State}) \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State}) \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$$

Requisitos para el punto fijo

- Hay funcionales con más de un punto fijo, y
- hay funcionales que no tienen ningún punto fijo.

¿Cómo determinar el punto fijo que nos interesa?

- Imponer requisitos y demostrar que a lo sumo un punto fijo los satisfará, y
- garantizar que todos los funcionales utilizados en la definición de la semántica tienen un punto fijo tal.

FIX $F = g_0 : (\mathbf{State} \hookrightarrow \mathbf{State})$ tal que:

- g_0 es punto fijo: $F\ g_0 = g_0$, y
- Todo punto fijo de F, $(F\ g = g)$, satisface
  $\forall s, s' \in \mathbf{State} \quad g_0\ s = s' \implies g\ s = s'$.

Conjuntos parcialmente ordenados

Orden parcial sobre **State** $\hookrightarrow$ **State**:

$$g_1 \sqsubseteq g_2 \stackrel{\text{def}}{=} \forall s, s' \in \textbf{State}\ g_1\ s = s' \implies g_2\ s = s'$$

Ejercicio 5.8

Demostrar la caracterización alternativa de $\sqsubseteq$ para **State** $\hookrightarrow$ **State**:

$$g_1 \sqsubseteq g_2 \iff \text{grafo}(g_1) \subseteq \text{grafo}(g_2)$$

Conjunto parcialmente ordenado $(D, \sqsubseteq_D)$:

$$\begin{array}{lll} d \sqsubseteq_D d & & \text{(reflexiva)} \\ d_1 \sqsubseteq_D d_2 \ \wedge\ d_2 \sqsubseteq_D d_3 & \implies\ d_1 \sqsubseteq_D d_3 & \text{(transitiva)} \\ d_1 \sqsubseteq_D d_2 \ \wedge\ d_2 \sqsubseteq_D d_1 & \implies\ d_1 = d_2 & \text{(antisimétrica)} \end{array}$$

Interpretación: si $d_1 \sqsubseteq_D d_2$, entonces d_2 contiene más información que d_1.

Elemento mínimo de D: $\forall d' \in D\ \ d \sqsubseteq_D d'$. d no contiene información.

Notación (*bottom*): $\bot_D$.

El conjunto de transformadores de estado

$(\mathbf{State} \hookrightarrow \mathbf{State}, \sqsubseteq)$ es un conjunto parcialmente ordenado.

$$\bot\ s = \text{INDEFINIDO}, \quad \text{para todo } s$$

Requisitos para FIX F

Punto fijo $F(\mathsf{FIX}\ F) = \mathsf{FIX}\ F$;

Mínimo punto fijo; $F\ g = g \implies \mathsf{FIX}\ F \sqsubseteq g$.

¿Cómo garantizar que todo funcional que aparece en la definición de la semántica de **While** tiene un punto fijo mínimo?

Conjuntos parcialmente ordenados completos (*cpo's*)

Consideramos $(D, \sqsubseteq_D)$, $Y \subseteq D$, $d \in D$:

- d es una cota superior de Y si $\forall d' \in Y \;\; d' \sqsubseteq d$;
- d es la cota superior mínima de Y si es cota superior y
  d'' cota superior de $Y \implies d \sqsubseteq d''$.

Una cota superior mínima de Y añade el mínimo posible de información a la contenida en los elementos de Y.

Cotas superiores mínimas (lub's) y Cadenas

Si Y tiene una cota superior mínima (única) se denota por $\bigsqcup Y$.
Y es una cadena en $(D, \sqsubseteq_D)$, si $\forall d_1, d_2 \in Y \;\; d_1 \sqsubseteq d_2 \;\vee\; d_2 \sqsubseteq d_1$.

Ejercicios 5.18 + 5.19

Considerar $(\mathcal{P}(S), \subseteq)$. Demostrar que todo subconjunto de $\mathcal{P}(S)$ tiene una cota superior mínima. Repetir para $(\mathcal{P}(S), \supseteq)$. ¿Qué ocurre con $(\mathcal{P}_{fin}(S), \subseteq)$?

