

Tema 5. Semántica denotacional

La diferencia fundamental de esta nueva semántica con las operacionales es que se consigue definir de manera completamente *composicional*.

Función semántica, valores semánticos y definición composicional

La semántica denotacional asocia significado a las instrucciones de **WHILE** a través de la siguiente función:

$$\mathcal{S}_{ds} : \mathbf{Stm} \rightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$$

que se define de la siguiente manera para cada instrucción:

$$\begin{aligned}\mathcal{S}_{ds} \llbracket x := a \rrbracket s &= s [x \mapsto \mathcal{A} \llbracket a \rrbracket s] \\ \mathcal{S}_{ds} \llbracket \mathbf{skip} \rrbracket &= \text{id} \\ \mathcal{S}_{ds} \llbracket S_1 ; S_2 \rrbracket &= \mathcal{S}_{ds} \llbracket S_2 \rrbracket \circ \mathcal{S}_{ds} \llbracket S_1 \rrbracket \\ \mathcal{S}_{ds} \llbracket \mathbf{if } b \mathbf{ then } S_1 \mathbf{ else } S_2 \rrbracket &= \text{cond} (\mathcal{B} \llbracket b \rrbracket, \mathcal{S}_{ds} \llbracket S_1 \rrbracket, \mathcal{S}_{ds} \llbracket S_2 \rrbracket) \\ \mathcal{S}_{ds} \llbracket \mathbf{while } b \mathbf{ do } S \rrbracket &= \text{FIX } F \\ &\text{donde } F \ g = \text{cond} (\mathcal{B} \llbracket b \rrbracket, g \circ \mathcal{S}_{ds} \llbracket S \rrbracket, \text{id})\end{aligned}$$

Como se puede observar, tenemos dos funciones auxiliares `cond` y `FIX` que debemos definir a su vez. Empezamos con `cond` que tiene como tipo

$$\begin{aligned}\text{cond} : (\mathbf{State} \rightarrow \mathbf{T}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \\ \rightarrow (\mathbf{State} \hookrightarrow \mathbf{State})\end{aligned}$$

y se define como

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

Por otro lado, el tipo de `FIX` será

$$\text{FIX} : ((\mathbf{State} \hookrightarrow \mathbf{State}) \rightarrow (\mathbf{State} \hookrightarrow \mathbf{State})) \rightarrow (\mathbf{State} \hookrightarrow \mathbf{State})$$

y su definición será

$$\text{FIX } F = g \text{ tal que } F g = g.$$

Sin embargo, no todas las funciones tienen punto fijo o no es único por lo que tendremos que hacer un estudio más detallado para refinar la definición.

Teoría de puntos fijos. Dominios semánticos. Funciones continuas

Requerimientos para el punto fijo

Al ejecutar un bucle del lenguaje **WHILE** se nos presentan tres posibles casos:

- El bucle termina correctamente para cualquier posible estado
- El bucle cicla *localmente*, es decir, existen una cantidad *finita* de estados en los que el bucle no termina.
- El bucle cicla *globalmente*, es decir, existen una cantidad *infinita* de estados en los que el bucle no termina.

Los dos primeros casos no dan problemas ya que el punto fijo es único. Sin embargo, en el tercer caso podemos tener múltiples funciones que son punto fijo. Por esta razón, debemos encontrar la función g_0 punto fijo que cumpla la siguiente propiedad:

$$\forall g \text{ punto fijo de } F \Leftrightarrow g_0 s = s' \Rightarrow g s = s', \text{ pero no al revés.}$$

O en otras palabras, que g_0 contenga la *mínima cantidad de información* posible (lo que la hace la más general posible).

Conjuntos parcialmente ordenados

Definición: (Orden parcial)

Sean $g_1, g_2 \in \mathbf{State} \hookrightarrow \mathbf{State}$. Decimos que $g_1 \sqsubseteq g_2$ si, y sólo si,

$$\forall s, s' \in \mathbf{State}, g_1 s = s' \Rightarrow g_2 s = s'$$

Es decir, toda la información contenida en g_1 , también lo está en g_2 .

Observación:

El conjunto $\mathbf{State} \hookrightarrow \mathbf{State}$ junto con \sqsubseteq es un ejemplo de *conjunto parcialmente ordenado* ya que se cumplen las propiedades *reflexiva*, *transitiva* y *antisimétrica*.

