Denotációs szemantika

Dr. Horpácsi Dániel
ELTE Informatikai Kar

2023-2024-2

Leíró szemantika,

avagy a matematikai szemantika

Emlékeztető: formális szemantikadefiníciók

- A programozási nyelvek szemantikája általában informálisan kerül leírásra, továbbá praktikusan a nyelv fordítóprogramja definiálja
- Ezek közül egyiket sem tekintjük formális definíciónak, nem használhatóak bizonyításhoz

Megoldás: használjuk a jó öreg matematikát és logikát!

Két alapvető megközelítés:

- Operációs (műveleti) – relációval
  - Strukturális

    Minden lépés modellezése
  - Természetes

    A kezdő- és végállapotok közötti reláció felállítása

- **Denotációs (leíró)** – kompozicionális függvénnyel

  Minden szintaktikus elemhez hozzárendeli a jelentését

- Az alapvető imperatív programkonstrukciók (a While nyelv) műveleti szemantikáját már áttekintettük
- Az operációs szemantika megadta, hogyan értékelődnek ki a kifejezések, illetve hogyan hajtódnak végre programok lépésről-lépésre

- Most egy még absztraktabb megközelítést tekintünk át, amikor a szemantikadefiníciót **denotációs függvények** adják meg
- Nem azt adjuk meg, hogyan működik, hanem hogy **mi a hatása**

- **Matematikai objektumokkal jellemezzük egy-egy konstrukció jelentését** (pl. a számliterálokat az értékük jellemzi, a programok jelentését pedig parciális függvényekkel írjuk le)

Emlékeztető: az absztrakt szintaxis

Szintaktikus kategóriák és azok definíciói, amelyek a nyelv absztrakt szintaxisfáit definiálják

Szintaktikus kategóriák

$$
\begin{array}{lll l}
n & \in & \mathrm{Num} & \text{(számliterálok)} \\
x & \in & \mathrm{Var} & \text{(változószimbólumok)} \\
a & \in & \mathrm{Aexp} & \text{(aritmetikai kifejezések)} \\
b & \in & \mathrm{Bexp} & \text{(logikai kifejezések)} \\
S & \in & \mathrm{Stm} & \text{(utasítások)}
\end{array}
$$

Produkciós szabályok

$$
\begin{array}{lll}
a & ::= & n \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid \text{-}a \\
b & ::= & \mathbf{true} \mid \mathbf{false} \mid a_1 = a_2 \mid a_1 < a_2 \mid \neg b \mid b_1 \wedge b_2 \\
S & ::= & \mathbf{skip} \mid x := a \mid S_1 ; S_2 \mid \mathbf{if}\ b\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2 \mid \mathbf{while}\ b\ \mathbf{do}\ S
\end{array}
$$

Denotációs szemantika: az alapötlet

- Az operációs szemantikában induktívan definiálunk relációkat ($\Rightarrow$ és $\rightarrow$), amelyek az úgynevezett konfigurációk (programállapotok) között adják meg a lehetséges átmeneteket

- Amikor egy S utasítás hatásáról beszélünk, akkor a környezetére gyakorolt hatása az érdekes, tehát a mi esetünkben az, hogy hogyan változtatja meg az állapotot: $\{(s, s') \mid \langle S, s\rangle \rightarrow s'\}$

  $\{(s, s') \mid \langle S, s\rangle \Rightarrow^* s'\}$

S mindig meghatároz egy parciális függvényt az állapotok között: ez az S **denotációja**.

Denotációs szemantika: az alapötlet

- A denotációs szemantikában minden szintaktikus kategóriához (nyelvi konstrukcióhoz, résznyelvhez) megadunk egy szemantikus domaint (halmazt)
- Majd a szintaktikus domain elemeihez a szemantikus domain elemeit rendeljük a deotációs függvénnyel

A denotációs függvények...