Conjuntos parcialmente ordenados completos

Ejercicio 5.21

Construir un subconjunto de **State** $\hookrightarrow$ **State** que no tenga cotas superiores.

Ejercicio 5.22

Siendo: $$g_n\ s = \begin{cases} s[y \mapsto (s\ x)!, x \mapsto 1] & \text{si } 0 < s\ x \wedge s\ x \leq n \\ \text{INDEFINIDO} & \text{e.c.c.} \end{cases}$$

Demostrar que $\Upsilon_0 = \{g_n \mid n \geq 0\}$ es una cadena.
Caracterizar las cotas superiores de Υ_0 y determinar la mínima de ellas.

Ordenes completos por cadenas y Retículos completos

Orden completo por cadenas (ccpo): toda cadena tiene cota superior mínima.

Retículo completo: todo subconjunto de D tiene cota superior mínima.

Todo ccpo tiene elemento mínimo: $\bot = \bigsqcup \emptyset$

Lema 5.25: $(\textbf{State} \hookrightarrow \textbf{State}, \sqsubseteq)$ es un ccpo.

$$\text{grafo}(\bigsqcup \Upsilon) = \bigcup \{\text{grafo}(g) \mid g \in \Upsilon\}$$

Funciones monótonas

Monotonía y sus propiedades

$f : D \longrightarrow D'$ es monótona entre los ccpo's $(D, \sqsubseteq)$ y $(D', \sqsubseteq')$, si

$$\forall d_1, d_2 \in D \; d_1 \sqsubseteq d_2 \implies f\ d_1 \sqsubseteq' f\ d_2$$

Composición $f : D \longrightarrow D'$ y $f' : D' \longrightarrow D''$ monótonas $\implies$
$f' \circ f : D \longrightarrow D''$ monótona.

Conservación de cadenas $f : D \longrightarrow D'$ monótona y $Y \subseteq D$ cadena $\implies$
$\{f\ d \mid d \in Y\}$ es también cadena en D', y

$$\bigsqcup' \{f\ d \mid d \in Y\} \sqsubseteq' f(\bigsqcup Y)$$

Ejemplo

$(\mathcal{P}(\mathbb{N} \cup \{a\}), \subseteq)$ ccpo y $f : \mathcal{P}(\mathbb{N} \cup \{a\}) \longrightarrow \mathcal{P}(\mathbb{N} \cup \{a\})$:

$$f\ X = \begin{cases} X & \text{si } X \text{ es finito} \\ X \cup \{a\} & \text{si } X \text{ es infinito} \end{cases}$$

f es monótona, pero tomando $Y = \{\ \{0, 1, \ldots, n\} \mid n \geq 0\}$,
$\bigsqcup\{f\ X \mid X \in Y\} = \bigsqcup Y = \mathbb{N} \neq \mathbb{N} \cup \{a\} = f\ \mathbb{N} = f(\bigsqcup Y)$.

Funciones continuas

Continuidad y estricticidad

$f : D \longrightarrow D'$ entre $(D, \sqsubseteq)$ y $(D', \sqsubseteq')$ (ambos ccpo),

Continua: Monótona y para toda cadena no vacía, $Y \subseteq D$,

$$\bigsqcup' \{f\ d \mid d \in Y\} = f(\bigsqcup Y)$$

Estricta: $f\ \bot = \bot$ (o sea, $\bigsqcup' \{f\ d \mid d \in Y\} = f(\bigsqcup Y)$, para toda cadena).

Composición de funciones continuas

$f : D \longrightarrow D'$ y $f' : D' \longrightarrow D''$ continuas $\Longrightarrow$
$f' \circ f : D \longrightarrow D''$ también continua.

Ejercicio 5.36

Demostrar que si f y f' son estrictas, entonces $f' \circ f'$ también es estricta.

