Programozáselmélet
Előadás jegyzet

Készült Borsi Zsolt előadásai
és Gregorics Tibor gyakorlatai alapján

Sárközi Gergő, 2021-22-1. félév
Nincsen lektorálva!

Tartalomjegyzék

1. Bevezetés

Intervallum $[a..b] ::= \{x \in \mathbb{Z} | a \leq x \wedge x \leq b\}$
Üres, ha $a > b$.

Reláció, rendezett pár $A \times B ::= \{(a, b) | a \in A \wedge b \in B\}$
Az $A \times B$ halmaz minden részhalmazát, az üres halmazt is, relációnak nevezzük.
$|A \times B| = |A| * |B|$
A rendezett pár felfogható relációként is: A-ról B-re képezés.
Jelölése: $R \subseteq A \times B$
Legyen $A = 1, 2, 3$, $B = a, b, c, d$ és $R = (1, a), (1, b), (2, c)$.
Ekkor $R \subseteq A \times B$, valamint $D_R = \{1, 2\}$, és $R(1) = \{a, b\}$.
$R(x) = \varnothing \Leftrightarrow |R(x)| = 0 \Leftrightarrow x \notin D_R$
$D_R ::= \{a \in A | \exists b \in B : (a, b) \in R\}$
$R_R ::= \{b \in B | \exists a \in A : (a, b) \in R\}$
$R(a) ::= \{b \in B | (a, b) \in R\}$
Reláció inverze:
$R^{-1} = \{(b, a) \in B \times A | (a, b) \in R\}$

Függvény, determinisztikus reláció -
Legyen A és B nemüres halmaz. $R \subseteq A \times B$ függvény, ha $\forall a \in A : |R(a)| \leq 1$.
Függvény jelölése: $R \in A \to B$ (ekkor $|R(a)| \leq 1$)
vagy $R : A \to B$ (ekkor $|R(a)| = 1$, azaz $D_R = A$)
$R : A \to B$ esetén sokszor $R(a) = \{b\}$ helyett csak $R(a) = b$-t használunk

Kompozíció $P \subseteq A \times B$ és $Q \subseteq B \times C$
Normál:
$Q \circ P = \{(a, c) \in A \times C \mid \exists b \in B : (a, b) \in P \wedge (b, c) \in Q\}$
Szigorú: $(\exists (a, b) \in P : \exists c \in C : (b, c) \notin Q \implies (Q \odot P)(a) = \varnothing)$
$Q \odot P = \{(a, c) \in A \times C \mid \exists b \in B : (a, b) \in P \wedge (b, c) \in Q \wedge P(a) \subseteq D_Q\}$

Sorozatok $H^{**} = H^* \cup H^\infty$
$H = \{1, 2, 3\}$
$< 1, 5, 3 > \notin H^*$, mert $5 \notin H$
$< 1, 1, 3 > \in H^*$
$< 1, 1, 1, 1, ... > \notin H^*$, viszont $\in H^\infty$

Egyéb -
hamis mindent implikál: (HAMIS $\implies$ bármi) = IGAZ

2. Alapfogalmak

2.1. Állapottér

- Állapot: változókkal címkézett értékek halmaza
  - példa: $a = \{v_1 : a_1, v_2 : a_2\}$
  - változó használata: $n(\{n : 6\}) = 6$ (valójában $\{6\}$, de elnézzük)
- Érték: típusérték-halmaz egy eleme
- Állapottér: összes lehetséges állapot halmaza
  - jelölés: $A = (v_1 : A_1, v_2 : A_2)$
- Az A állapottérnek altere a B állapottér ($B < A$), ha a B állapottér típusértékei (számmosságot is figyelmbe véve) benne vannak A-ban is.
  - példa: $A = (x : \mathbb{N}, y : \mathbb{N}, z : \mathbb{N})$ és $B = (y : \mathbb{N}, x : \mathbb{N})$ akkor $B < A$.
  - A típusérték-halmazoknak meg kell egyezniük: az nem "elég", ha az egyik a másiknak csak a részhalmaza.
    Itt egyik sem részhalmaza a másiknak: $A = (x : \mathbb{N})$, $B = (x : \mathbb{Z})$
  - TODO lehet különböző a változó neve?

2.2. Feladat

- Feladat: reláció, amely egy állapottérről ugyan arra az állapottérre képez
  - Ha A állapottér, akkor $F \subseteq A \times A$ egy feladat.
  - Tehát $F = \{\}$ is egy feladat.

