

Folyamatok specifikációja:

A specifikációs feltételek a programra fogalmazzuk meg kikötéseket. Fontos különbséget tenni a specifikációs feltételek és a programtulajdonságok között: míg az előbbiek a program elkészülte előtt adottak – ezek alapján fogalmazzuk meg a feladatot – addig utóbbi a már elkészült program tulajdonságait jelenti.

Hétféle specifikációs feltételt fogalmazzunk meg. Az azonos feltételtípushoz tartozó feltételeket egy relációban gyűjtjük össze, így 7 specifikációs relációt tudunk leírni.

Legyen $P, Q, R, U : A \rightarrow \mathcal{L}$ logikai függvény.

$\triangleright, \rightarrow, \hookrightarrow \subseteq \mathcal{P}(A) \times \mathcal{P}(A)$ relációk és $FP, INIT, inv, TERM \subseteq \mathcal{P}(A)$ halmazok (vagyis unáris relációk).

A relációkhoz infix jelölések is tartoznak.

Jelölések:

- $P \triangleright Q ::= (\lceil P \rceil, \lceil Q \rceil) \in \triangleright$ (P stabil, feltéve, hogy nem Q)
- $P \rightarrow Q ::= (\lceil P \rceil, \lceil Q \rceil) \in \rightarrow$ (P biztosítja Q-t)
- $P \hookrightarrow Q ::= (\lceil P \rceil, \lceil Q \rceil) \in \hookrightarrow$ (P-ből elkerülhetetlen Q)
- $Q \hookrightarrow FP ::= \lceil Q \rceil \in TERM$ (Q-ből a program biztosan fixpontba jut)
- $FP \Rightarrow R ::= \lceil R \rceil \in FP$ (R teljesül fixpontban)
- $inv P ::= \lceil P \rceil \in inv$ (P invariáns)
- $Q \in INIT ::= \lceil Q \rceil \in INIT$ (Q igaz kezdetben)

A specifikációs relációkat 4 csoportra osztjuk aszerint, hogy milyen jellegű feltételt fogalmazzunk meg a segítségükkel. Alább láthatók ezek a csoportok

- $A P \triangleright Q$ és az $inv P$ feltételeket *biztonságossági feltételeknek* nevezzük.
 - $P \triangleright Q$: Ha a program egy állapotára teljesül a $P \wedge \neg Q$ feltétel, akkor a $P \triangleright Q$ azt jelenti, hogy a program vagy sose mozdul el innen, vagy ebből az állapotból közvetlenül Q-ba lép.
 - $inv P$ két dolgot jelent:
 - A program kezdeti értékadásakor P igazzá válik.
 - P stabil ($\Leftrightarrow P \triangleright \text{Hamis}$), azaz P-ből a program minden elemi lépése után P igaz marad.
- $P \rightarrow Q$ és $P \hookrightarrow Q$ *haladási feltételek*.
 - $P \rightarrow Q$ két dolgot jelent:
 - $P \triangleright Q$ teljesül.
 - A programnak van egy olyan utasítása, amely minden olyan állapottérbeli pontból, ahol P igaz, Q-ba visz.
 - $P \hookrightarrow Q$: A program P-ből előbb-utóbb Q-ba jut. $Q \hookrightarrow FP$ azt jelenti, hogy a program Q-ból előbb-utóbb biztosan fixpontba jut.
- Az $FP \Rightarrow R$ *fixpont feltétel* azt jelenti, hogy R-nek teljesülnie kell, amikor a program

fixpontba jut.

- $Q \in \text{INIT}$ *kezdeti feltétel*. Elegendő, ha ezen feltételnek megfelelő állapotokból kiindulva megfelelően működik a program.

Megjegyzés: P stabilitási feltétel, ha $P \triangleright \text{Hamis}$ (ld. feljebb). P konstans feltétel, ha P és $\neg P$ is stabilitási feltétel.

$\triangleright, \rightarrow, \hookrightarrow, \text{FP}, \text{INIT}, \text{inv}, \text{TERM}$ relációk a *specifikációs relációk*, elemei a *specifikációs feltételek*.
 $A \triangleright, \rightarrow, \hookrightarrow, \text{inv}, \text{TERM}$ relációk elemei az *átmenetfeltételek*, az INIT, FP relációk elemei pedig a *peremfeltételek*. Az INIT reláció a *környezeti előírások* csoportjába tartozik.