Teorema del punto fijo

Teorema 5.37:

- Si $f : D \longrightarrow D$ es montona, existe $\bigsqcup\{f^n \ \bot \mid n \geq 0\} \in D$.
- Si $f : D \longrightarrow D$ es continua tomaremos

$$\text{FIX}\, f = \bigsqcup\{f^n \ \bot \mid n \geq 0\} \in D$$

lo que est justificado pues $FIX\, f$ es el mínimo punto fijo de f:

Punto fijo: $f(FIX\, f) = FIX\, f$.

Ejercicio 5.40: $f\ d \sqsubseteq d \implies \text{FIX}\, f \sqsubseteq d$.

Aplicación de la teoría de puntos fijos

1. Nos aseguramos de manejar órdenes completos por cadenas (ccpo's).
2. Y de que las funciones con las que trabajamos sean continuas.
3. Usamos sus mínimos puntos fijos $FIX\, f$.

Dominio de funciones continuas

Dominio de funciones continuas - Ejercicio 5.41 (Primera parte)

Siendo $(D, \sqsubseteq)$, $(D', \sqsubseteq')$ ccpo's, definimos $(D \longrightarrow D', \sqsubseteq_F)$, tomando sólo las funciones continuas entre D y D', y con

$$f_1 \sqsubseteq_F f_2 \stackrel{\text{def}}{=} \forall d \in D \ \ (f_1\ d) \sqsubseteq (f_2\ d).$$

Demostrar que $(D \longrightarrow D', \sqsubseteq_F)$ es un ccpo.
¿Qué función es su elemento mínimo?

Definición y continuidad del funcional FIX - Ejercicio 5.41 (Segunda parte)

Consideramos el *funcional* $FIX : (D \longrightarrow D) \longrightarrow D$.
Demostrar que FIX es monótono y continuo, o sea que

$$\mathsf{FIX}(\bigsqcup_F \mathcal{F}) = \bigsqcup\{\mathsf{FIX}\, f \mid f \in \mathcal{F}\}.$$

para toda cadena $\mathcal{F} \subseteq (D \longrightarrow D)$ de funciones continuas.

Definición cuidadosa de la semántica denotacional

$$\mathcal{S}_{\mathrm{d}}[\![\texttt{while } b \texttt{ do } S]\!] = \mathsf{FIX}\ F \text{ donde } F\ g = \mathsf{cond}(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{\mathrm{d}}[\![S]\!], \mathsf{id})$$

Hay que demostrar que F es continua. Tenemos $F = F_1 \circ F_2$, con $F_1\ g = \mathsf{cond}(\mathcal{B}[\![b]\!], g, \mathsf{id})$ y $F_2\ g = g \circ \mathcal{S}_{\mathrm{d}}[\![S]\!]$, por lo que basta demostrar que F_1 y F_2 son continuas.

Funcional cond. Su continuidad (separada)

Fijados $g_0 : \mathbf{State} \hookrightarrow \mathbf{State}$, $p : \mathbf{State} \longrightarrow \mathbf{T}$, consideramos $F\ g = \mathsf{cond}(p, g, g_0)$. $F : (\mathbf{State} \hookrightarrow \mathbf{State}) \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$, es continua.
Ejercicio 5.44: Demostrar que $F'\ g = \mathsf{cond}(p, g_0, g)$ también es continua.

Funcional composición $\circ$. Su continuidad (separada)

Fijada $g_0 : \mathbf{State} \hookrightarrow \mathbf{State}$, se define $F\ g = g \circ g_0$.
$F : (\mathbf{State} \hookrightarrow \mathbf{State}) \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$, es continua.
Ejercicio 5.46: Demostrar que $F'\ g = g_0 \circ g$ también es continua.

