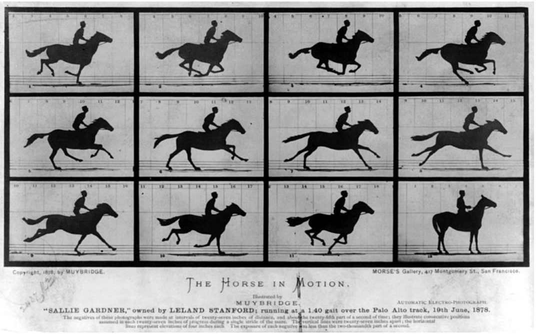
Nociones Básicas de Sémantica: Semántica Operacional

Análisis de Lenguajes de Programación

Mauro Jaskelioff 23/08/2019



Semántica de un lenguaje

¿Cómo saber si dos programas son equivalentes?

```
int add1 (int x, int y)
{ return (x + y);
}

int add2 (int x, int y)
{ return (y + x);
}
```

Semántica de un lenguaje

¿Cómo saber si dos programas son equivalentes?

$$\texttt{(if b then p else q); r} \stackrel{?}{=} \texttt{if b then (p;r) else (q;r)}$$

$$\texttt{p; if b then q else r} \stackrel{?}{=} \texttt{if b then (p;q) else (p;r)}$$

$$\texttt{p; if b then q else r} \stackrel{?}{=} \texttt{if (p; b) then q else r}$$

$$\texttt{b/0} \stackrel{?}{=} \texttt{*NULL}$$

Semántica Formal

- Para saber si dos programas son equivalentes necesitamos:
  - Dar una descripción **precisa** de los programas.
  - Definir que es lo que se puede observar de un programa.
  - Elegir una noción apropiada de equivalencia.
- Todo esto es difícil, pero vale la pena.

Beneficios de la semántica formal

- Implementación: compiladores, optimizaciones correctas, análisis estático.
- Verificación: Soporte para razonar acerca de las propiedades de los programas.
- Diseño de Lenguajes: Resolver interacciones sutiles entre las características del lenguaje.

Enfoques para Semántica

Hay varios enfoques para dar semántica a un lenguaje. Los más comunes son:

- **Operacional**: El significado de un programa está dado por los pasos de computación que el programa realiza cuando se ejecuta.
- **Denotacional**: Dar un objeto matemático D y definir una función de interpretación

$$[\![-]\!] : T \to D$$

- **Axiomático**: Dar leyes sobre los programas. El significado es todo lo que se puede derivar de esas leyes.
  - Generalmente se usa sólo en lenguajes imperativos (ej: *Hoare* Logic.)

Semántica Operacional

- Definimos los pasos que un programa da durante su evaluación.
- Las propiedades surgen del análisis de esta ejecución.
- Hay dos formas usuales de hacer esto:
  - Paso chico (*small step*): La evaluación de las expresiones se hace paso por paso.
  - Paso grande (*big step*): Los pasos intermedios se ignoran y se da directamente el resultado.

Valores en Semántica Operacional

- Definimos una semántica operacional para un lenguaje.
- Tomamos el fragmento booleano del lenguaje visto en la clase anterior y definimos los términos $\mathcal{T}$:

$$
\begin{aligned}
t ::= {} & \mathtt{T} \\
\mid {} & \mathtt{F} \\
\mid {} & \mathtt{if}\ t\ \mathtt{then}\ t\ \mathtt{else}\ t
\end{aligned}
$$

- Los valores $\mathcal{V}$ son **un subconjunto de los términos**.

$$v ::= \mathtt{T} \mid \mathtt{F}$$

- Notación: Usamos la metavariable t para términos y v para valores.

Relación de Evaluación de Paso Grande

- Definimos la relación de evaluación $\Downarrow \subseteq \mathcal{T} \times \mathcal{V}$

$$\frac{}{v \Downarrow v} \qquad \text{(B-Val)}$$

$$\frac{t_1 \Downarrow \mathtt{T} \qquad t_2 \Downarrow v}{\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \Downarrow v} \qquad \text{(B-IfTrue)}$$

$$\frac{t_1 \Downarrow \mathtt{F} \qquad t_3 \Downarrow v}{\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \Downarrow v} \qquad \text{(B-IfFalse)}$$

- $t \Downarrow v$ se lee "t evalúa al valor v".
- La relación $\Downarrow$ es la menor relacion que satisface las 3 reglas de arriba.

