Fixpont elmélet

Dr. Horpácsi Dániel
ELTE Informatikai Kar
2023-2024-2

Az előadás témája

Domainek és fixpontok

Hogyan adjuk meg a ciklus denotációját?

Emlékeztető: denotációs szemantika

$$\mathcal{S}_{ds}[\![program]\!] = effect$$

- Szintaktikus objektumokhoz szemantikus, matematikai objektumokat rendel
- A hozzárendelt denotáció adja meg a szintaktikus elem jelentését
- A denotációs függvény teljes és kompozicionális
- Általában mintaillesztéssel definiáljuk

Emlékeztető: denotációs szemantika

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

A While leíró szemantikája

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

Az iteratív programkonstrukciók denotációjának kompozicionális megadása valamivel bonyolultabb!

Az előző előadáson említettük, elvárjuk, hogy a ciklus szemantikája megegyezzen az egylépéses kifejtésének szemantikájával:

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

Ha így akarnánk megadni, akkor ez egy rekurzív definíció lenne:

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

- Egyrészt nem kompozicionális, másrészt nem világos, jól definiált-e, meghatároz-e egy (parciális) függvényt.
- Vajon igaz, hogy a rekurzív függvényhívások mindig "kisebb" értékekre vonatkoznak? Mi az, hogy "kisebb" állapot? (Tipp: hátralévő iterációk.)

Több lehetséges denotáció?

$$\begin{aligned}\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] &= cond(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] \circ \mathcal{S}_{ds}[\![S]\!], id_{\text{State}}) \\ g &= cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id_{\text{State}})\end{aligned}$$

Vizsgáljuk meg a következő utasítást:

$$\textbf{while } \neg x = 0 \textbf{ do } x := x - 1$$

Itt g több értéket is felvehet, több függvényt is megad a fenti egyenleg megoldása g-re (az első esetben s' tetszőleges):

$$g\ s = \begin{cases} s[x \mapsto 0] & \text{ha} \quad s[x] \geq 0 \\ s' & \text{egyébként} \end{cases}$$

$$g\ s = \begin{cases} s[x \mapsto 0] & \text{ha} \quad s[x] \geq 0 \\ undef & \text{egyébként} \end{cases}$$

Több lehetséges denotáció?

Vizsgáljuk meg a következő utasítást:

$$\textbf{while } \neg x = 0 \textbf{ do } x := x - 1$$

Itt g több értéket is felvehet, több függvényt is megad a fenti egyenleg megoldása g-re (az első esetben s' tetszőleges):

$$g\ s \quad = \begin{cases} s[x \mapsto 0] & \text{ha} \quad s[x] \geq 0 \\ s' & \text{egyébként} \end{cases}$$

$$g\ s \quad = \begin{cases} s[x \mapsto 0] & \text{ha} \quad s[x] \geq 0 \\ \mathit{undef} & \text{egyébként} \end{cases}$$

Valójában az első megoldás nem megfelelő, mert olyan helyeken is jelentést rendel a ciklushoz, ahol nem terminál. A második pontosabban definiálja, mert csak ott rendel hozzá jelentést, ahol terminál.

A domain a jelentéseket leíró szemantikus objektumokat tartalmazó halmaz és az elemein értelmezett műveletek együttese.

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

Az utasítások esetében a domain egy láncteljes részbenrendezett halmaz (cpo) lesz, amely rendelkezik minimális (bottom) elemmel:

$$(\mathrm{State} \hookrightarrow \mathrm{State}, \sqsubseteq, \perp) \quad \text{ahol} \quad \mathrm{State} = \mathrm{Var} \to \mathbb{Z}$$

Fixpont elmélet: hogyan lehet meghatározni függvények fixpontját?
Domain elmélet: hogyan lehet a fixpont elméletben használható domaineket és rajtuk folytonos függvényeket konstrutálni?

Az utasítások szemantikus tartománya

Az utasítás kategóriájához rendelt szemantikus domain egy részbenrendezett matematikai struktúra:

$$(\mathrm{State} \hookrightarrow \mathrm{State}, \sqsubseteq, \perp)$$

ahol a $\sqsubseteq$ rendezés a parciális függvényeket ($\mathrm{State} \hookrightarrow \mathrm{State}$) a következőképp rendezi (g_1 kevésbé definiált, mint g_2):

$$g_1 \sqsubseteq g_2 \qquad \text{amennyiben} \qquad g_1 s = s' \Rightarrow g_2 s = s'$$

Valóban rendezés, mert reflexív, tranzitív és antiszimmetrikus reláció:

$$g_1 \sqsubseteq g_1 \qquad \text{(reflexív)}$$

$$g_1 \sqsubseteq g_2 \text{ és } g_2 \sqsubseteq g_3 \text{ implikálja } g_1 \sqsubseteq g_3 \qquad \text{(tranzitív)}$$

$$g_1 \sqsubseteq g_2 \text{ és } g_2 \sqsubseteq g_1 \text{ implikálja } g_1 = g_2 \qquad \text{(antiszimmetrikus)}$$

