Programozási módszertan

3. Elemi programok, programkonstrukciók

Feldhoffer Gergely

2012

Elemi programok

Elemi program a SKIP, ABORT, és az értékadás:

$$S = \{(a, \alpha) \in A \times A^{**} | \alpha = red(< a, b >) \vee \alpha =< aaa \cdots >\}$$

$$SKIP = \{(a, \alpha) \in A \times A^{**} | \alpha =< a >\}$$

$$ABORT = \{(a, \alpha) \in A \times A^{**} | \alpha =< aaa \cdots >\}$$

Értékadás

Az értékadás kicsit bonyolultabb. Minden értékadás lényegében egy függvény, ami hozzárendel sok állapottérbeli ponthoz ahhoz tartozó cél-pontokat. Az x:=x+1 például minden x változóértékhez az eggyel x állapottérkomponensbeli értékez rendeli hozzá a többi dimenzió nyugalomban hagyásával.

A legegyszerűbb értékadás tehát az egy változónak a teljes típusértékhalmazon értelmezett függvényével írható fel:
$S_E = \{(a, \alpha) \in A \times A^{**} | \alpha = red(< a, b >) \wedge b = E(a)\}$
ahol $\mathcal{D}_E = A \wedge E$ függvény

Értékadás - bonyolultabb esetek

Értékkiválasztás:$S_E = \{(a, \alpha) \in A \times A^{**} | \alpha = red(< a, b >) \wedge b \in E(a)\}$
ahol $\mathcal{D}_E = A \wedge E \subseteq A \times A$

Lehet kezelni azt, ha E nincs mindenhol értelmezve, pl. a $/=$ b. Ilyenkor a nem értelmezett pontokban a program abortál:
$S_E = \{(a, \alpha) \in A \times A^{**} | \alpha = red(< a, b >) \wedge a \in \mathcal{D}_E \wedge b \in E(a)\} \cup \{(a, \alpha) \in A \times A^{**} | \alpha =< aaa \cdots > \wedge a \notin \mathcal{D}_F\}$

Tetszőleges állapotváltozás felfogható értékadásnak. A nagy kérdés mindig az, hogy milyen jellegű értékadásokat támogat az a gép, amire a programot írjuk. A bonyolultságelmélet orákulumnak hívja azokat az egzotikus megengedett értékadásokat, amiket egy hagyományos gép nem képes egy értékadásban kiszámolni, és ezzel tételeket lehet belátni, pl ha az utazó ügynök problémát megengedjük egy lépéses műveletnek egy tetszőleges reprezentációban, akkor azon a gépen P=NP.

Elemi programok leggyengébb előfeltétele

$lf(SKIP, R) = R$

$lf(ABORT, R) = Hamis$

$\lceil lf(S_E, R) \rceil = \{a \in A | E(a) \in \lceil R \rceil\}$
ha $\mathcal{D}_E = A \wedge E$ függvény

Értékadás felírása változókkal

Példa:
$A = \underset{x}{\mathbb{Z}} \times \underset{y}{\mathbb{Z}}, E = \{(a, b) \in A \times A | y(b) = 2 * x(a) \wedge x(b) = x(a)\}$

ilyenkor S_E felírható úgy, hogy $y := 2 * x$

Az E függvény tipikusan felbontható $E_1, E_2, \dots E_n$ egydimenziós relációkra/függvényekre változónként. Az egyszerre több változó értékét megváltoztató értékadást szimultán értékadásnak nevezzük.

Programkonstrukciók

Háromféle programkonstrukciót használunk: szekvencia, elágazás, ciklus

A szekvencia programok egymás utáni végrehajtását jelenti, az elágazás feltételes végrehajtást, a ciklus pedig ismétlést egy adott ciklusfeltétel teljesüléséig.

Def (szekvencia)

Legyenek $S_1, S_2 \subseteq A \times A^{**}$ programok. Az $S \subseteq A \times A^{**}$ relációt S_1 és S_2 szekvenciájának nevezzük, és $(S_1; S_2)$-vel jelöljuk, ha $\forall a \in A$:

$$\begin{aligned} S(a) &= \{\alpha \in A^{\infty} | \alpha \in S_1(a)\} \cup \\ &\cup \{red(concat(\alpha, \beta)) \in A^{**} | \alpha \in S_1(a) \cap A^* \wedge \beta \in S_2(\tau(\alpha))\}. \end{aligned}$$

S_1
S_2

Elágazás

Def (elágazás)

Legyenek $\pi_1, \ldots, \pi_n : A \to \mathbb{L}$ *feltételek,* $S_1, \ldots, S_n$ *programok* A*-n. Ekkor az* $IF \subseteq A \times A^{**}$ *relációt az* S_i*-kből képzett* π_i*-k által meghatárpzott elágazásnak nevezzük, és* $(\pi_1 : S_1, \ldots, \pi_n : S_n)$*-nel jelöljük, ha* $\forall a \in A$

$$IF(a) = \bigcup_{i=1}^{n} w_i(a) \cup w_0(a),$$

ahol $\forall i \in [1..n]$:

$$w_i(a) = \begin{cases} S_i(a) & \text{ha}\, \pi_i(a) \\ \emptyset & \text{kulonben} \end{cases}$$