2.2.1. Példa

Adjuk meg egy pozitív egész szám egy pozitív osztóját.
$A = (n : \mathbb{N}^+, d : \mathbb{N}^+)$
$D_F = A$
$F \subseteq A \times A$
$F(\{n : 6, d : 11\}) = \{\{n : 6, d : 2\}, \{n : 6, d : 3\}, ...\}$
Itt a kimenetnél nézőpont kérdése, hogy n=6 feltétel-e, vagy meg lehetne változtatni.
$F(x, y) = \{(u, v); u = x \wedge v|x\}$
$F(x, *) = \{(*, v); v|x\}$
$F = \{(u, v) \in A \times A \mid d(v)|n(u)\}$

2.3. Program

- Leképezés, amely megadja bármelyik kiinduló állapotra az abból induló végrehajtási sorozatokat.
- Program: $S \subseteq A \times (\overline{A} \cup \{fail\})^{**}$ ahol:
  - $\overline{A} = \bigcup_{A \leq B} B$
  - $D_S = A$ (minden bemenetet lekezel)
  - $\forall a \in A : \forall vs \in S(a) : |vs| \geq 1 \wedge vs_1 = a$ (végrehajtási sorozat nem üres és első tagja a bemenet)
  - $\forall vs \in R_S : (\forall i \in \mathbb{N}^+ : i < |vs| \rightarrow vs_i \neq fail)$ (csak az utolsó állapot lehet fail)
  - $\forall vs \in R_S : |vs| < \infty \rightarrow vs_{|vs|} \in A \cup \{fail\}$ (az utolsó állapot, ha van ilyen, csak alap-állapottérbeli vagy fail lehet)

2.3.1. Alap-állapottér

- nincsenek benne a segédváltozók
- kiindulási állapot mindig alap-állapottérbeli
- végállapot alap-állapottérbeli, vagy fail

2.3.2. Végrehajtási sorozat

- első tagja mindig a bemeneti állapot
- ha van benne fail, akkor csak leghátul lehet
- segédváltozók a végrehajtási sorozat első és utolsó tagjában nincsenek, viszont a többiben igen
- legalább 1, max. végtelen hosszú
- egy kezdő állapotból több különböző végrehajtási sorozat indulhat
- értékadás, ami nem változtat semmit is generál új végrehajtási sorozatot

2.3.3. Példa

$A = (x : \{1, 2, 3\})$
S program (azaz halmaz) elemei: (mindegyik felírás helyes)
$\{x : 1\} \rightarrow < \{x : 1\}, \{x : 3\} >$
$1 \rightarrow < 1, 1, 1 >$
$(2) \rightarrow < (2), (3), (4) >$
$(3, < 3, fail >)$

2.4. Programfüggvény

- megadja egy program lehetséges végállapotait (amik sem fail, sem végtelenek nem lehetnek) egy adott kezdőállapot esetén
- Programfüggvény: egy reláció, pl. $p(S) \subseteq A \times A$
  - ez a $S \subseteq A \times (\overline{A} \cup \{fail\})^{**}$ programhoz van
  - $D_{p(S)} = \{a \in A | S(a) \subseteq \overline{A}^*\}$ (ott értelmes, ahol S csak véges, csak nem fail végrehajtási sorozatot ad)
  - $\forall a \in D_{p(S)} : p(S)(a) = \{b \in A | \exists vs \in S(a) : b = vs_{|vs|}\}$ (azaz $p(S)(x) =$ 'S program lehetséges végállapotjai (halmaza) x kezdőállapot esetén')
  - ha $x \notin D_{p(S)}$, akkor $p(S)(x) = \varnothing$

2.5. Megoldás

- S program megoldja az F feladatot (S program teljesen helyes F feladatra nézve) ha:
  - $D_F \subseteq D_{p(S)}$ (feladat minden kezdőállapotát le tudja kezelni)
  - $\forall a \in D_F : p(S)(a) \subseteq F(a)$ (program végállapotja, bármilyen feladatban értelmezett kezdőállapot esetén, részhalmaza a feladat "megoldáshalmazának")

2.6. Parciális helyesség

- Nem hasznos, nem nagyon fogjuk használni.
- Gyenge programfüggvény:
  - $\widetilde{p}(S) \subseteq A \times (A \cup \{fail\})$ reláció
  - $S \subseteq A \times (\overline{A} \cup \{fail\})^{**}$ programhoz való
  - $D_{\widetilde{p}(S)} = \{a \in A | S(a) \cap (\overline{A} \cup \{fail\})^* \neq \varnothing\}$ (ott értelmes, ahol S ad véges végállapotot is)
    - fail-t megenged, de végtelent nem
    - értelmezve van, ha azonos bemenethez van végtelen és véges megoldás is
  - $\forall a \in D_{\widetilde{p}(S)} : \widetilde{p}(S)(a) =$
    $\{b \in A \cup \{fail\} | \exists vs \in S(a) \cap (\overline{A} \cup \{fail\})^* : b = vs_{|vs|}\}$
    (azaz $\widetilde{p}(S)(x) =$ 'S program lehetséges végállapotjai (halmaza) x kezdőállapot esetén, csak véges vs-eket nézve')
- Parciális helyesség (S program parciálisan helyes az F feladatra nézve):
  - ha $\forall a \in D_F : \widetilde{p}(S)(a) \subseteq F(a)$
  - szóval végtelen vs-ek kivételével: mint a rendes megoldás