A fenti rendezés szerinti legkisebb (minimális) elem ($\perp$) a $\mathrm{State} \hookrightarrow \mathrm{State}$ halmazban egyértelmű, és a következőképp adható meg:

$$\perp s = \mathit{undef} \quad \text{minden } s \text{ állapotra}$$

A $\perp$ tehát az üres függvény (a relációja üres halmaz, $\mathit{graph}(\perp) = \{\}$), és nyilvánvalóan minden halmazbeli függvény több helyen definiált, tehát adódik, hogy $\perp \sqsubseteq g$ minden $g \in \mathrm{State} \hookrightarrow \mathrm{State}$ függvényre.

Kérdés: vajon van-e olyan While program, aminek a denotációja a $\perp$ függvény lesz?

Több lehetséges denotáció?

Vizsgáljuk meg a következő utasítást:

$$\textbf{while } \neg x = 0 \textbf{ do } x := x - 1$$

Itt g több értéket is felvehet, több függvényt is megad a fenti egyenleg megoldása g-re (az első esetben s' tetszőleges):

$$g\ s \quad = \begin{cases} s[x \mapsto 0] & \text{ha} \quad s[x] \geq 0 \\ s' & \text{egyébként} \end{cases}$$

$$g\ s \quad = \begin{cases} s[x \mapsto 0] & \text{ha} \quad s[x] \geq 0 \\ \mathit{undef} & \text{egyébként} \end{cases}$$

Vegyük észre: a második (a jobbnak tekintett megoldás) kisebb a fenti rendezés szerint.

A ciklus jelentése megadható lenne rekurzív függvénnyel, ahol $iter_k(p, g)$ megadja a végállapotkat azokban a kezdőállapotokban, ahol a ciklus kevesebb mint k lépésben terminál.

$$\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!]s = \begin{cases} s' & \text{ha} \quad \exists k : iter_k(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![S]\!])s = s' \\ & \text{és} \quad s' \neq undef \\ undef & \text{egyébként} \end{cases}$$

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

$$iter_0(p, g)s = undef$$

$$iter_{k+1}(p, g)s = \begin{cases} iter_k(p, g)(g\ s) & \text{ha} \quad p\ s = tt \\ s & \text{ha} \quad p\ s = ff \end{cases}$$

Kitekintés: az iteráció definíciója

$$\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!]s = \begin{cases} s' & \text{ha} \quad \exists k : iter_k(\mathcal{B}[\![b]\!], \mathcal{S}_{ds}[\![S]\!])s = s' \\ & \text{és} \quad s' \neq undef \\ undef & \text{egyébként} \end{cases}$$

$$iter_0(p, g) = \perp$$
$$iter_{k+1}(p, g) = cond(p, iter_k(p, g) \circ g, id)$$

Az $iter_k$ határértéke a ciklus jelentéséhez tart (egyre több kezdőállapotban definiált, egyre kevesebb helyen nem definiált).

Nem pontosan így adjuk meg a ciklus denotációját, de ez lesz a megoldás lényege.

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

A formula jobb oldalát (a *kontextust*) egy magasabb rendű függvénnyé alakíthatjuk, amely a g-től függ. Ez az ún. generálófüggvény.

$F : (\mathrm{State} \hookrightarrow \mathrm{State}) \to (\mathrm{State} \hookrightarrow \mathrm{State})$

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

Így már jól látszik, hogy valójában a formula megoldásait megkapjuk F fixpontjainak kiszámításával. F az úgynevezett finomító függvény.

$$\mathcal{S}_{ds}[\![\textbf{while } b \textbf{ do } S]\!] = \mathrm{FIX}\ F$$

A generálófüggvény fixpontjai

- Minden ciklusnak megadható a leíró szemantikája?
- Minden ciklus esetén lesz fixpontja F-nek?
- Lehet több fixpontja? Melyik lesz a ciklus szemantikája?
- Hogyan lehet kiszámítani a fixpontot?

Megmutatjuk, hogy minden ciklusnak definiált a leíró szemantikája és mindig ki is tudjuk számítani (az F legkisebb fixpontjaként).

A fixpont kiszámítása

Következmény: ha azt szeretnénk, hogy a denotáció a lehető legkevesebb helyen legyen definiálva, akkor valójában a $\sqsubseteq$ rendezés szerinti legkisebb fixpontot kell meghatározni az F függvény esetében.

Intuíció:

- A ciklus informális szemantikája szerint a ciklusmag hatását addig ismételjük, amíg a feltétel hamissá nem válik.
- A leíró szemantikában a ciklus szemantikáját az F ismételt alkalmazásával közelítjük, majd az F legkisebb fixpontjában megkapjuk a szemantika legpontosabb közelítését.
- Ez valójában a fixpontszámításnak egy módszere, amelyet bizonyos előfeltételek mellett alkalmazhatunk (cpo + folytonosság).