Az absztrakt program tulajdonságai:

A leggyengébb előfeltétel és általánosítása

Def.: Leggyengébb előfeltétel, legszigorúbb utófeltétel

Legyen s egy utasítás, $Q, R: A \rightarrow \mathcal{L}$ logikai fgv-ek. Az $\text{lf}(s, R): A \rightarrow \mathcal{L}$ logikai fgv. az R utófeltétel s utasításra vonatkozó leggyengébb előfeltétele, ahol:

- $\lceil \text{lf}(s, R) \rceil ::= \{a \in \mathcal{D}_{p(s)} \mid p(s)(a) \subseteq \lceil R \rceil\}$

Az $\text{sp}(s, Q): A \rightarrow \mathcal{L}$ logikai függvény a Q előfeltétel legszigorúbb utófeltétele az s -re nézve, ahol:

- $\lceil \text{sp}(s, Q) \rceil ::= p(s)(\lceil Q \rceil)$

Leggyengébb előfeltétel alaptulajdonságai:

- 1) $\text{lf}(s, \text{Hamis}) = \text{Hamis}$ (csoda kizárásának elve)
- 2) Ha $\mathcal{D}_{p(s)} = \lceil \text{Igaz} \rceil$, akkor $\text{lf}(s, \text{Igaz}) = \text{Igaz}$
- 3) $\lceil \text{lf}(s, R) \rceil = \lceil R \circ p(s) \rceil$
- 4) Ha $P \Rightarrow Q$, akkor $\text{lf}(s, P) \Rightarrow \text{lf}(s, Q)$ (monotonitás)
- 5) $\text{lf}(s, Q) \vee \text{lf}(s, R) \Rightarrow \text{lf}(s, Q \vee R)$
- 6) $\text{lf}(s, Q) \wedge \text{lf}(s, R) = \text{lf}(s, Q \wedge R)$

Def.: Leggyengébb előfeltétel általánosítása

$\text{lf}(S, R) ::= \forall s \in S: \text{lf}(s, R)$, ahol S egy ^{párhuzamos} program, „ $s \in S$ ” jelentése pedig, hogy s a S egy utasítása.

Általánosított leggyengébb előfeltétel alaptulajdonságai:

- 1) $\text{lf}(S, \text{Hamis}) = \text{Hamis}$
- 2) $\text{lf}(\text{s, Igaz}) = \text{Igaz}$ "nagy S"
- 3) Ha $P \Rightarrow Q$, akkor $\text{lf}(S, P) \Rightarrow \text{lf}(S, Q)$
- 4) $\text{lf}(S, Q) \vee \text{lf}(S, R) \Rightarrow \text{lf}(S, Q \vee R)$
- 5) $\text{lf}(S, Q) \wedge \text{lf}(S, R) = \text{lf}(S, Q \wedge R)$

Invariánsok és elérhető állapotok

Jelölés: $\text{inv}_S(\lceil Q \rceil)$: Azon logikai függvények igazsághalmaza, amelyek az S programra nézve invariánsok, ha S egy $\lceil Q \rceil$ -beli állapotból indul. $\lceil P \rceil \in \text{inv}_S(\lceil Q \rceil)$ - röviden írjuk így is: $P \in \text{inv}_S(Q)$.

Def.: Legszigorúbb invariáns

$\text{INV}_S(Q)$: azon P logikai függvények konjunkciója, amelyekre $P \in \text{inv}_S(Q)$.

Def.: Invariáns tulajdonság

- $\text{inv}_S : \mathcal{P}(A) \rightarrow \mathcal{P}(\mathcal{P}(A))$. $\text{inv}_S(\lceil Q \rceil) \subseteq \mathcal{P}(A)$.
- $\text{inv}_S(\lceil Q \rceil) ::= \{ \lceil P \rceil \mid P \Rightarrow \text{lf}(s_0, P) \text{ és } P \Rightarrow \text{lf}(s, P) \}$ "nagy S"

Megjegyzés: A definíció szerint $\forall S, Q: \text{Igaz} \in \text{inv}_S(Q)$.

Lemma:

Invariánsok konjunkciója zárt az „ \wedge ” műveletre nézve.

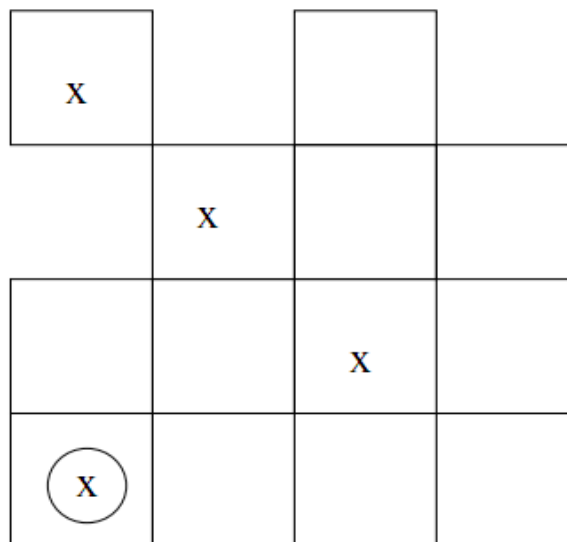
Az előbbi lemma **következménye**: Az $\text{inv}_S(Q)$ halmaznak egyértelműen létezik legkisebb eleme és az éppen $\text{INV}_S(Q)$. Ezt nevezzük a *legszigorúbb invariánsnak*.

Def.: Mindig igaz tulajdonság

- $\text{true}_S : \mathcal{P}(A) \rightarrow \mathcal{P}(\mathcal{P}(A))$. $\text{true}_S(\lceil Q \rceil) \subseteq \mathcal{P}(A)$.
- $\text{true}_S(\lceil Q \rceil) ::= \{ \lceil P \rceil \mid \text{INV}_S(Q) \Rightarrow P \}$

Megjegyzés: $\forall S, Q: \text{Igaz} \in \text{true}_S(Q)$.