2.7. Elemi programok

- Elemi program definíció: $\forall a \in A : S(a) \subseteq$
  $\{< a >, < a, fail >, < a, a, a, ... >, < a, b > | b \in A\}$
- Nevezetes elemi programok:
  - $\forall a \in A : ABORT(a) = \{< a, fail >\}$
  - $\forall a \in A : SKIP(a) = \{< a >\}$
  - értékadás (x := 1)
  - szimultán értékadás (x,y := y,x)
  - értékkiválasztás (x :∈ {n-1, n-2})

2.8. Programok, programfüggvények, stb. relációi

2.8.1. $S_1 \subseteq S_2 \implies D_{p(S_2)} \subseteq D_{p(S_1)}$

Bizonyítás:
$a \in D_{p(S_i)} \Leftrightarrow S_i(a) \subseteq \overline{A}^*$ (def)
$S_1 \subseteq S_2 \implies S_1(a) \subseteq S_2(a)$
$(S_2(a) \subseteq \overline{A}^*) \wedge (S_1(a) \subseteq S_2(a)) \implies (S_1(a) \subseteq (S_2(a) \subseteq \overline{A}^*)$

2.8.2. $S_1 \subseteq S_2 \implies \forall a \in D_{p(S_2)} : p(S_1)(a) \subseteq p(S_2)(a)$

Bizonyítás:
$x \in p(S_i)(a) \Leftrightarrow \exists vs \in S_i(a) : vs_{|vs|} = x$ (def)
$S_1 \subseteq S_2 \implies S_1(a) \subseteq S_2(a)$
$(\exists vs \in S_1(a) : vs_{|vs|} = x) \wedge (S_1(a) \subseteq S_2(a)) \implies (\exists vs \in S_2(a) : vs_{|vs|} = x)$

2.8.3. $S' \subseteq S$ és S megoldja F-et, akkor S' is

Lásd előző kettő bizonyítás:
$D_F \subseteq D_{p(S)} \subseteq D_{p(S')}$
$p(S')(a) \subseteq p(S)(a) \subseteq F(a)$

2.8.4. $F \subseteq F'$ és S megoldja F'-t, attól még S nem oldja meg F-t

Példa: F' nem determinisztikus, F kevesebb megoldást fogad el.

2.8.5. Determinisztikusság relációk

F, S és p(S) determinisztikussága nagyrészt (vagy egyáltalán?) nem következtethető ki egymásból.

2.8.6. Szigorúbb feladat

- $F_2 \subseteq A \times A$ feladat szigorúbb, mint $F_1 \subseteq A \times A$, ha:
  - $D_{F_1} \subseteq D_{F_2}$ (több helyen értelmezett)
  - $\forall a \in D_{F_1} : F_2(a) \subseteq F_1(a)$ (kevesebb jó kimenet)
  - Tehát egy feladat önmagának is a szigorítása
- Állítás: ha S program megoldja F_2-t, akkor F_1-et is
- Bizonyítás: $D_{F_1} \subseteq D_{F_2} \subseteq D_{p(S)}$
  és $\forall a \in D_{F_1} : p(S)(a) \subseteq F_2(a) \subseteq F_1(a)$

3. Logikai függvények

- Legyen $R \in A \to \mathbb{L}$, ekkor $\lceil R \rceil = \{a \in A | R(a) = igaz\}$
- Igazsághalmaz: $\lceil R \rceil$
- D_R nem feltétlenül A ($\in$ jelölés miatt), tehát attól még, hogy nem igaz valahol, nem feltétlenül hamis ott. Éppen ezért egy ilyen logikai függvényt nem lehet megadni csak az igazságtáblájával.
  - Viszont megegyezés miatt igazságtáblával megadott logikai függvény mindenhol értelmezve van.
- $R : A \to \mathbb{L}$ viszont már a teljes A-n értelmezve van
- Nevezetes logika függvények:
  - $\forall a \in A : IGAZ(a) = \{igaz\}$
    $\lceil IGAZ \rceil = A$
  - $\forall a \in A : HAMIS(a) = \{hamis\}$
    $\lceil HAMIS \rceil = \varnothing$
- Példa: $A = (x : \mathbb{N})$, $R = (1 < x < 3)$

3.1. Műveletek logika függvényekkel

- Legyen A felett Q, R logikai függvények
- $Q \wedge R = \{a \in A | Q(a) \wedge R(a)\}$
- $Q \vee R = \{a \in A | Q(a) \vee R(a)\}$