Cómo leer las reglas

- El axioma

$$\frac{}{v \Downarrow v} \qquad \text{(B-VAL)}$$

  dice que para todo valor v, vale que $v \Downarrow v$.

- La regla

$$\frac{t_1 \Downarrow \mathtt{T} \qquad t_2 \Downarrow v}{\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \Downarrow v} \qquad \text{(B-IFTRUE)}$$

  nos dice que para todo término t_1, t_2, y t_3
  si es vale que $t_1 \Downarrow \mathtt{T}$
  y vale que $t_2 \Downarrow v$
  entonces vale $\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \Downarrow v$

Las reglas son esquemas

- En las reglas, t_1, t_2, t_3 y v son **metavariables**.
- Cada regla es un esquema para una cantidad (posiblemente infinita) de reglas como:

$$\frac{}{\mathtt{T} \Downarrow \mathtt{T}}$$

$$\frac{\mathtt{T} \Downarrow \mathtt{F} \qquad \mathtt{F} \Downarrow \mathtt{F}}{\mathtt{if}\ \mathtt{T}\ \mathtt{then}\ \mathtt{T}\ \mathtt{else}\ \mathtt{F} \Downarrow \mathtt{F}}$$

- Esta última regla es tonta, pero **válida**. ¿Qué significa esto?

Relación de evaluación, más formalmente

- Una **instancia** de una regla de inferencia se obtiene reemplazando consistentemente cada metavariable por el mismo término en la conclusión como en las premisas.
- Una regla **satisface** una relación, si cada instancia de la regla
  - La conclusión está en la relación, o bien,
  - Alguna de las premisas no está en la relación.
- La relación de evaluacion $\Downarrow$ es la menor relación binaria sobre términos que satisface las reglas. (B-IfTrue, B-IfFalse, y B-Val.)
- Cuando (t, v) está en la relación de evaluación decimos que el **juicio de evaluación** $t \Downarrow v$ es **derivable**.

Derivabilidad

- Que $\Downarrow$ sea la menor relación nos dice que $t \Downarrow v$ es derivable si y sólo si sigue de las reglas:
  - es el axioma $\text{B-V}\textsc{al}$,
  - o bien es la conclusión de $\text{B-I}\textsc{f}\text{T}\textsc{rue}$ o $\text{B-I}\textsc{f}\text{F}\textsc{alse}$ con una premisa derivable.
- La derivabilidad de un juicio se justifica con un **árbol de derivación**.
  - Las hojas del árbol son etiquetadas con instancias de $\text{B-V}\textsc{al}$.
  - Los nodos internos del árbol son etiquetados con instancias de $\text{B-I}\textsc{f}\text{T}\textsc{rue}$ o de $\text{B-I}\textsc{f}\text{F}\textsc{alse}$.

Ejemplo de Árbol de Derivación

Probamos que

$$\texttt{if}\ (\texttt{if F then T else T})\ \texttt{then F else T} \Downarrow \texttt{F}$$

$$\dfrac{\dfrac{\dfrac{}{\texttt{F} \Downarrow \texttt{F}}\ (\textsc{B-Val}) \qquad \dfrac{}{\texttt{T} \Downarrow \texttt{T}}\ (\textsc{B-Val})}{\texttt{if F then T else T} \Downarrow \texttt{T}}\ (\textsc{B-IfFalse}) \qquad \dfrac{}{\texttt{F} \Downarrow \texttt{F}}\ (\textsc{B-Val})}{\texttt{if}\ (\texttt{if F then T else T})\ \texttt{then F else T} \Downarrow \texttt{F}}\ (\textsc{B-IfTrue})$$

Ejercicio: Probar que

$$\texttt{if F then T else}\ (\texttt{if T then F else T}) \Downarrow \texttt{F}$$

Relación de Evaluación de Paso Chico

- La semántica se da por una relación entre "estados" de una máquina abstracta.
- Definimos la relación de evaluación $\rightarrow \subseteq \mathcal{T} \times \mathcal{T}$

$$\texttt{if T then } t_2 \texttt{ else } t_3 \rightarrow t_2 \quad (\text{E-IfTrue})$$

$$\texttt{if F then } t_2 \texttt{ else } t_3 \rightarrow t_3 \quad (\text{E-IfFalse})$$

$$\frac{t_1 \rightarrow t_1{}'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \rightarrow \texttt{if } t_1{}' \texttt{ then } t_2 \texttt{ else } t_3} \; (\text{E-If})$$

- $t \rightarrow t'$ se lee "t evalúa a t' en un solo paso".