Attól, hogy egy egy tulajdonság mindig igaz, még nem biztos, hogy invariáns. Ezt könnyű szemléltetni Horváth Zoltán tanár úr példájával:



Adott a fent látható tábla és egy huszár, amelyik a körrel jelölt mezőből indul és „L” alakban léphet (tehát úgy, mint a sakkban, lólépésben) és nem léphet le a tábláról. Könnyen látszik, hogy a huszár nem léphet olyan mezőre, amely nem „x”-szel jelölt. Azonban mégsem igaz, hogy invariáns tulajdonság lenne az, hogy a huszár mindig ezeken a jelölt mezőkön marad – ugyanis a bal felső sarokban levő (jelölt) mezőről léphet jelöletlen mezőre is. Mivel a program a körrel jelölt mezőről indítja a huszárt, ezért normális működés során ez a hiányosság nem tapasztalható. Csakis akkor, ha valami miatt a huszár a bal felső sarokban levő mezőre kerül (például egy konkurens program odalépteti).

Levonható a következtetés: az invariáns és a mindig igaz tulajdonság között a különbség a nem elérhető állapotoknál jelentkezik: egy mindig igaz állítás csak azt biztosítja, hogy az az elérhető állapotokban igaz lesz. Az invariáns viszont már az állapottérben lévő, de a program normális működése során nem elérhető állapotokban is. Azért fontos különbség ez, mert két program (komponens) párhuzamos futtatása eredményezheti azt, hogy valamely (vagy akár mindkét)

program számára elérhetővé válnak eddig nem elérhető állapotok. Tehát párhuzamos futtatás esetén csak a komponensek invariáns tulajdonságaira támaszkodhatunk.

Lemma: Az invariáns mindig igaz.

Formálisan: $\text{inv}_S(Q) \subseteq \text{true}_S(Q)$

Lemma: Mindig igaz állítások konjunkciója mindig igaz.

Formálisan: Ha $I \in \text{true}_S(Q)$ és $J \in \text{true}_S(Q)$, akkor $I \wedge J \in \text{true}_S(Q)$.

Következmény: Mindig igaz és invariáns konjunkciója mindig igaz.

Formálisan: Ha $I \in \text{inv}_S(Q)$ és $J \in \text{true}_S(Q)$, akkor $I \wedge J \in \text{true}_S(Q)$.

Következmény: A $\text{true}_S(Q)$ halmaznak egyértelműen létezik legkisebb eleme és ez éppen az $\text{INV}_S(Q)$.

Tétel: $\text{INV}_S(Q)$ igazsághalmaza éppen a $\lceil Q \rceil$ -ből elérhető állapotok halmaza.

Lemma: Ha $I \in \text{inv}_S(Q)$, $J \in \text{true}_S(Q)$ és $I \wedge J \Rightarrow \text{lf}(S, J)$, akkor $I \wedge J \in \text{inv}_S(Q)$.

Biztonságossági tulajdonságok

Def.: Stabil feltéve, hogy – tulajdonság

- $\triangleright_S \subseteq \mathcal{P}(A) \times \mathcal{P}(A)$
- $\triangleright_S ::= \{(\lceil P \rceil, \lceil Q \rceil) \mid (P \wedge \neg Q) \Rightarrow \text{lf}(S, (P \vee Q))\}$

Jelölés: $P \triangleright_S Q ::= (\lceil P \rceil, \lceil Q \rceil) \in \triangleright_S$

Megjegyzés: P tulajdonság S programban stabil, ha $P \triangleright_S$ Hamis.

Lemma (gyakorlaton: *stabilal kiegészítés tétele*) - stabilal metszésnek is nevezik

Ha $P \triangleright_S Q$ és $K \triangleright_S$ Hamis, akkor $P \wedge K \triangleright_S Q \wedge K$.

Bizonyítás:

Kifejtjük a két \triangleright_S -et:

- $P \triangleright_S Q$ miatt $(P \wedge \neg Q) \Rightarrow \text{lf}(S, (P \vee Q))$
- $K \triangleright_S$ Hamis miatt $(K \wedge \text{Igaz}) \Rightarrow \text{lf}(S, (K \vee \text{Hamis}))$ (a „Hamis” elhagyható a jobb oldalról, az „Igaz” pedig a bal oldalról)

$$P \wedge \neg Q \wedge K \Rightarrow \text{lf}(S, (P \vee Q)) \wedge \text{lf}(S, K) =$$

A fenti alakbra alkalmazhatjuk az általánosított lf „ \wedge ” műveletre vonatkozó alaptulajdonságát, így kapjuk azt az alakot, hogy:

$$= \text{lf}(S, (P \vee Q) \wedge K) = \text{lf}(S, (P \wedge K) \vee (Q \wedge K))$$

Ebből az alakból pedig a \triangleright_S definícióját alkalmazva megkapjuk a kívánt állítást.

Lemma: Minden invariáns stabil tulajdonság.

Formálisan: Ha $\exists Q: K \in \text{inv}_S(Q)$, akkor $K \triangleright_S$ Hamis.