3.2. Következik reláció, maga után vonás

- $Q, R \in A \to \mathbb{L}$ és $A \neq \varnothing$
- $(Q \implies R) \Leftrightarrow (\lceil Q \rceil \subseteq \lceil R \rceil)$
  - ha Q(x) igaz, akkor R(x) is
- Vegyünk tetszőleges $P \in A \to \mathbb{L}$ függvényt
  - $P \implies IGAZ$
  - $HAMIS \implies P$

4. Leggyengébb előfeltétel

- Logikai függvény, amely pontosan ott igaz, ahonnan kiindulva S hibátlanul terminál és az összes lehetséges végállapotban R igaz.
- Legyen $S \subseteq A \times (\overline{A} \cup \{fail\})^{**}$ program és $R \in A \to \mathbb{L}$ logika függvény
- $lf(S,R) : A \to \mathbb{L}$
  - tehát minden A-beli ponthoz (pontosan egy) logikai értéket rendel
- $\lceil lf(S,R) \rceil = \{a \in A | a \in D_{p(S)} \wedge p(S)(a) \subseteq \lceil R \rceil\}$
- Leggyengébb, mert $P \implies lf(S,R)$ esetén P szigorúbb, mint $lf(S,R)$

4.1. lf tulajdonságai

- Legyen A felett S egy program és Q, R logikai függvények
- Általános tulajdonságok:
  - $lf(S, HAMIS) = HAMIS$
  - ha $Q \implies R$, akkor $lf(S,Q) \implies lf(S,R)$
  - $lf(S,Q) \wedge lf(S,R) = lf(S, Q \wedge R)$
  - $lf(S,Q) \vee lf(S,R) \implies lf(S, Q \vee R)$
  - $\lceil lf(S, IGAZ) \rceil = D_{p(S)}$
- Elemi függvényekkel:
  - $\lceil lf(ABORT, R) \rceil = \varnothing$
  - $lf(ABORT, R) = HAMIS$
  - $\lceil lf(SKIP, R) \rceil = \lceil R \rceil$
  - $lf(SKIP, R) = R$

4.2. lf kiszámolása

- $\lceil lf(S,R) \rceil = \{a \in A \mid a \in D_{p(S)}$ és behelyettesítjük R-be S-t $\}$
- pl.: $\lceil lf((x := x+1), (x < 10)) \rceil = \{a \in A \mid igaz \wedge (x < 10)^{\leftarrow x := x+1}\} = \{a \in A \mid x' = x(a) + 1 \wedge x' < 10\} = \{a \in A \mid x(a) + 1 < 10\} = (x < 9)$

5. Paramétertér, specifikáció tétele

5.1. Paramétertér

- B az $F \subseteq A \times A$ feladat paramétertere, ha
  - létezik $F_1 \subseteq A \times B$ és $F_2 \subseteq B \times A$
  - hogy $F = F_2 \circ F_1$
- Minden feladatnak létezik paramétertere:
  legyen $B = A$, $F_1 = id$ és $F_2 = F$
- Minden feladatnak végtelen sok paramétertere van
- paraméter: paramétertér egy eleme

5.2. Specifikáció tétele

- Legyen:
  - $S \subseteq A \times (\overline{A} \cup \{fail\})^{**}$ egy program
  - $F \subseteq A \times A$ egy feladat
  - B az F egy paramétertere
    - $F_1 \subseteq A \times B$ és $F_2 \subseteq B \times A$
    - $F = F_2 \circ F_1$
- Definiáljunk kettő $A \to \mathbb{L}$ logikai fv-t minden paraméterhez $(\forall b \in B)$
  - $\lceil Q_b \rceil = F_1^{-1}(b)$ (olyan a-k, amikhez F_1 rendel egy b-t)
  - $\lceil R_b \rceil = F_2(b)$ (olyan a-k, amiket F_2 rendel egy b-hez)
- Ekkor: $(\forall b \in B : Q_b \implies lf(S, R_b)) \implies (S$ megoldja F-et)
- Szóval ez egy elégséges, de nem szükséges feltétel
- Előfeltétel gyengítésével és/vagy utófeltétel szigorításával egy *szigorúbb feladat*ot kapunk.