Acerca de la relación de Evaluación de Paso Chico

- Notar que `T` y `F` no evalúan a nada.
- Las reglas a veces se dividen en reglas de computación (E-IfTrue y E-IfFalse) y reglas de congruencia (E-If.)
- La relación de evaluación fija una **estrategia de evaluación**.
  - En `if` t_1 `then` t_2 `else` t_3, se debe evaluar t_1 antes de evaluar t_2 o t_3.

Ejercicio

Modificar la relación de evaluación

$$\texttt{if T then } t_2 \texttt{ else } t_3 \rightarrow t_2 \qquad (\text{E-IfTrue})$$

$$\texttt{if F then } t_2 \texttt{ else } t_3 \rightarrow t_3 \qquad (\text{E-IfFalse})$$

$$\frac{t_1 \rightarrow t_1'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \rightarrow \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3} \; (\text{E-If})$$

para que en `if` t_1 `then` t_2 `else` t_3 se evalúe primero t_2, luego t_3 y finalmente t_1.

Recordar que usamos la metavariable v para representar valores (en este caso `T` o `F`.)

Ejemplo: Árbol de Derivación

Sea

$$
\begin{aligned}
s &= \texttt{if T then F else T} \\
t &= \texttt{if } s \texttt{ then T else T} \\
u &= \texttt{if F then T else T}
\end{aligned}
$$

entonces podemos justificar que

$$\texttt{if } t \texttt{ then F else F} \to \texttt{if } u \texttt{ then F else F}$$

con el árbol

$$
\dfrac{\dfrac{\dfrac{}{s \to \texttt{F}}\ (\text{E-IfTrue})}{t \to u}\ (\text{E-If})}{\texttt{if } t \texttt{ then F else F} \to \texttt{if } u \texttt{ then F else F}}\quad (\text{E-If})
$$

Inducción sobre una derivación

- Sea P un predicado sobre una derivación de juicios de evaluación.
- Si, para cada derivación $\mathcal{D}$,
  - dado $P(C)$, para todas las subderivaciones inmediatas C,
  - podemos probar $P(\mathcal{D})$
- entonces $P(\mathcal{D})$ vale para todo $\mathcal{D}$.

Determinismo de la evaluación de un paso (chico)

Teorema

Si $t \to t'$ *y* $t \to t''$*, entonces* $t' = t''$*.*

- Para probarlo usamos **inducción sobre la derivación** $t \to t'$.
- Si la última regla utilizada es E-IfTrue, entonces t tiene la forma `if` t_1 `then` t_2 `else` t_3, con $t_1 = \mathtt{T}$. Pero entonces,
  - la última regla no puede ser E-IfFalse ya que no podemos tener $t_1 = \mathtt{T}$ y $t_1 = \mathtt{F}$.
  - Tampoco puede ser E-If, ya que esa regla pide que $t_1 \to {t_1}'$ para algún ${t_1}'$, pero `T` no evalúa a ningún término.
- Ejercicio: Terminar la prueba.

Forma Normal

- Un término t está en **forma normal** si no se le puede aplicar ninguna regla de evaluación.
- O sea, t está en forma normal si no existe t' tal que $t \to t'$.
- Para nuestro lenguaje simple las formas normales son `T` y `F` (los valores).
- **Teorema**: Todo valor está en forma normal.
- En general el converso no vale (por ej. errores de ejecución), pero para nuestro lenguaje tenemos el siguiente teorema:
- Si t está en forma normal, entonces t es un valor.
  - Prueba: por inducción estructural sobre t en el contrarecíproco.

Evaluación de pasos múltiples

- La relación de **pasos múltiples** $\rightarrow^*$ es la clausura reflexivo-transitiva de $\rightarrow$.
- Es decir es la menor relación tal que

$$\frac{t \rightarrow t'}{t \rightarrow^* t'} \qquad \frac{}{t \rightarrow^* t} \qquad \frac{t \rightarrow^* t' \qquad t' \rightarrow^* t''}{t \rightarrow^* t''}$$

Teorema (Unicidad de Formas Normales)

Si $t \rightarrow^* u$ *y* $t \rightarrow^* u'$*, donde* u *y* u' *son formas normales, entonces* $u = u'$.