Proposición 5.47:

La semántica denotacional de While está correctamente definida:

$$\mathcal{S}_{\mathrm{d}} : \mathbf{Stm} \longrightarrow (\mathbf{State} \hookrightarrow \mathbf{State}).$$

Ejercicios: evaluando la semántica y extendiendola

Ejercicio 5.49

Desarrollar la semántica de la sentencia
$\texttt{z} := 0; \texttt{while y} \leq \texttt{x do } (\texttt{z} := \texttt{z} + 1; \texttt{x} := \texttt{x} - \texttt{y})$.

Ejercicio 5.50

Demostrar que $\mathcal{S}_{\mathrm{d}}[\![\texttt{while true do skip}]\!] = \bot$, o sea,
es la función completamente indefinida.

Ejercicio 5.51

Extender el lenguaje **While** con la sentencia $\texttt{repeat } S \texttt{ until } b$ y dar la cláusula semántica que define $\mathcal{S}_{\mathrm{d}}$ para sus aplicaciones.
Demostrar, de manera directa, que $\mathcal{S}_{\mathrm{d}}$ sigue estando bien definida tras la extensión.

Propiedades

Equivalencia semántica (bajo la semántica denotacional)

S_1 y S_2 son equivalentes si $\mathcal{S}_{\mathrm{d}}[\![S_1]\!] = \mathcal{S}_{\mathrm{d}}[\![S_2]\!]$.

Ejercicio 5.53

Demostrar que los siguientes pares de sentencias son equivalentes:

- S ; `skip` y S
- `while` b `do` S y `if` b `then` $(S$; `while` b `do` $S)$ `else skip`
- S_1 ; $(S_2$; $S_3)$ y $(S_1$; $S_2)$; S_3

Ejercicio 5.54

Demostrar que `repeat` S `until` b es semánticamente equivalente a

S ; `while` $\neg b$ `do` S.

Equivalencia con la semántica operacional

Teorema 5.55:

$$\forall S \in \mathbf{Stm}\ \mathcal{S}_{\mathrm{ss}}[\![S]\!] = \mathcal{S}_{\mathrm{d}}[\![S]\!]$$

Lema 5.56: $\forall S \in \mathbf{Stm}\ \ \mathcal{S}_{\mathrm{ss}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{d}}[\![S]\!]$, o equivalentemente

$$\forall S \in \mathbf{Stm}\, \forall s, s' \in \mathbf{State}\ \ \langle S, s\rangle \Rightarrow^* s' \implies \mathcal{S}_{\mathrm{d}}[\![S]\!]s = s'$$

Lema 5.57: $\forall S \in \mathbf{Stm}\ \ \mathcal{S}_{\mathrm{d}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{ss}}[\![S]\!]$

Resumen de la demostración (ambos lemas)

1. Primero utilizar inducción sobre el árbol de derivación
   - Si una sentencia ejecuta un paso en la semántica de paso corto y no termina, entonces el significado en la semántica denotacional no cambia.
   - Si una sentencia ejecuta un paso en la semántica de paso corto y termina, entonces se obtiene el mismo resultado en la semántica denotacional.

   Después utilizar inducción sobre la longitud de la secuencia de derivación.
2. Inducción estructural

   $\mathcal{S}_{\mathrm{ss}}$ satisface versiones "más débiles" de las cláusulas que definen $\mathcal{S}_{\mathrm{d}}$: si $\mathcal{S}_{\mathrm{d}}[\![S]\!] = \Psi(\ldots \mathcal{S}_{\mathrm{d}}[\![S']\!] \ldots)$, vemos que $\mathcal{S}_{\mathrm{ss}}[\![S]\!] \sqsupseteq \Psi(\ldots \mathcal{S}_{\mathrm{ss}}[\![S']\!] \ldots)$.

Ejercicios

Ejercicio 5.59

Extender la demostración del Teorema 5.55 para incluir la sentencia

`repeat` S `until` b.

Ejercicio 5.61

Demostrar de forma directa que $\forall S \in \mathbf{Stm}$ $\mathcal{S}_{\mathrm{bs}}[\![S]\!] = \mathcal{S}_{\mathrm{d}}[\![S]\!]$.