5.2.1. Bizonyítás: $(\forall b \in B : Q_b \implies lf(S, R_b)) \implies (S$ **mo.** F**-et)**

- $D_F \subseteq D_{p(S)}$
  - $a \in D_F \implies \exists b \in B : (a, b) \in F_1 \implies a \in \lceil Q_b \rceil$
  - $\lceil lf(S, R_b) \rceil = \{a \in D_{p(S)} | p(S)(a) \subseteq \lceil R_b \rceil\} \subseteq D_{p(S)}$
  - $(Q_b \implies lf(S, R_b)) \implies \lceil Q_b \rceil \subseteq \lceil lf(S, R_b) \rceil \subseteq D_{p(S)}$
  - beláttuk, hogy $a \in D_F \implies a \in \lceil Q_b \rceil \subseteq D_{p(S)}$
- $\forall a \in D_F : p(S)(a) \subseteq F(a)$
  - $\lceil lf(S, R_b) \rceil = \{a \in D_{p(S)} | p(S)(a) \subseteq \lceil R_b \rceil\}$
    - tehát $p(S)(a) \subseteq \lceil R_b \rceil$
  - $F_2(b) \subseteq F(a)$
  - beláttuk, hogy $p(S)(a) \subseteq \lceil R_b \rceil = F_2(b) \subseteq F(a)$

5.3. Feladat specifikációja

$F(n, *) = \{(n, d) \mid d|n\}$
$A = (n : \mathbb{N}^+, d : \mathbb{N}^+)$ (állapottér)
$B = (n' : \mathbb{N}^+)$ (paramétertér)
$\forall b \in B : Q_b(a) = (n(a) = n'(b))$
$\forall b \in B : R_b(a) = (n(a) = n'(b) \wedge d(a)|n(a))$
Rövidebben:
$Q = (n = n')$ (előfeltétel)
$R = (Q \wedge d|n)$ (utófeltétel)

5.4. Példa

Adott egy egész szám. Növeljük meg 1-gyel!
$A = (x : \mathbb{Z})$
$B = (x' : \mathbb{Z})$
$Q = (x = x')$
$R = (x = x' + 1)$
Helyesség belátása:
$S = (x := x + 1)$, tehát ezt kell belátni: $Q \implies lf(x := x + 1, R)$
R-ben helyettesítés: $Q \implies (x + 1 = x' + 1)$
Azaz: $(x = x') \implies (x + 1 = x' + 1)$

6. Új jelölések

- Ha egy $l : \mathbb{L}$ igaz, akkor *valami*: $l \to valami$
- Intervallum: $\forall i \in [1..n] : ...$
- Függvény: $f : \mathbb{Z} \to \mathbb{Z}$
- Tömb: $x : \mathbb{Z}^n$, ahol $n \in \mathbb{N}$ (n-t nem kell külön bevezetni)
  - 1-től indexelünk, meghozzá így: $x[1]$

7. Programkonstrukciók

7.1. Szekvencia

- Legyen S_1 és S_2 program A állapottéren
- $(S_1; S_2)$ reláció a két program szekvenciája
- $(S_1; S_2)(a) =$
  $\{vs \in \overline{A}^{\infty} \mid vs \in S_1(a)\}$ (S_1 végtelen vs-t ad)
  $\cup\{vs \in (\overline{A} \cup \{fail\})^* \mid vs \in S_1(a) \wedge vs_{|vs|} = fail\}$ (S_1 vs vége fail)
  $\cup\{vs \in (\overline{A} \cup \{fail\})^{**} \mid vs = x \oplus y \wedge x \in S_1(a) \wedge |x| < \infty \wedge x_{|x|} \neq fail \wedge y \in S_2(x_{|x|})\}$ (S_1 vs véges és nem fail, ekkor S_2-vel folytatjuk)
- Nem adom meg a szekvencia állapotterét, csak a két programot: több lehetséges alap állapottér van, mert lehetnek segédváltozók.
- TODO ezeket ellenőrizni (hivatalos jegyzetben nincsenek benne)
  - $p((S_1; S_2)) = p(S_2) \odot p(S_1)$
  - $D_{p((S_1;S_2))} = \{a \in D_{p(S_1)} \mid p(S_1)(a) \subseteq D_{p(S_2)}\}$

7.1.1. Szekvencia mint megoldó program

- S_1 megoldja F_1-t, S_2 megoldja F_2-t, S legyen $(S_1; S_2)$
- S nem oldja meg $F_2 \circ F_1$-t (F_1 legyen tágabb, stb... bonyolult)
- S megoldja $F_2 \odot F_1$-t (hosszú a bizonyítás, gyakon nem is volt MÉG)

7.2. Elágazás

- Legyen $S_1, S_2, ..., S_n$ programok, $\pi_1, \pi_2, ..., \pi_n$ logikai függvények A felett
  - azaz $\pi_i \in A \to L$, nem pedig $\pi_i : A \to L$
- $IF = IF(\pi_1 : S_1, ..., \pi_n : S_n)$
  - $\forall a \in A : IF(a) = w_0(a) \cup \bigcup_{i=1..n} w_i(a)$
  - $w_0(a) = \begin{cases} \{< a, fail >\} \text{ ha } \forall i \in [i..n] : (a \in D_{\pi_i} \wedge \neg \pi_i(a)) \\ \varnothing \text{ különben} \end{cases}$
  - $w_i(a) = \begin{cases} S_i(a) \text{ ha } a \in D_{\pi_i} \wedge \pi_i(a) \\ \varnothing \quad \text{ ha } a \in D_{\pi_i} \wedge \neg \pi_i(a) \\ \{< a, fail >\} \text{ ha } a \notin D_{\pi_i} \end{cases}$
- Megjegyzések
  - több π_i egyszerre teljesül: mindet belerakjuk a halmazba
  - π_i valahova nem rendel semmit: fail-t is rendelünk oda (de a vs első eleme a valami) (Emlékeztető: igazsághalmazzal megadott logikai fv megegyezés szerint mindenhova rendel valamit.)
  - összes π_i egyszerre hamis: fail-t rendelünk oda (de a vs első eleme a valami)