Teorema (La evaluación termina)

Para todo término t *hay una forma normal* t' *tal que* $t \rightarrow^* t'$.

Más resultados

- La evaluación de paso grande tiene propiedades similares

Teorema (Determinismo)

Si $t \Downarrow v$ *y* $t \Downarrow v'$ *entonces* $v = v'$.

Teorema (Terminación)

Para todo término t, *existe* v *tale que* $t \Downarrow v$.

- Relación entre las dos semánticas

Teorema (Equivalencia de paso grande y chico)

Para todo término t *y valor* v, $t \Downarrow v$ *sii* $t \to^* v$.

Semántica del Lenguaje de Expresiones Aritméticas

- Trabajamos ahora con el lenguaje de expresiones aritméticas completo.

$$
\begin{aligned}
t ::= & \ \mathtt{T} \mid \mathtt{F} \mid \mathtt{if}\ t\ \mathtt{then}\ t\ \mathtt{else}\ t \\
\mid & \ 0 \mid \mathtt{succ}\ t \mid \mathtt{pred}\ t \mid \mathtt{iszero}\ t
\end{aligned}
$$

- Para definir los valores agregamos una nueva categoría sintáctica de valores numéricos:

$$
\begin{aligned}
v \ & ::= \mathtt{T} \mid \mathtt{F} \mid nv \\
nv & ::= 0 \mid \mathtt{succ}\ nv
\end{aligned}
$$

- Vamos a definir la relación de evaluación para el lenguaje completo, agregando reglas a las existentes.

Nuevas reglas de evaluación de paso chico

$$\frac{t_1 \to t_1{}'}{\texttt{succ}\ t_1 \to \texttt{succ}\ t_1{}'} \qquad (\textsc{E-Succ})$$

$$\texttt{pred}\ 0 \to 0 \qquad (\textsc{E-PredZero})$$

$$\texttt{pred}\ (\texttt{succ}\ nv_1) \to nv_1 \qquad (\textsc{E-PredSucc})$$

$$\frac{t_1 \to t_1{}'}{\texttt{pred}\ t_1 \to \texttt{pred}\ t_1{}'} \qquad (\textsc{E-Pred})$$

$$\texttt{iszero}\ 0 \to \texttt{T} \qquad (\textsc{E-isZeroZero})$$

$$\texttt{iszero}\ (\texttt{succ}\ nv_1) \to \texttt{F} \qquad (\textsc{E-IsZeroSucc})$$

$$\frac{t_1 \to t_1{}'}{\texttt{iszero}\ t_1 \to \texttt{iszero}\ t_1{}'} \qquad (\textsc{E-IsZero})$$

Acerca de las nuevas reglas

- Notar el rol que juega la categoría sintáctica nv en la estrategia de evaluación.
- Por ejemplo, no se puede usar $\text{E-P}{\scriptsize\text{RED}}\text{S}{\scriptsize\text{UCC}}$ para concluir que $\texttt{pred}\ (\texttt{succ}\ (\texttt{pred}\ 0)) \to \texttt{pred}\ 0$.
- Notar que términos como $\texttt{succ}\ \texttt{F}$ son formas normales, pero **no son valores**.
- Si t es una forma normal pero no es un valor, decimos que t está **atascado** (stuck).
- Un término atascado se puede pensar como error de run-time. No se puede seguir la ejecución porque se llegó a un estado sin sentido.

Ejercicios

- Probar que la relación de evaluación es determínistica. O sea que si $t \to t'$, y $t \to t''$, entonces $t' = t''$.
- Probar que todo valor es una forma normal.

Resumen

- Diferentes formas de especificar la semántica de lenguajes.
- Semántica operacional de paso grande y de paso chico
  - Valores,relación de evaluación, árbol de derivación, forma normal, términos atascados.
  - Propiedades: determinismo, valores como forma normal, unicidad de formas normales, terminación.

- **Referencias:** Types and Programming Languages. Benjamin Pierce. Capítulo 3.