és

$$w_0(a) = \begin{cases} < a, a, a, \cdots > & \text{ha}\, \forall i \in [1..n] : \neg\pi_i(a) \\ \emptyset & \text{kulonben} \end{cases}$$

Elágazás

Ez az elágazásfogalom megenged néhány érdekes lehetőséget: a hagyományos nyelvek elágazásai automatikusan SKIP-et rendelnek a le nem fedett feltételekhez, pl egy switch()-ben ha nincs default, és egyik feltétel sem teljesül. Ez az elágazásfogalom viszont ABORT-ot rendel hozzá, mert az általánosság megszorítása nélkül feltehető, hogy ha a programozó szándéka az, hogy "különben SKIP" akkor jelzi.

Előfordulhat, hogy egy állapotban több π_i igaz, ezekben az elágazás nem determinisztikus.

Ciklus

Def (ciklus)

Legyen π feltétel és S_0 program A-n. A $DO \subseteq A \times A^{**}$ relációt az S_0-ból π feltétellel képzett ciklusnak nevezzük, ha

- $\forall a \notin \lceil \pi \rceil : DO(a) = \{< a >\}$
- $\forall a \in \lceil \pi \rceil : DO(a) =$

$$
\begin{aligned}
& \{\alpha \in A^{**} | \exists \alpha^1, \ldots, \alpha^n \in A^{**} : && (1)\\
& \alpha = red(concat(\alpha^1, \ldots, \alpha^n)) \wedge \alpha^1 \in S_0(a) \wedge && (2)\\
& \wedge(\forall i \in [1..n-1] : \alpha^i \in A^* \wedge && (3)\\
& \wedge \alpha^{i+1} \in S_0(\tau(\alpha^i)) \wedge \pi(\tau(\alpha^i))) \wedge && (4)\\
& \wedge((\alpha^n \in A^\infty) \vee (\alpha^n \in A^* \wedge \neg\pi(\tau(\alpha^n))))\} && (5)\\
\cup \quad & \{\alpha \in A^\infty | \forall i \in \mathbb{N} : && (6)\\
& \exists \alpha^i \in A^* : \alpha = red(concat(\alpha^1, \alpha^2, \ldots)) \wedge && (7)\\
& \alpha^1 \in S_0(a) \wedge (\forall i \in \mathbb{N} : \alpha^i \in A^* \wedge && (8)\\
& \alpha^{i+1} \in S_0(\tau(\alpha^i)) \wedge \pi(\tau(\alpha^i)))\} && (9)
\end{aligned}
$$

Azaz: a ciklus a ciklusmag olyan véges ismétlése, amelyben a kapott sorozat a ciklusmag sorozatainak redukált konkatenáltja(2) ahol az egyes lefutások az első n-1 esetben végesek (3), az egyes lefutások ott indulnak ahol az előző megállt(4) és a ciklusfeltétel még igaz(4) és vagy végtelen sorozat a vége, vagy egy olyan véges sorozat, aminek a végére már nem teljesül a ciklusfeltétel(5). Az is lehet, hogy a program úgy kerül végtelen ciklusba, hogy nem tartalmaz végtelen sorozatot az egyes ciklusmag lefutásokban, hanem végtelen sokáig igaz a ciklusfeltétel(6-9).

Programkonstrukciók programfüggvényei

Az egyes programkonstrukciók értelmes használatához érdemes felírni a programfüggvényeiket.

Az a célunk, hogy a megoldás fogalmához a specifikáció és a leggyengébb előfeltétel fogalmain keresztül jussunk el, amihez szükség van a programfüggvényekre.

Szekvencia programfüggvénye

Az S szekvencia programfüggvénye az S_1 és S_2 programfüggvényeinek szigorú kompozíciója

$$\begin{aligned} p(S) &= p(S_2) \odot p(S_1) = \\ &= \{(a,c) \in A \times A | \exists b \in B : (a,b) \in p(S_1) \wedge (b,c) \in p(S_2) \wedge \\ &\quad \wedge \underbrace{p(S_1)(a) \subseteq \mathcal{D}_{p(S_2)}}_{szigoritas}\} \end{aligned}$$

Elágazás programfüggvénye

Tétel (Az elágazás programfüggvénye)

- $\mathcal{D}_{p(IF)} = \{a \in A | \bigvee_{i=1}^{n} \pi_i(a) \wedge \forall i \in [1..n] : \pi_i(a) \Rightarrow a \in \mathcal{D}_{p(S_i)}.\}$
  Tehát az elágazás programfüggvénye azokon az a elemeken értelmezett, amelyekre legalább egy feltétel igaz, és minden igaz feltétel esetén a megfelelő programág programfüggvénye értelmezve van a-ban (tehát egyik végrehajtási ág sem száll el).
- $\forall a \in \mathcal{D}_{p(IF)} p(IF)(a) = \bigcup_{i=1}^{n} pw_i(a),$
  $ahol pw_i(a) = \begin{cases} p(S_i(a)) & \text{ha } \pi_i(a) \\ \emptyset & \text{kulonben} \end{cases}$
  Minden $\mathcal{D}_{p(IF)}$-beli elemre a programfüggvény az igaz ágak programfüggvényeinek a képeinek az unióját rendeli a-hoz.