7.2.1. Programfüggvény

- $D_{p(IF)} = \{a \in A \mid a \in \bigcap D_{\pi_i} \wedge a \in \bigcup \lceil \pi_i \rceil \wedge (a \in \lceil \pi_i \rceil \implies a \in D_{p(S_i)})\}$
- $p(IF)(a) = \bigcup_{i=1..n} \{x \in p(S_i)(a) \mid a \in \lceil \pi_i \rceil \}$

7.3. Ciklus

- Legyen S program, π logikai függvény A felett
- $DO(\pi : S)$ az elöl tesztelő ciklus (S ciklusmag, π ciklusfeltétel)
  - $DO(a) = \begin{cases} (S; DO)(a) & \text{ha } a \in D_\pi \wedge \pi(a) \\ < a > & \text{ha } a \in D_\pi \wedge \neg\pi(a) \\ < a, fail > & \text{ha } a \notin D_\pi \end{cases}$
  - végtelen ciklus jelölése:
    - $DO(\pi : S) = \{1 \rightarrow < 1, 1, 1... >\}$
    - $DO(\pi : S) = \{1 \rightarrow < 1, (2, 3, 1)^\infty ... >\}$
    - $DO(\pi : S) = \{1 \rightarrow < 1, (2, 3, 1)^*, 4 >\}$ (*: véges sokszor)
- Programfüggvények:
  - TODO $p(DO)$ és $D_{p(DO)}$ =?
  - $D_{p(S)} \not\subseteq D_{p(DO)}$ (mert: IGAZ és SKIP)
  - $D_{p(DO)} \not\subseteq D_{p(S)}$ (mert: HAMIS és ABORT)

8. Levezetési szabályok

8.1. Szekvencia levezetési szabálya

- Legyen $S = (S_1; S_2)$ program és Q, Q', R logikai függvény
- Ha $Q \implies lf(S_1, Q')$ és $Q' \implies lf(S_2, R)$
- Akkor (és csak akkor) $Q \implies lf(S, R)$
- Magyarázat: $Q \implies lf(S, R)$ jelentése: S elvisz Q-ból R-be
  - Szóval: $Q \to (S_1) \to Q' \to (S_2) \to R$

8.2. Elágazás levezetési szabálya

- Legyen $IF = IF(\pi_1 : S_1, ..., \pi_n : S_n)$ és Q, R logikai függvény
- Ha
  - $Q \implies AND_{i=1}^{n} \pi_i \vee \neg\pi_i$ (mindegyik mindenhol értelmes)
  - $Q \implies OR_{i=1}^{n} \pi_i$ (mindenhol legalább egy teljesül)
  - $\forall i \in [1..n] : Q \wedge \pi_i \implies lf(S_i, R)$ (bármelyik ágon R igaz lesz)
- Akkor $Q \implies lf(IF, R)$

8.3. Ciklus levezetési szabálya

- Legyen $DO = DO(\pi : S)$ és P, Q, R logikai függvény, ahol P invariáns és $t : A \to \mathbb{Z}$ terminálo állítás
- Ha
  - $Q \implies P$ (invariáns alapból teljesül)
  - $P \wedge \neg\pi \implies R$ (jó helyen áll meg)
  - $P \implies \pi \vee \neg\pi$ (ciklusfeltétel nem abortál)
  - $P \wedge \pi \implies t > 0$ (cikluson belül van még lépés)
  - $P \wedge \pi \wedge t = t_0 \implies lf(S, P \wedge t < t_0)$ (invariáns marad, lépés - -)
- Akkor $Q \implies lf(DO, R)$
- P az invariáns állítás, t a terminálófüggvény (hány lépés van hátra)

9. Helyességbizonyítás

9.1. Módszere

- Be akarjuk látni, hogy S megoldja az A, B, Q, R specifikációjú feladatot.
- 1. lépés: ST (Specifikáció Tétele) miatt elég belátni: $Q \implies lf(S, R)$
- 2. lépés: fenti állítás belátása, S-től függően:
  - értékadás/kiválasztás: közvetlenül belátható (R-be behelyettesítés)
  - szekvencia: SZLSZ (Szekvencia Levezetési Szabálya) használata
  - elágazás: ELSZ (Elágazás Levezetési Szabálya) használata
  - ciklus: CLSZ (Ciklusz Levezetési Szabálya) használata
- A 2. lépésben rekurzívan ismételjük a 2. lépést, amíg be nem látunk minden állítást.