Definición: (Elemento mínimo)

Llamaremos *elemento mínimo*, denotado por \perp , de un conjunto parcialmente ordenado (D, \sqsubseteq) a aquel que cumpla que

$$\forall d \in D, \perp \sqsubseteq d.$$

Decimos que \perp *no contiene información*.

Proposición: (Unicidad del elemento mínimo)

Si un (D, \sqsubseteq) tiene elemento mínimo, entonces este es único.

Lema: (Elemento mínimo de $\text{State} \hookrightarrow \text{State}$)

El conjunto $(\text{State} \hookrightarrow \text{State}, \sqsubseteq)$ es parcialmente ordenado y la función definida como

$$\perp s = \text{undef}, \forall s \in \text{State}$$

es su elemento mínimo.

Con toda esta nueva información podemos actualizar la definición de **FIX** para que sea correcta:

- $\text{FIX } F$ es un punto fijo, es decir, $F (\text{FIX } F) = \text{FIX } F$.
- $\text{FIX } F$ es el punto fijo *mínimo*, es decir,

$$\text{si } F g = g \Rightarrow \text{FIX } F \sqsubseteq g.$$

Conjuntos parcialmente ordenados completos

Hasta este momento hemos conseguido obtener la unicidad del punto fijo del funcional, sin embargo, todavía no sabemos si existe tal punto fijo. Por esta razón, necesitamos restringir todavía más el conjunto de funcionales con el que estamos tratando.

Definición: (Cotas superiores)

Sea (D, \sqsubseteq) un conjunto parcialmente ordenado, $Y \subset D$ y $d \in D$. Diremos que d es un *cota superior* de Y si

$$\forall d' \in Y, d' \sqsubseteq d$$

Y será la cota superior *mínima* si

$$\forall d'' \text{ cota superior de } Y \Rightarrow d \sqsubseteq d''.$$

Esta cota superior mínima (*lub*) se denota, si existe, como $\sqcup Y$.

Definición: (Cadenas)

En las condiciones anteriores diremos que Y es una *cadena* si

$$\forall d_1, d_2 \in Y, d_1 \sqsubseteq d_2 \text{ ó } d_2 \sqsubseteq d_1.$$

Definición: (Orden y retículo completo por cadenas)

- Decimos que (D, \sqsubseteq) es un *orden completo por cadenas* (*ccpo*) si para toda cadena $Y \subset D$, existe una cota superior mínima $\sqcup Y$.
- El conjunto D será un *retículo completo* si todo $Y \subset D$ tiene $\sqcup Y$.

Proposición: (Elemento mínimo de un ccpo)

Si (D, \sqsubseteq) es un *ccpo*, entonces existe un elemento mínimo tal que

$$\perp = \sqcup \emptyset.$$

Proposición:

El conjunto $(\mathbf{State} \hookrightarrow \mathbf{State}, \sqsubseteq)$ es un *ccpo*.

Funciones continuas

En esta sección trataremos con *ccpos*, por tanto, cada vez que aparezca un conjunto D, D' , etc. asumiremos que es tal y sus operadores de orden serán, respectivamente, $\sqsubseteq, \sqsubseteq'$, etc.

Definición: (Monotonía)

Diremos que una función $f : D \rightarrow D'$ es monótona si

$$\forall d_1, d_2 \in D : d_1 \sqsubseteq d_2 \Rightarrow f d_1 \sqsubseteq' f d_2.$$

Proposición: (Propiedades monotonía)

Sea $f : D \rightarrow D'$ y $f' : D' \rightarrow D''$ ambas monótonas, entonces se cumple que

- La composición también es monótona:

$$f' \circ f : D \rightarrow D'' \text{ es monótona.}$$

- Se conservan las cadenas. Sea $Y \subset D$ una cadena, entonces $f Y = \{f d \mid d \in Y\}$ es una cadena de D' y

$$\bigsqcup' f Y \sqsubseteq' f \left(\bigsqcup Y \right).$$

Observación:

A pesar de lo que se pueda pensar, la última propiedad de la monotonía no nos permite decir que el elemento mínimo de aplicar la f a todos los puntos de Y sea *igual* a aplicar f al elemento mínimo de Y . Solo nos interesaran las funciones que *sí* lo conserven. De aquí la siguiente definición.