Tétel: (\triangleright_S és az invariánsok szigoríthatósága)

Ha $(P \wedge J) \triangleright_S (Q \wedge J)$, $J \in \text{inv}_S(Q)$ és $K \in \text{inv}_S(Q)$, akkor:

- $J \wedge K \in \text{inv}_S(Q)$
Biz.: inv_S zárt az „ \wedge ” műveletre nézve
- $(P \wedge J \wedge K) \triangleright_S (Q \wedge J \wedge K)$

Biz.: $J, K \in \text{inv}_s(Q)$ miatt J, K stabil_s, tehát alkalmazható a stabillal való metszés tétele, amivel megkapjuk a kívánt állítást.

Tétel: (\triangleright_s és a legszigorúbb invariáns)

Ha $(P \wedge J) \triangleright_s (R \wedge J)$ és $J \in \text{inv}_s(Q)$, akkor:

- $P \wedge \text{INV}_s(Q) \triangleright_s R \wedge \text{INV}_s(Q)$

Biz.: $\text{INV}_s(Q) \in \text{inv}_s(Q)$, tehát alkalmazható az előző tétel. Azonban $\text{INV}_s(Q)$ definíciójából következik, hogy $\text{INV}_s(Q) \wedge J = \text{INV}_s(Q)$.

Haladási tulajdonságok

Def.: Biztosítja tulajdonság

- $\rightarrow_s \subseteq \mathcal{P}(A) \times \mathcal{P}(A)$
- $\rightarrow_s ::= \{(\lceil P \rceil, \lceil Q \rceil) \mid P, Q \in \triangleright_s \wedge \exists s \in S: (P \wedge \neg Q \Rightarrow \text{lf}(s, Q))\}$

Jelölés: $P \rightarrow_s Q ::= (\lceil P \rceil, \lceil Q \rceil) \in \rightarrow_s$

Lemma (stabillal kiegészítés tétele \rightarrow_s -re)

Ha $P \rightarrow_s Q$ és $K \triangleright_s$ Hamis, akkor $P \wedge K \rightarrow_s Q \wedge K$.

Bizonyítás:

- Az \rightarrow_s definíció \triangleright_s -re vonatkozó része következik a \triangleright_s -re vonatkozó stab. kieg. tételből.
 Be kell látni: $\exists s \in S: P \wedge K \wedge \neg(Q \wedge K) \Rightarrow \text{lf}(s, Q \wedge K)$
- Tudjuk: $\exists s \in S: P \wedge \neg Q \Rightarrow \text{lf}(s, Q)$. Legyen „s” egy ilyen utasítás. $K \Rightarrow \text{lf}(s', K)$ az összes s' $\in S$ -re (ugyanis K stabil_s), tehát a kiválasztott „s”-re is.
- Alkalmazva a két „s”-re vonatkozó állításra az „ \wedge ” műveletet: $P \wedge \neg Q \wedge K \Rightarrow \text{lf}(s, Q) \wedge \text{lf}(s, K)$. Ebből az alakból pedig az lf „ \wedge ” műveletre vonatkozó alaptulajdonsága miatt kapjuk, hogy:
- $\text{lf}(s, Q) \wedge \text{lf}(s, K) = \text{lf}(s, Q \wedge K)$

Tétel: (\rightarrow_s és az invariánsok szigoríthatósága)

Ha $(P \wedge J) \rightarrow_s (Q \wedge J)$, $J \in \text{inv}_s(Q)$ és $K \in \text{inv}_s(Q)$, akkor:

- $J \wedge K \in \text{inv}_s(Q)$
- $(P \wedge J \wedge K) \rightarrow_s (Q \wedge J \wedge K)$

Biz.: Mivel rendelkezésre áll az \rightarrow_s esetén is stab. kieg. tétel, ezért lehet ezt a tételt is ugyanolyan módon bizonyítani, mint a \triangleright_s hasonló tételét.

Tétel: (\rightarrow_s és a legszigorúbb invariáns)

Ha $(P \wedge J) \rightarrow_s (R \wedge J)$ és $J \in \text{inv}_s(Q)$, akkor:

- $P \wedge \text{INV}_s(Q) \rightarrow_s R \wedge \text{INV}_s(Q)$

Biz.: $A \triangleright_s$ -re vonatkozó tételhez hasonlóan.

Def.: Elkerülhetetlen tulajdonság

- $\hookrightarrow_s \subseteq \mathcal{P}(A) \times \mathcal{P}(A)$
- \hookrightarrow_s az \rightarrow_s tranzitív diszjunktív lezártja, vagyis:
 - $\rightarrow_s \subseteq \hookrightarrow_s$

- *Tranzitivitás:* $(\lceil P \rceil, \lceil Q \rceil) \in \hookrightarrow_s$ és $(\lceil Q \rceil, \lceil R \rceil) \in \hookrightarrow_s$, akkor $(\lceil P \rceil, \lceil R \rceil) \in \hookrightarrow_s$.
- *Diszjunkció:* Ha bármely W megszámlálható halmazra:
 $\forall m: (m \in W :: (\lceil P(m) \rceil, \lceil Q \rceil) \in \hookrightarrow_s)$, akkor $((\lceil \exists m: m \in W :: P(m) \rceil), \lceil Q \rceil) \in \hookrightarrow_s$

Jelölés: $P \hookrightarrow_s Q ::= (\lceil P \rceil, \lceil Q \rceil) \in \hookrightarrow_s$

Tehát $P \hookrightarrow_s Q$ pontosan akkor igaz, ha az ezekkel a szabályokkal levezethető.