Mivel $\perp$ a legkisebb elem, ezért $\perp \sqsubseteq F(\perp)$, továbbá mivel F monoton, F ismételt alkalmazásaival egy láncot (rendezett részhalmazt) kapunk:

$$\perp \sqsubseteq F(\perp) \sqsubseteq F(F(\perp)) \sqsubseteq F(F(F(\perp))) \sqsubseteq \ldots$$

Ezen sorozat ($\mathrm{State} \hookrightarrow \mathrm{State}$ függvények növekvő lánca) egyre pontosabb közelítését adja a ciklus szemantikának.

Ha F folytonos, akkor a lánc legkisebb felső korlátja megadja a ciklus szemantikáját, amely egyenlő az F funkcionál legkisebb fixpontjával (Kleene és Tarski fixponttétele).

- Kleene fixponttétele szerint ha F folytonos, akkor az előző dián felírt lánc legkisebb felső korlát megadja az F legkisebb fixpontját:

$$FIX\ F = \sqcup\{F^n(\perp) \mid n \geq 0\}$$

- Informálisan: $FIX\ F$ a “legkisebb” $\mathrm{State} \hookrightarrow \mathrm{State}$ típusú függvény, amely minden információt tartalmaz, amely az $F^n(\perp)$ láncban előáll. Azaz,

$$\perp \sqsubseteq F(\perp) \sqsubseteq F(F(\perp)) \sqsubseteq \ \dots \ \sqsubseteq FIX\ F$$

- Megjegyzés: a legkisebb fixpont egyértelmű.

- Tehát a legkisebb fixpont előállítható a következő formulával:

$$\mathit{FIX}\ F = \sqcup\{F^n(\perp) \mid n \geq 0\}$$

- Ehhez azonban tudnunk kell, hogy a láncnak létezik-e legkisebb felső korlátja.

- A $(\mathrm{State} \hookrightarrow \mathrm{State}, \sqsubseteq, \perp)$ struktúra egy láncteljes részbenrendezett halmaz (complete partial order, cpo), azaz minden láncnak van egy egyértelmű legkisebb felső korlátja:

  Legyen Y egy $\mathrm{State} \hookrightarrow \mathrm{State}$ lánc, ekkor

$$\mathit{graph}(\sqcup Y) = \cup\{\mathit{graph}(g) \mid g \in Y\}$$

- Tehát a legkisebb fixpont előállítható a következő formulával:

$$\mathit{FIX}\ F = \sqcup\{F^n(\perp) \mid n \geq 0\}$$

- Azt már tudjuk, hogy ez a felső korlát létezik, de ahhoz, hogy megegyezzen egy fixponttal ($F\ \big(\mathit{FIX}\ F\big) = \mathit{FIX}\ F$), az F funkcionálnak folytonosnak kell lennie.

- F (Scott-)folytonos akkor és csak akkor, ha monoton és megtartja a láncok legkisebb felső korlátját, tehát:

$$\sqcup \big\{F\big(g\big) \mid g \in Y\big\} = F\big(\sqcup Y\big)$$

Az F folytonossága

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

- Az F felbontható két másik függvény kompozíciójára:

$$\begin{aligned} F\ (g) &= cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id_{\text{State}}) \\ &= (F_2 \circ F_1)(g) \quad \text{ahol} \\ &\qquad F_1(g) = g \circ \mathcal{S}_{ds}[\![S]\!] \\ &\qquad F_2(g) = cond(\mathcal{B}[\![b]\!], g, id_{\text{State}}) \end{aligned}$$

- Ha be kell látnunk, hogy F folytonos, elég belátni, hogy F_1 és F_2 folytonosak, mert két folytonos függvény kompozíciója is folytonos lesz.
- (monotonitás + legkisebb felső korlát megtartása => folytonosság)

- Ezzel teljesen áttekintettük a magnyelv (While) leíró szemantikáját
- Formálisan definiáltuk a $\mathcal{S}_{ds} : \mathrm{Stm} \to (\mathrm{State} \to \mathrm{State})$ függvényben
- A ciklusnak kompozicionálisan definiáltuk a jelentését

- A denotációs szemantika ($\mathcal{S}_{ds}$) totális, tehát minden lehetséges utasításnak megadja a jelentését (strukturális indukcióval belátható)

- A magnyelv csak ciklusokkal fejez ki ismétlődést, de a rekurzív függvények esetében is szükség lesz hasonló számításokra
- Azok a módszerek, amiket a ciklus szemantikájának definiálásakor felhasználtunk, alkalmasak a rekurzív függvények szemantikájának leírására is
- Valójában most is egy rekurzív függvénynek adtuk meg a szemantikáját, hiszen a ciklus kifejtésével egy olyan rekurzív formulát kaptunk, amelyet rekurzív függvénydefiníciók jelentésének meghatározásakor kapunk

λ

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