9.2. Megvalósítási tippek

- Belátok valamit, ami nem definícióból következik (pl. $1 < 2$): PIPA
- Értékkiválasztás: $lf(x :\in h, x\ even) = (h \neq \varnothing \wedge \forall e \in h : e\ even)$
- $Q \implies$ ... esetén vegyem úgy, hogy Q igaz és lássam be ...-ot
- $(\forall k \in [1..n] : x[k] = ...)$-on $(x[i] = ...)$ végrehajtása: $\forall k$ 3 részre bontása: $(\forall k \in [1..i-1] : ...)$ és $(x[i] = ...)$ és $(\forall k \in [i+1..n] : ...)$
- $\vee$ (vagy)-ból következés: esetszétválasztás, akár negált is felhasználható, pl.: $(a \vee b \implies ...)$-nél: $(a \implies)$ vagy $(\neg a \wedge b \implies)$
- Értékadások és számolások, amik abortálhatnak:
  - $a := b$ ahol $a : A$ és $b : B$ abortál, ha $B \not\subseteq A$
  - $a \mod b$ abortál, ha $b = 0$
  - $x[i]$ ahol $x : \mathbb{Z}^n$ abortál, ha $i \notin [1..n]$

10. Pszeudokód jelölések

- Ciklus: *while VALAMI do* [...] *od*

11. Párhuzamos programok

- Értékadás és feltétel kiértékelés nem megszakítható (atomikus)

11.1. Annotációk (pl. $\{Q_7\}$)

- Egy program egy adott pontjához egy logikai függvény: elő/utófeltétel.
- Segédváltozókat is felvehetünk ennek érdekében. Ezzel a program megváltozik, de a helyessége nem: a segédváltozók nem módosítjak a nem-segédeket.

11.2. Atomi program

- $[S](a) = S(a)$
- Atomikus: nem megszakítható
- S nem tartalmazhat sem ciklust, sem várakoztató utasítást (?)
- Levezetési szabály:
  - Ha $Q \implies lf(S, R)$
  - Akkor $Q \implies lf([S], R)$

11.3. Várakoztató utasítás

- $\beta \in A \to \mathbb{L}$ (őrfeltétel)
- Stukis jelölés: szekvencia, ahol a felső rész felet boltív van ($\cap$)
- $(await\ \beta\ then\ S\ ta)(a) =$
  - ha $a \in D_\beta \wedge \beta(a)$ akkor $[S](a)$ (β és S együtt atomikus)
  - ha $a \in D_\beta \wedge \neg\beta(a)$ akkor $(skip, await\ \beta\ then\ S\ ta)(a)$
  - ha $a \notin D_\beta$ akkor $< a, fail >$
- Levezetési szabály
  - Ha
    - $Q \implies \beta \vee \neg\beta$ (β kiértékelhető)
    - $Q \wedge \beta \implies lf(S, R)$
  - Akkor $Q \implies lf(await\ \beta\ then\ S\ ta, R)$
- S nem tartalmazhat sem ciklust, sem várakoztató utasítást (?)

11.4. Párhuzamos blokk

- S_i: komponens programok
  - vagy S_i atomikus (nem megszakítható)
  - vagy $S_i = (u_i, T_i)$ ahol u_i az első utasítás, T_i a többi
- Stukis jelölés: $x := x + 2 \;||\; x := 0$
- $(parbegin\ S_1||...||S_n\ parend)(a) = \bigcup_{i=1}^{n} B_i(a)$
- $B_i(a) =$
  - ha S_i atomikus: $(S_i; parbegin\ S_1||...||S_{i-1}||S_{i+1}||...||S_n\ parend)(a)$
  - ha $S_i = (u_i, T_i)$: $(u_i; parbegin\ S_1||...||S_{i-1}||T_i||S_{i+1}||...||S_n\ parend)(a)$
- Holtpontra jutott, ha minden be nem fejeződött komponense
  - vagy egy várakoztató utasítás, aminek őrfeltétele hamis (azaz blokkolt)
  - vagy egy szekvencia, amelyben van blokkolt várakoztató utasítás
- Levezetési szabály: (PLSZ)
  - Legyenek Q_i és R_i $(i = [1..n])$ logikai függvények (annotációk)
  - Ha
    - $Q \implies Q_1 \wedge ... \wedge Q_n$ (belépési feltétel)
    - $R_1 \wedge ... \wedge R_n \implies R$ (jó helyen állunk meg)
    - $\forall i \in [1..n] : Q_i \implies lf(S_i, R_i)$
    - $Q_i \implies lf(S_i, R_i)$ teljes helyességi interferencia formulák interferenciamentesek
    - párhuzamos blokk holtpontmentes
  - Akkor $Q \implies lf(parbegin\ S_1||...||S_n\ parend, R)$

11.5. Kritikus utasítás

- Vagy egy értékadás
- Vagy egy atomikusan végrehajtott program, ami tartalmaz értékadást
  - A várakoztató utasítás atomikus!!!