Lemma: (\Rightarrow és \hookrightarrow_s)

Ha $\lceil P \rceil \subseteq \lceil Q \rceil$, akkor $(P, Q) \in \hookrightarrow_s$ tetszőleges S programra.

Biz.: $(P, P) \in \rightarrow_s$, így $(P, Q) \in \rightarrow_s$. Ebből a \hookrightarrow_s első képzési szabályának alkalmazásával kapjuk az állítást.

Lemma (stabillal kiegészítés tétele \hookrightarrow_s -re)

Ha $P \hookrightarrow_s Q$ és $K \triangleright_s \text{Hamis}$, akkor $P \wedge K \hookrightarrow_s Q \wedge K$.

Biz.: --viszonylag hosszú, szerintem nem kérik, a HZ-féle jegyzet 56. oldalán van--

Tétel: Az előbbi tétel általánosítása (PSP tétel relációs alakja) A bizonyítás strukturális indukcióval történik.

Ha $P \hookrightarrow_s Q$ és $R \triangleright_s B$, akkor $P \wedge R \hookrightarrow_s (Q \wedge K) \vee B$.

Tétel: (\hookrightarrow_s és az invariánsok szigoríthatósága)

Ha $(P \wedge J) \hookrightarrow_s (Q \wedge J)$, $J \in \text{inv}_s(Q)$ és $K \in \text{inv}_s(Q)$, akkor:

- $J \wedge K \in \text{inv}_s(Q)$
- $(P \wedge J \wedge K) \hookrightarrow_s (Q \wedge J \wedge K)$

Biz.: Mivel rendelkezésre áll az \hookrightarrow_s esetén is stab. kieg. tétel, ezért lehet ezt a tételt is ugyanolyan módon bizonyítani, mint a \triangleright_s hasonló tételét.

Tétel: (\hookrightarrow_s és a legszigorúbb invariáns)

Ha $(P \wedge J) \hookrightarrow_s (R \wedge J)$ és $J \in \text{inv}_s(Q)$, akkor:

- $P \wedge \text{INV}_s(Q) \hookrightarrow_s R \wedge \text{INV}_s(Q)$

Biz.: A \triangleright_s -re vonatkozó tételhez hasonlóan.

Lemma: (\hookrightarrow_s – jobb oldal gyengítése)

Ha $P \hookrightarrow_s Q$ és $Q \Rightarrow R$, akkor $P \hookrightarrow_s R$.

Biz.: $Q \Rightarrow R$, azaz $\lceil Q \rceil \subseteq \lceil R \rceil$, tehát alkalmazhatjuk a \Rightarrow és a \hookrightarrow_s kapcsolatra vonatkozó lemmát, amivel kapjuk, hogy $Q \hookrightarrow_s R$. Ezután már alkalmazhatjuk a \hookrightarrow_s tranzitivitási szabályát, amivel kapjuk az állítást.

Def.: (Elkerülhetetlen feltétlenül pártatlan ütemezés mellett)

$(P, Q) \in \hookrightarrow_s^1$ akkor és csak akkor, ha $\forall a \in P$ az S által az a -hoz rendelt fában bármelyik, a feltétlenül pártatlan ütemezésnek megfelelő végrehajtási úton véges (esetleg nem korlátos) távolságban van olyan pont, amelynek címkéje eleme Q halmaznak.⁷

Megjegyzés: HZ jegyzetében ez a „hullámos nyíl” jel kicsit máshogy néz ki – én most hirtelen ezt találtam, ami hasonlít.

Tétel: (\hookrightarrow_s helyessége és teljessége) $\hookrightarrow_s = \hookrightarrow_s^1$.

Biz.: --viszonylag hosszú, szerintem nem kéri, a HZ-féle jegyzet 57. oldalán van--

Következmény: Egy S program akkor és csak akkor rendelkezik a \leadsto_s tulajdonsággal, ha rendelkezik a \hookrightarrow_s tulajdonsággal is – amely levezethető a program $U \rightarrow_s V$ alakú tulajdonságaiból.

Fixpont tulajdonságok

A konkrét program az absztrakt program végrehajtásának prefixe. Egy absztrakt program sohasem terminál olyan értelemben, hogy nem hajt végre több elemi műveletet. Azonban egy izolált absztrakt programot terminálnak tekinthetünk, ha elérte egy fixpontját. Ez azt jelenti, hogy további műveletek hatására nem történhet állapotváltozás.