- **Teljesek**: a szemantikus függvényeket a szintaktikus kategóriák minden mintájára megadjuk
- **Kompozicionálisak**: az összetett kifejezések jelentését a részkifejezések jelentéséből komponáljuk
  (vö. “vasrúd” és “vasút”)

Hogyan definiáljuk az imperatív konstrukciókra?

Az utasítások jelentését egy parciális függvénnyel karakterizáljuk, amely állapotokat képez állapotokra. Az operációs szemantikával ellentétben a szemantikus függvény definíciója nem egy extra lépés, hanem maga a szemantikadefiníció.

$$\mathcal{S}_{ds} : \mathrm{Stm} \to (\mathrm{State} \hookrightarrow \mathrm{State})$$

Minden S utasításhoz tartozni fog egy parciális függvény:

$$\mathcal{S}_{ds}[\![S]\!] \in \mathrm{State} \hookrightarrow \mathrm{State}$$

- Hogyan definiálnánk a szekvenciát?
- Hogyan definiálnánk az elágazást?
- Hogyan definiálnánk a ciklust?

$$\mathcal{S}_{ds} : \mathrm{Stm} \to (\mathrm{State} \hookrightarrow \mathrm{State})$$

Az operációs szemantikával ellentétben a szemantikus függvény megadása nem kiegészítő lépés, hanem maga a szemantikadefiníció.

A While leíró szemantikája

$$\begin{aligned}
\mathcal{S}_{ds}[\![\textbf{skip}]\!] &= id_{\mathrm{State}} \\
\mathcal{S}_{ds}[\![x := a]\!]s &= s[x \mapsto \mathcal{A}[\![a]\!]s] \\
\mathcal{S}_{ds}[\![S_1; S_2]\!] &= ? \\
\mathcal{S}_{ds}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!] &= ? \\
\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] &= ?
\end{aligned}$$

$$\mathcal{S}_{ds} : \mathrm{Stm} \to (\mathrm{State} \hookrightarrow \mathrm{State})$$

Az operációs szemantikával ellentétben a szemantikus függvény megadása nem kiegészítő lépés, hanem maga a szemantikadefiníció.

A While leíró szemantikája

$$\begin{aligned}
\mathcal{S}_{ds}[\![\textbf{skip}]\!] &= id_{\mathrm{State}} \\
\mathcal{S}_{ds}[\![x := a]\!]s &= s[x \mapsto \mathcal{A}[\![a]\!]s] \\
\mathcal{S}_{ds}[\![S_1; S_2]\!] &= \mathcal{S}_{ds}[\![S_2]\!] \circ \mathcal{S}_{ds}[\![S_1]\!] \\
\mathcal{S}_{ds}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!] &= ? \\
\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] &= ?
\end{aligned}$$

Az esetfüggő szemantika megadására bevezetünk egy segédfüggvényt. A *cond* függvény esetszétválasztást definiál:

$$\mathit{cond} : (\mathrm{State} \to \mathrm{Boolean}) \times (\mathrm{State} \hookrightarrow \mathrm{State}) \times (\mathrm{State} \hookrightarrow \mathrm{State}) \to (\mathrm{State} \hookrightarrow \mathrm{State})$$

Az értelmezési tartománya egy hármas (azt mondjuk, három paramétere van): az első a feltétel, a másik kettő az ágak. Az értékkészlete a szokásos állapotok közötti parciális függvény.

$$\mathit{cond}(p, g_1, g_2)s = \begin{cases} g_1\ s & \text{ha} \quad p\ s = tt \\ g_2\ s & \text{ha} \quad p\ s = ff \end{cases}$$

A *cond* az állapot függvényében használja az egyik vagy másik szemantikadefiníciót: ha a feltétel az állapotban igaz, akkor az első ágat használja, ha hamis, akkor a másodikat.

$$\mathcal{S}_{ds} : \mathrm{Stm} \to (\mathrm{State} \hookrightarrow \mathrm{State})$$

Az operációs szemantikával ellentétben a szemantikus függvény megadása nem kiegészítő lépés, hanem maga a szemantikadefiníció.