11.6. Interferenciamentesség

- Legyen u egy pre_u előfeltételű kritikus utasítás S_i-ben
- Legyen s egy tetszőleges, pre_s előfeltételű utasítás S_j-ben (S_j tetszőleges)
- u nem interferál a $Q_j \implies lf(S_j, R_j)$ teljes helyességi formulával ha:
  - $pre_u \wedge R_j \implies lf(u, R_j)$ (u végrehajtása nem rontja el R_j-t)
  - $pre_u \wedge pre_s \implies lf(u, pre_s)$ (u végrehajtása nem rontja el pre_s-t)
  - $pre_u \wedge t = t_0 \implies lf(u, t \leq t_0)$
    ahol t egy tetszőleges S_j-beli ciklus terminálófüggvénye
    (u végrehajtása miatt nem "számol" visszafelé ciklus)
- Gyakorlatban:
  - $Q_1 = (x = 0)$; $Q_2 = igaz$
  - $R_1 = R_2 = (x = 0) \vee (x = 2)$
  - be kell látni: $1 \rightsquigarrow 2 : \{Q_1\}x := x + 2 \rightsquigarrow Q_2, R_2$
    - első rész: $R_2 \wedge Q_1 \implies lf(x := x + 2, R_2)$
    - második rész: $Q_2 \wedge Q_1 \implies lf(x := x + 2, Q_2)$
  - be kell látni: $2 \rightsquigarrow 1 : \{Q_2\}x := 0 \rightsquigarrow Q_1, R_1$
    - Ha 1 ciklus lenne: $\{Q_2\}x := 0 \rightsquigarrow Q_1, R_1, P$ és fenti t-s
    - Ha nincs közös változó: "nincs krit. ut. az 1. ágra nézve"
  - Alternatíva: "elég" belátni: a kritikus utasítás egy annotációt sem ront el, valamint a ciklus terminálófüggvény sem számol felfelé.
  - Emlékeztető: hamis bármit implikál ($\implies$ előtt ellentmondás)

11.7. Elégséges feltétel holtpontmentességre

- Legyen m *await* és n párhuzamos blokk egy S szekvenciában
  - A_m : *await* β_m *then* S_m *ta*
  - T_n : *parbegin* $S_1^n||...||S_{n_n}^n$ *parend*
- $D(S) = [\bigvee_{j=1}^{m}(pre(A_j) \wedge \neg\beta_j)] \vee [\bigvee_{k=1}^{n} D1(T_k)]$
  - vagy egy *await* miatt várok
  - vagy egy párhuzamos blokk van holtpontban
- $D1(T_k) = [\bigwedge_{j=1}^{n_k}(post(S_i^k) \vee D(S_i^k))] \wedge [\bigvee_{i=1}^{n_k} D(S_i^k)]$
  - az összes komponens már befejezedőtött vagy holtpontban van
  - és legalább egy komponens holtpontban van
- Ha $D(S) = HAMIS$, akkor biztosan nincs holtpont. Egyébként nem mondunk semmit.

11.7.1. Holtpontmentesség: egy await egy kétágú párhuzamos blokkban

- Legyen $S =$ *parbegin* (*await* β *then* S_1 *ta*) || S_2 *parend*
- Bizonyítás nélkül használható tény: csak akkor van ekkor holtpont ha $pre($*await* β *then* S_1 *ta*$) \wedge \neg\beta \wedge post(S_2)$
- Ha ez ellentmondást ad ($= hamis$), akkor nincs holtpont

11.7.2. Holtpontmentesség: két await egy kétágú párhuzamos blokkban

- Legyen $S =$ *parbegin* (*await* β_1 *then* S_1 *ta*) || (*await* β_2 *then* S_2 *ta*) *parend*
  - Átfogalamzva: $S =$ *parbegin* $A_1||A_2$ *parend*
- Bizonyítás nélkül használható tény: csak akkor van ekkor holtpont ha
  - vagy: $pre(A_1) \wedge \neg\beta_1 \wedge post(A_2)$
  - vagy: $pre(A_2) \wedge \neg\beta_2 \wedge post(A_1)$
  - vagy: $pre(A_1) \wedge \neg\beta_1 \wedge pre(A_2) \wedge \neg\beta_2$
- Ha ez ellentmondást ad ($= hamis$), akkor nincs holtpont

11.7.3. Holtpontmentesség: nincs await

- Mindig holtpontmentes.