Az S programról azt mondjuk, hogy fixpontba jutott az A altér felett, ha az A altérhez tartozó változókra vonatkozó egyszerű értékadások mindegyikére teljesül, hogy az értékadás hatásrelációja független az altérhez nem tartozó állapotterkomponensektől és

- az értékadás determinisztikus és a jobb oldalán álló függvénykompozíciók értéke azonos a baloldalon álló változók értékével (tehát egyik változó értéke sem változik), vagy
- a feltételes értékadás feltétele hamis, vagy
- az értékadás nemdeterminisztikus és az altér azon részhalmaza felett, amely részhalmazban az értékadás determinisztikus és a feltétele igaz, ott a jobb oldalon álló függvénykompozíciók értéke azonos a baloldalon álló változók értékével.

Legyen $S=(s_0, \{s_1, \dots, s_n\})$, ahol $\forall s_j \in S$ egy (szimultán, nemdeterminisztikus) feltételes értékadás, $s_j: \parallel (v_i : \in F_{ji}(v_1, \dots, v_n), \text{ ha } \pi_{ji})$.

$i \in [1, n]$

Jelöljük $\pi_{j, id}$ -vel azt a logikai függvényt, melynek igazsághalmazára leszűkítve az F_{ji} reláció determinisztikus, azaz: $\pi_{j, id}(a) \Leftrightarrow (|F_{ji}(a)|=1)$.

Def.: (Fixpontok halmaza)

$\text{fixpont}_S ::= (\bigwedge_{j \in J, i \in [1..n]} (\neg \pi_{ji} \vee (\pi_{j, id} \wedge v_i = F_{ji}(v_1, \dots, v_n))))$.

Jelölés: φ_S

Ha az értékadások mindegyike determinisztikus, akkor a program fixpontjait jellemző logikai függvényt a következőképpen lehet megkapni:

- a szimultán értékadásokat átalakítjuk egyenlőséggé
- ezen értékadások feltételeit implikációvá alakítjuk, konjunkciót teszünk a feltételek közé és a kialakított egyenlőség elé tesszük.

Példa: $S=(\text{SKIP}, \{k := k+1, \text{ ha } k < N\})$

$\text{fixpont}_S = (k < N \rightarrow k=k+1) = (k \geq N)$

Def.: (Fixpont tulajdonság)

$\text{FP}_S ::= \{ \ulcorner R \urcorner \mid \text{fixpont}_S \Rightarrow R \}$.

Lemma: (Fixpont tulajdonság gyengítése)

Ha $R \Rightarrow Q$ és $R \in \text{FP}_S$, akkor $Q \in \text{FP}_S$.

Tétel: (A fixpontfeltétel finomítása)

Ha S megfelel az $inv_h P, FP_h \Rightarrow R$ specifikációs feltételnek és $P \wedge R \Rightarrow Q$, akkor S megfelel a $FP_h \Rightarrow Q$ specifikációs feltételnek is.

Biz.: Nem írom le, mert a záróvizsga tételben maga ez a tétel nincs is benne. Viszont azért mondtam itt ki, mert majd szükség lesz rá későbbi bizonyításhoz. A bizonyítás megtalálható a jegyzet 74. oldalán.

Terminálási tulajdonságok

Def.: (Biztosan fixpontba jut – tulajdonság)

$TERMS ::= \{ \lceil Q \rceil \mid Q \hookrightarrow_s \text{fixponts} \}$

Megjegyzés: A program biztosan fixpontba jut, ha egy alkalmasan megválasztott függvény, az ún. *variánsfüggvény* értéke bármely állapot elérése után a jövőben elkerülhetetlenül csökken.

Nevezetes feladatok megoldása párhuzamos és elosztott programokkal

Asszociatív művelet eredményének kiszámítása

Legyen H egy tetszőleges halmaz. $\circ : H \times H \rightarrow H$ tetszőleges kétoperandusú művelet, amelyről még azt tudjuk, hogy asszociatív.

Definiáljuk $f : H^* \rightarrow H$ függvényt. Jelentse $f a$ „ \circ ” művelet egyszeri, vagy ismételt alkalmazását.

Tehát $h_1 \circ h_2$ ekvivalens azzal, hogy $f(\langle\langle h_1, h_2 \rangle\rangle)$. f -et kiterjesztjük az egyelemű sorozatokra is: $f(\langle\langle h \rangle\rangle) = h$.

Adott $a \in H^*$: a véges, nem üres H -beli elemek sorozata. Tegyük fel, hogy a sorozat egyes elemei közvetlenül elérhetőek: $a = \langle\langle a_1, \dots, a_n \rangle\rangle$ ($n \geq 1$). Legyen $G_a : [1..n] \rightarrow H$ egy függvény, ahol $G_a(i) = f(\langle\langle a_1, \dots, a_i \rangle\rangle)$ ($i \in [1..n]$). A feladat, hogy számítsuk ki G_a értékét minden $i \in [1..n]$ -re.

A feladat specifikációja

Reprezentáljuk az a sorozatot egy vektorral, a G_a függvényt pedig transzformáljuk egy vektorba.