A While leíró szemantikája

$$\begin{aligned}
\mathcal{S}_{ds}[\![\textbf{skip}]\!] &= id_{\mathrm{State}} \\
\mathcal{S}_{ds}[\![x := a]\!]s &= s[x \mapsto \mathcal{A}[\![a]\!]s] \\
\mathcal{S}_{ds}[\![S_1; S_2]\!] &= \mathcal{S}_{ds}[\![S_2]\!] \circ \mathcal{S}_{ds}[\![S_1]\!] \\
\mathcal{S}_{ds}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!] &= cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![S_1]\!], \mathcal{S}_{ds}[\![S_2]\!]) \\
\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] &= ?
\end{aligned}$$

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

Az operációs szemantikával ellentétben a szemantikus függvény megadása nem kiegészítő lépés, hanem maga a szemantikadefiníció.

A While leíró szemantikája

$$\begin{aligned}
\mathcal{S}_{ds}[\![\textbf{skip}]\!] &= id_{\mathrm{State}} \\
\mathcal{S}_{ds}[\![x := a]\!]s &= s[x \mapsto \mathcal{A}[\![a]\!]s] \\
\mathcal{S}_{ds}[\![S_1 ; S_2]\!] &= \mathcal{S}_{ds}[\![S_2]\!] \circ \mathcal{S}_{ds}[\![S_1]\!] \\
\mathcal{S}_{ds}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!] &= cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![S_1]\!], \mathcal{S}_{ds}[\![S_2]\!]) \\
\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] &= \mathrm{FIX}\ F \\
\text{ahol } F\ g &= cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id_{\mathrm{State}})
\end{aligned}$$

A speciális zárójelek: $\llbracket$ és $\rrbracket$

- Az $\mathcal{A} : \mathrm{Aexp} \to (\mathrm{State} \to \mathrm{Integer})$ szemantikus függvényt alkalmazva az *a* kifejezésre ($\mathcal{A}\llbracket a \rrbracket$) megkapjuk az *a* szintaktikus elem szemantikáját.

- Miért kell az *a*-t, a függvény argumentumát speciális zárójelek közé tenni? Azért, mert ezzel jelezzük, hogy ez a paraméter nem matematikai formula, hanem egy szintaktikus elem, így nem szabad hagyományos módon kiértékelni.
- Az $\mathcal{A}\llbracket 3 + 2 \rrbracket$ formulát akár $\mathcal{A}(\text{"}3 + 2\text{"})$ formában is jelölhetnénk (a lényeg a megkülönböztetés), de az előbbi jelölés az elterjedt
- Amikor a szemantikus függvényt adjuk meg, akkor szintaktikus mintákat írunk a zárójelek közé, tehát a szintaktikus elem metaváltozókat tartalmazhat (pl. $\mathcal{A}\llbracket a_1 + a_2 \rrbracket$)

A függvénykompozíciót úgy értelmezzük, hogy csak akkor definiált, ha mindkét utasítás definiált az adott állapotban (az első az eredetiben, a második pedig a rákövetkezőben).
Ha valamelyik komponens szemantikája nem definiált, akkor a szekvencia sem definiált (pl. akár az első, akár a második program divergál, a szekvencia is divergál).

$$\begin{aligned}
&\mathcal{S}_{ds}[\![S_1;S_2]\!]s \\
&\quad = (\mathcal{S}_{ds}[\![S_2]\!] \circ \mathcal{S}_{ds}[\![S_1]\!])s \ = \ \mathcal{S}_{ds}[\![S_2]\!](\mathcal{S}_{ds}[\![S_1]\!]s) = \\
&\quad = \begin{cases} s'' & \text{ha } \exists s' \in \mathrm{State} : \mathcal{S}_{ds}[\![S_1]\!]s = s' \text{ és } \mathcal{S}_{ds}[\![S_2]\!]s' = s'' \\ \mathit{undef} & \text{ha } \mathcal{S}_{ds}[\![S_1]\!]s = \mathit{undef} \\ & \text{vagy } \exists s' \in \mathrm{State} : \mathcal{S}_{ds}[\![S_1]\!]s = s' \text{ és } \mathcal{S}_{ds}[\![S_2]\!]s' = \mathit{undef} \end{cases}
\end{aligned}$$