Definición: (Continuidad)

Diremos que $f : D \rightarrow D'$ es *continua* si es monótona y para toda cadena no vacía $Y \subset D$ se cumple que

$$\bigsqcup' f Y = f \left(\bigsqcup Y \right).$$

Si además se cumple para la vacía, es decir, $\perp = f \perp$ diremos que es *estricta*.

Proposición: (Composición continua)

Sea $f : D \rightarrow D'$ y $f' : D' \rightarrow D''$ ambas continuas, entonces

$$f' \circ f : D \rightarrow D''$$

también es continua.

Teorema: (Caracterización del elemento mínimo de f)

Sea $f : D \rightarrow D$ una función continua tal que D (ccpo) tiene elemento mínimo \perp . Entonces

$$\text{FIX } f = \bigsqcup \{f^n \perp \mid n \in \mathbb{N}\}$$

es un elemento de D y, de hecho, es el punto fijo mínimo de f .

Observaciones

- En el anterior teorema hemos usado como notación $f^0 = \text{id}$ y $f^{n+1} = f \circ f^n$ para $n > 0$.
- Cabe destacar que, como $f^n \perp \sqsubseteq f^{n+1} \perp$, $\forall n \in \mathbb{N}$ (al ser continua) el conjunto $\{f^n \perp\}$ es realmente una cadena.

Resumen sección

Para concluir la sección enumeraré las propiedades que deseamos para manejar correctamente los puntos fijos en nuestra semántica:

- Será necesario que el dominio en el que trabajamos sea un *ccpo*.
- Las funciones que tratemos tendrán que ser *continuas* en ese *ccpo*.
- Usaremos los *puntos fijos mínimos* de estas funciones.

Corrección de la definición

Tras construir toda la estructura auxiliar, volvemos al propósito inicial del capítulo, definir *composicionalmente* y *correctamente* la semántica denotacional. Es por esto que ahora debemos asegurarnos que nuestro dominio sea un *ccpo* y que las funciones que utilizamos en la definición de la semántica sean continuas.

Proposición: (Dominio de funciones continuas)

Sean D y D' dos *ccpo*'s, si definimos $(D \rightarrow D', \sqsubseteq_F)$ tomando solo las funciones continuas entre D y D' y con

$$f_1 \sqsubseteq_F f_2 := \forall d \in D : (f_1 d) \sqsubseteq (f_2 d)$$

tenemos que $(D \rightarrow D', \sqsubseteq_F)$ es un *ccpo*.

Proposición: (Continuidad de FIX)

El funcional $\text{FIX} : (D \rightarrow D) \rightarrow D$ es continuo.

Proposición: (Continuidad del condicional)

La función $F : (\text{State} \hookrightarrow \text{State}) \rightarrow (\text{State} \hookrightarrow \text{State})$ definida como:

$$F g = \text{cond}(p, g, g_0), \quad g_0 \in \text{State} \hookrightarrow \text{State}$$

es continua.

Proposición: (Continuidad funcional composición)

La función $F : (\text{State} \hookrightarrow \text{State}) \rightarrow (\text{State} \hookrightarrow \text{State})$ definida como:

$$F g = g \circ g_0, \quad g_0 \in \text{State} \hookrightarrow \text{State}$$

es continua.

Y con esto tenemos el resultado que buscábamos.

Proposición: (Corrección semántica denotacional)

La función semántica \mathcal{S}_{ds} es total, es decir, estar correctamente definida.

Propiedades de la semántica

Definición: (Equivalencia semántica)

Decimos que dos sentencias S_1 y S_2 son semánticamente equivalentes si se cumple que

$$\mathcal{S}_{ds}[[S_1]] = \mathcal{S}_{ds}[[S_2]].$$

Equivalencia entre la semántica operacional y la denotacional

Teorema: (Equivalencia entre semántica operacional y denotacional)

Para toda sentencia del lenguaje **WHILE** tenemos que

$$\mathcal{S}_{ds}[[S]] = \mathcal{S}_{sos}[[S]].$$