Jelöljük ezeket a -val és g -vel. Értelmszerűen ezen vektorok értékei H -beli értékek. Első lépésben a következő kikötéseket tesszük: fixpontban a g vektor i -edik eleme legyen $G_a(i)$, illetve a program biztosan elérje fixpontját.

$$A = G \times G \quad G = \text{vektor}([1..n], H), n \geq 1$$

$g \quad a$

$$B = G$$

a'

$$(a = a') \in \text{INIT}_{a'}$$

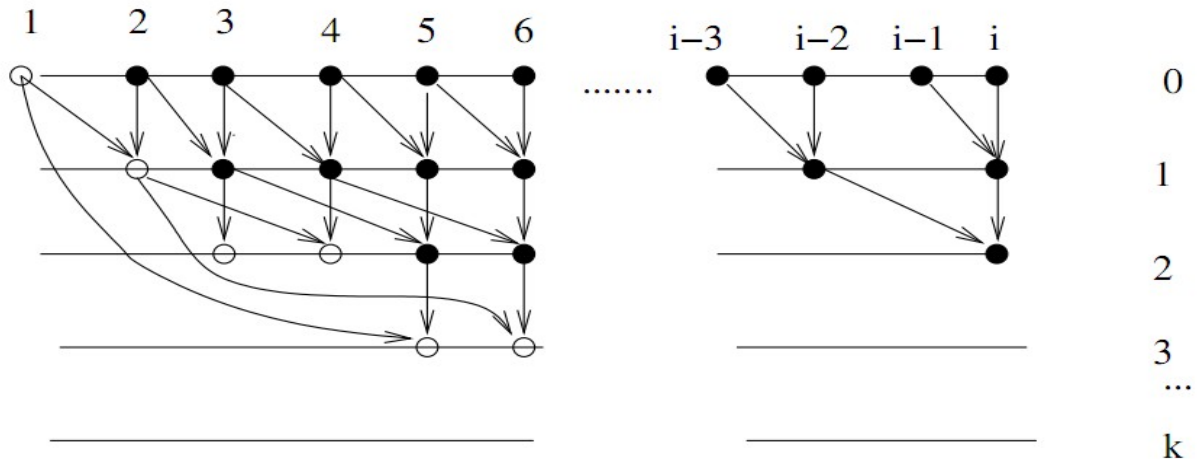
$$\text{Igaz} \hookrightarrow FP_{a'}$$

$$FP_{a'} \Rightarrow (a = a' \wedge \forall i \in [1..n] : g(i) = f(\langle\langle a_1, \dots, a_i \rangle\rangle))$$

Megállapíthatjuk, hogy a G_a függvény i helyen felvett értékének meghatározását megkönnyíti, ha ismerjük f értékét bármely $[u..v] \subseteq [1..i]$ intervallum elemeivel indexelt $\langle\langle a_u, \dots, a_v \rangle\rangle$ részsorozatra.

Megfigyelhető az is, hogy egy részsorozatra kapott eredményt bármely, a részsorozatot tartalmazó részsorozatra vonatkozó eredmény meghatározásánál hasznosíthatjuk.

Ezen gondolatmenet alapján bővítsük a feladat állapotterét és finomítsuk a specifikációt. Vezessük be h függvényt: $h(a,i,k)$ jelentse f értékét a azon részsorozatára, amelynek utolsó eleme a_i és hossza vagy éppen 2^k , vagy, ha $i < 2^k$, akkor első eleme a_1 . Ezt a függvényt (is) változóba kell transzformálni – ez a változó lesz a gs kétdimenziós vektor. A gs , k , t változók és a h függvény kapcsolatát invariáns állítással írjuk le a finomított specifikációban.



A fenti (Horváth Zoltán tanár úr jegyzetéből kivágott) ábra ábrázolja a gs mátrix elemei közti kapcsolatot.

A finomított specifikáció:

$$A' = \underset{g}{G} \times \underset{a}{G} \times \underset{gs}{GS} \times \underset{k}{K} \times \underset{t}{T}$$

$$G = \text{vektor}([1..n], H)$$

$$GS = \text{vektor}([1..n, 0..\lceil \log(n) \rceil], H)$$

$$K = \text{vektor}([1..n], \mathcal{N}_0)$$

$$T = \text{vektor}([1..n], \mathcal{N}_0, n \geq 1)$$

A K vektor a 2^k függvény változóba transzformálása (tehát $K(i) = 2^i$). A T vektor pedig azt mutatja, hogy a gs mátrix hanyadik soráig van kitöltve.

A h parciális függvény pontos definíciója:

$$h : G \times [1..n] \times \mathcal{N}_0 \rightarrow H$$

$$h(a,i,k) =$$

- $f(\langle a_1, \dots, a_i \rangle),$ ha $i - 2^k + 1 \leq 1$
- $f(\langle a_{(i-2^k+1)}, \dots, a_i \rangle),$ ha $i - 2^k + 1 \geq 1.$

Válasszuk a $v : A \rightarrow \mathcal{N}_0$ variánsfüggvényt a következőképpen:

$$v = 4 * n * n - \sum_{i=1}^n (k(i) + \chi(k(i) = \lceil \log(i) \rceil \wedge g(i) = gs(i, k(i)))),$$

ahol $\mathcal{X}: L \rightarrow \{0,1\}$. $\mathcal{X}(\text{igaz}) = 1$, $\mathcal{X}(\text{hamis}) = 0$. ν azt adja meg, hogy a gs mátrix egyes oszlopaiban összesen hány olyan elem van, amely nem azonos a h függvény megfelelő helyen felvett értékével, illetve, hogy a g vektor értéke hány helyen különbözik a G_a függvény értékétől.