$$\mathcal{S}_{ds}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!]s =$$

$$= cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![S_1]\!], \mathcal{S}_{ds}[\![S_2]\!])s =$$

$$= \begin{cases} \mathcal{S}_{ds}[\![S_1]\!]s & \text{ha} \quad \mathcal{B}[\![b]\!]s = tt \\ \mathcal{S}_{ds}[\![S_2]\!]s & \text{ha} \quad \mathcal{B}[\![b]\!]s = ff \end{cases}$$

$$= \begin{cases} s' & \text{ha} \quad \mathcal{B}[\![b]\!]s = tt \;\text{ és }\; \mathcal{S}_{ds}[\![S_1]\!]s = s' \\ & \text{vagy ha } \mathcal{B}[\![b]\!]s = ff \;\text{ és }\; \mathcal{S}_{ds}[\![S_2]\!]s = s' \\ undef & \text{ha} \quad \mathcal{B}[\![b]\!]s = tt \;\text{ és }\; \mathcal{S}_{ds}[\![S_1]\!]s = undef \\ & \text{vagy ha } \mathcal{B}[\![b]\!]s = ff \;\text{ és }\; \mathcal{S}_{ds}[\![S_2]\!]s = undef \end{cases}$$

A korábbi szemantikadefiníciók alapján elvárjuk, hogy a ciklus szemantikája ekvivalens a kifejtésének szemantikájával:

$$
\begin{aligned}
&\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] \equiv \\
&\qquad \equiv \mathcal{S}_{ds}[\![\textbf{if } b \textbf{ then } (S; \textbf{while } b \textbf{ do } S) \textbf{ else skip}]\!] \\
&\qquad \equiv cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![S; \textbf{while } b \textbf{ do } S]\!], \mathcal{S}_{ds}[\![\textbf{skip}]\!]) \\
&\qquad \equiv cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] \circ \mathcal{S}_{ds}[\![S]\!], id_{\mathrm{State}})
\end{aligned}
$$

Továbbá gondolhatnánk, hogy definiálhatjuk is ezt az összefüggést kihasználva:

$$\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] = cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] \circ \mathcal{S}_{ds}[\![S]\!], id_{\mathrm{State}})$$

Azonban ez a definíció nem lenne kompozicionális, tehát nem ezt fogjuk használni.

Legyen $g = \mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!]$.

Alakítsuk át az előző formulát, helyettesítsük g-t:

$$g = cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id_{\mathrm{State}})$$

Tisztán látható, hogy ez egy rekurzív formula, mivel a g (a ciklus szemantikája) megjelenik a formula jobb oldalán is.

Ahhoz, hogy g-t meghatározzuk, bevezetünk egy F funkcionált:

$$F(g) = cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id_{\mathrm{State}})$$

Így F fixpontja megadja g-t. Azok lesznek “megfelelő” szemantikadefiníciók, amelyekre $F(g) = g$, de a ciklus szemantikájaként mi F legkisebb fixpontját fogjuk használni.

Tehát a ciklus denotációs szemantikáját a következő funkcionál fixpontjának kiszámításával kapjuk:

$$F(g) = cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id_{\mathrm{State}})$$

Azaz

$$F(g)s = \begin{cases} (g \circ \mathcal{S}_{ds}[\![S]\!])s & \text{ha} \quad \mathcal{B}[\![b]\!]s = tt \\ s & \text{ha} \quad \mathcal{B}[\![b]\!]s = ff \end{cases}$$

A fixpont kiszámításához egy fixpont-kombinátort (*FIX*) használunk, amely a tetszőlegesen sok fixpont közül a legkisebbet fogja megadni.

Szemantikus ekvivalencia

A szemantikus ekvivalencia megadja, mikor tekintünk két utasítást ekvivalensnek, mikor megegyező a két utasítás hatása.