Ciklus programfüggvénye

Tétel (A ciklus programfüggvénye)

A cikklus programfüggvénye megegyezik a ciklusmag programfüggvényének ciklusfeltételre vonatkozó lezártjával.

$$p(DO) = \overline{p(S_0)|_\pi}$$

Ehhez tudni kell mi az a lezárt, és a megszorítás.

Def (lezárt)

Az $R \subset A \times A$ reláció lezártja az $\overline{R} \subseteq A \times A$ reláció, amelyre:

- $\mathcal{D}_{\overline{R}} = \{a \in A | \nexists \alpha \in A^\infty : \alpha_1 \in R(a) \wedge \forall i \in \mathbb{N} : \alpha_{i+1} \in R(\alpha_i)\}$
- $\forall a \in \mathcal{D}_{\overline{R}} : \overline{R}(a) = \{b \in A | \exists k \in \mathbb{N}_0 : b \in R^k(a) \wedge b \notin \mathcal{D}_R\}$

Ciklus programfüggvénye

Tehát egy reláció lezártja azokban a pontokban van értelmezve, ahonnan kiindulve nem lehet végtelen sokszor alkalmazni a relációt. Ezekhez a pontokhoz olyan pontokat rendel, amelyeket úgy kapunk, hogy véges sokszor alkalmazzuk a relációt és kikerülünk az eredeti reláció értelmezési tartományából.

Def (Reláció feltételre vonatkozó megszorítása)

Legyen $R \subseteq A \times A$ és $\pi : A \to \mathbb{L}$. Az R reláció megszorítása a π feltételre:

$$R|_\pi = (R \cap (\lceil \pi \rceil \times A)) \cup \{(a, a) \in A \times A | a \in \lceil \pi \rceil \setminus \mathcal{D}_R\}$$

A feltételre való megszorítás pedig abban különbözik a halmazra/altérre történő megszorítástól, hogy identitást képez a feltétel igazsághalmazán a reláció eredeti értelmezési tartományán kívüli pontokban.

Levezetési szabályok

A következőkben olyan összefüggéseket írunk fel, amik segítségével ellenőrizhető, hogy helyesen működik-e egy programkonstrukció

Ezeket az összefüggéseket levezetési szabályoknak fogjuk nevezni

A levezetési szabályok megfordíthatóak lesznek: minden jól működő programkonstrukcióhoz meg lehet majd találni a megfelelő közbülső feltételeket kielégítő segédfeltételeket

Szekvencia levezetési szabálya

Tétel (A szekvencia levezetési szabálya)

Legyen $S = (S_1; S_2)$*, és adott* Q, R *és* Q' *állítás A-n. Ha*

1. $Q \Rightarrow \mathit{lf}(S_1, Q')$ *és*
2. $Q' \Rightarrow \mathit{lf}(S_2, R)$*,*

akkor $Q \Rightarrow \mathit{lf}(S, R)$*.*

Azaz egy Q' állításon keresztül bizonyítható a szekvencia helyessége

Elágazás levezetési szabálya

Tétel (elágazás levezetési szabálya)

Legyen $IF = (\pi_1 : S_1, \dots \pi_n : S_n)$, *és adott* Q, R *állítás* A*-n. Ha*

1. $Q \Rightarrow (\bigvee_{i=1}^{n} \pi_i)$
   (tehát minden $a \in A$ *elemre, amelyre* Q *igaz, arra igaz lesz legalább 1 feltétel) és*
2. $\forall i \in [1..n] : Q \wedge \pi_i \Rightarrow lf(S_i, R)$
   (mindegyik igaz feltétel esetén az végrehajtott programág a megfelelő utófeltételbe visz),

akkor $Q \Rightarrow lf(IF, R)$.

Ha az egyes ágakban használt programok a megfelelő π_i feltétel mellett megoldják az R feltétellel kifejezett részfeladatot, akkor a teljes elágazás is helyes

Ciklus levezetési szabálya

Tétel (ciklus levezetési szabálya)

Adott P, Q, R állítás A-n, $t : A \to \mathbb{Z}$ függvény és legyen $DO = (\pi, S_0)$. Ha

1. $Q \Rightarrow P$
2. $P \wedge \neg\pi \Rightarrow R$
3. $P \wedge \pi \Rightarrow t > 0$
4. $P \wedge \pi \Rightarrow lf(S_0, P)$
5. $P \wedge \pi \wedge t = t_0 \Rightarrow lf(S_0, t < t_0)$

akkor $Q \Rightarrow lf(DO, R)$

A P-t ciklusinvariánsnak nevezzük, és az a bevezetésének a célja, hogy meglegyen a ciklus folyamán a folytonos ellenőrzése az egyes változók jelentéseinek betartásának: igaznak kell lennie a ciklus belépési pontjában és a ciklusfeltételből kivezető utolsó iteráció végén is. A t-t terminálófüggvénynek nevezzük, és a ciklus végességét garantálja.