S_1 és S_2 szemantikusan ekvivalensek ($S_1 \equiv S_2$) a leíró szemantikában akkor és csak akkor, ha $\mathcal{S}_{ds}[\![S_1]\!] = \mathcal{S}_{ds}[\![S_2]\!]$.

Vagyis minden s és s' állapotra

- $\mathcal{S}_{ds}[\![S_1]\!]s = s'$ akkor és csak akkor $\mathcal{S}_{ds}[\![S_2]\!]s = s'$
- $\mathcal{S}_{ds}[\![S_1]\!]s = \mathit{undef}$ akkor és csak akkor $\mathcal{S}_{ds}[\![S_2]\!]s = \mathit{undef}$

Legyen $s = [x \mapsto 10]$ és $s' = [x \mapsto 11]$.

$$
\begin{aligned}
&\mathcal{S}_{ds}[\![x := x + 1 \;;\; \textbf{if } x > 10 \textbf{ then } x := 10 \textbf{ else skip}]\!]s = \\
&\quad = (\mathcal{S}_{ds}[\![\textbf{if } x > 10 \textbf{ then } x := 10 \textbf{ else skip}]\!] \circ \mathcal{S}_{ds}[\![x := x + 1]\!])s = \\
&\quad = \mathcal{S}_{ds}[\![\textbf{if } x > 10 \textbf{ then } x := 10 \textbf{ else skip}]\!](\mathcal{S}_{ds}[\![x := x + 1]\!]s) = \\
&\quad = \mathcal{S}_{ds}[\![\textbf{if } x > 10 \textbf{ then } x := 10 \textbf{ else skip}]\!](s[x \mapsto \mathcal{A}[\![x + 1]\!]s]) = \\
&\quad = \mathcal{S}_{ds}[\![\textbf{if } x > 10 \textbf{ then } x := 10 \textbf{ else skip}]\!](s[x \mapsto 11]) =
\end{aligned}
$$

$$
\begin{aligned}
&\quad = \mathcal{S}_{ds}[\![\textbf{if } x > 10 \textbf{ then } x := 10 \textbf{ else skip}]\!]s' = \\
&\quad = cond(\mathcal{B}[\![x > 10]\!], \mathcal{S}_{ds}[\![x := 10]\!], \mathcal{S}_{ds}[\![\textbf{skip}]\!])s' = \\
&\quad = \begin{cases} \mathcal{S}_{ds}[\![x := 10]\!]s' & \text{ha} \quad \mathcal{B}[\![x > 10]\!]s' = tt \\ \mathcal{S}_{ds}[\![\textbf{skip}]\!]s' & \text{ha} \quad \mathcal{B}[\![x > 10]\!]s' = ff \end{cases} = \\
&\quad = \mathcal{S}_{ds}[\![x := 10]\!]s' = \\
&\quad = s[x \mapsto \mathcal{A}[\![10]\!]]s' = \\
&\quad = s'[x \mapsto 10] = s
\end{aligned}
$$

- Bizonyítható, hogy az eddigi szemantikadefinícióink ($\mathcal{S}_{ds}$, $\mathcal{S}_{sos}$ és $\mathcal{S}_{ns}$) ekvivalensek, tehát ugyanazokat a függvényeket rendelik az utasításhoz
- Ezek a módszerek nem egymás konkurenciái, hiszen különböző absztrakciós szinten definiálják ugyanazt a jelentésfogalmat

- A fentiek közül a denotációs szemantika a legabsztraktabb
- Az operációsakkal ellentétben ez teljesen kompozicionális
- A denotációs szemantikával kinyert jelentésfogalmak tovább transzformálhatóak

- A denotációs szemantika alapötlete
- Leképezhető matematikai objektumok, domainek
- Kompozicionalitás és a strukturális indukció
- A különböző szemantikadefiníciók viszonya

λ

A While nyelv végrehajtható denotációs szemantikája elérhető a kurzus anyagai között. A leíró jellegű szemantikát Haskellben definiáltuk.