Lemma: (Asszociatív művelet – a feladat finomítása)

Az alábbi specifikáció finomítása az eredetinek:

$$(a=a') \in INIT_{a'} \quad (8.4)$$

$$Igaz \hookrightarrow Fp_{a'} \quad (8.5)$$

$$FP_{a'} \Rightarrow \forall i \in [1..n] : (k(i) = \lceil \log(i) \rceil \wedge (g(i) = gs(i, \lceil \log(i) \rceil))) \quad (8.6)$$

$$inv_{a'}(\forall i \in [1..n] : k(i) \leq \lceil \log(i) \rceil \wedge \forall k : k \leq k(i) : gs(i,k) = h(a,i,k)) \quad (8.7)$$

$$inv_{a'}(\forall i \in [1..n] : t(i) = 2^{k(i)}) \quad (8.8)$$

$$inv_{a'}(a=a') \quad (8.9)$$

Biz.:

Fixpontban (8.6) szerint $k(i) \leq \lceil \log(i) \rceil$ és $g(i) = gs(i, \lceil \log(i) \rceil)$, tehát (8.7) alapján $g(i) = gs(i, \lceil \log(i) \rceil) = h(a,i, \lceil \log(i) \rceil)$. Mivel $2^{\lceil \log(i) \rceil} \geq i$, ezért így h definícióját felhasználva $h(a,i, \lceil \log(i) \rceil) = f(\langle \langle a_1, \dots, a_i \rangle \rangle)$, így a (8.9) invariáns tulajdonság és a *Fixpontfinomítás tétele* alkalmazásával igazolni tudjuk, hogy a (8.6), (8.7), (8.9) feltételek együttesen finomítják a (8.3) feltételt.

A (8.8) feltétel triviálisan igaz, hiszen így definiáltuk a t vektort.

Lemma: Ha $(i - 2^k \geq 1)$, akkor $f(\langle \langle h(a,i - 2^k, k), h(a,i, k) \rangle \rangle) = h(a,i, k + 1)$.

Biz.:

Tudjuk, hogy $i - 2^k \geq 1$, tehát $h(a,i, k) = f(\langle \langle a_{(i-2^k+1)}, \dots, a_i \rangle \rangle)$.

- Ha $(i - 2^k) - 2^k + 1 \geq 1$, akkor $h(a,i - 2^k, k) = f(\langle \langle a_{(i-2^k-2^k+1)}, \dots, a_{(i-2^k)} \rangle \rangle)$. Ekkor, mivel f asszociatív, ezért $f(\langle \langle h(a,i - 2^k, k), h(a,i, k) \rangle \rangle) = f(\langle \langle a_{(i-2^k-2^k+1)}, \dots, a_{(i-2^k)}, a_{(i-2^k+1)}, \dots, a_i \rangle \rangle) = h(a,i, k+1)$.
- Ha $(i - 2^k) - 2^k + 1 < 1$, akkor $h(a,i - 2^k, k) = f(\langle \langle a_1, \dots, a_{(i-2^k)} \rangle \rangle)$. Mivel f asszociatív, ezért $f(\langle \langle h(a,i - 2^k, k), h(a,i, k) \rangle \rangle) = f(\langle \langle a_1, \dots, a_{(i-2^k)}, a_{(i-2^k+1)}, \dots, a_i \rangle \rangle) = h(a,i, k + 1)$.

A megoldás

Tétel: (Asszociatív művelet kiszámításának tétele I.)

Az alábbi program megfelel a (8.4) – (8.9) specifikációnak, azaz megoldja az asszociatív művelet kiszámításának feladatát.

$$s0: \quad \square_{i=[1..n]} \quad gs(i,0), t(i), k(i) := f(\langle \langle a_i \rangle \rangle), 1, 0$$

$$S : \{ \quad \square_{i=[1..n]} \quad gs(i, k(i) + 1), t(i), k(i) :=$$

- $f(\langle \langle gs(i, k(i)), gs(i - t(i), k(i)) \rangle \rangle), 2 * t(i), k(i) + 1,$
 $\quad \quad \quad \text{ha } (i - 2 * t(i) + 1 \geq 1) \wedge$
 $\quad \quad \quad \wedge (k(i - t(i)) \geq k(i))$

- $$f(<<gs(i,k(i)), gs(i-t(i),k(i-t(i)))>>), 2*t(i),k(i) + 1,$$

$$\text{ha } (i-t(i) \geq 1) \wedge (i-2*t(i) + 1 < 1) \wedge$$

$$\wedge k(i-t(i)) = \lceil \log(i-t(i)) \rceil$$

$$\square \quad g(i) := gs(i,k(i)), \text{ ha } (k(i) = \lceil \log(i) \rceil)$$

$$i=[1..n]$$

Biz.: Elég hosszú, nem hiszem, hogy kérnék, a jegyzet 101. oldalán kezdődik.

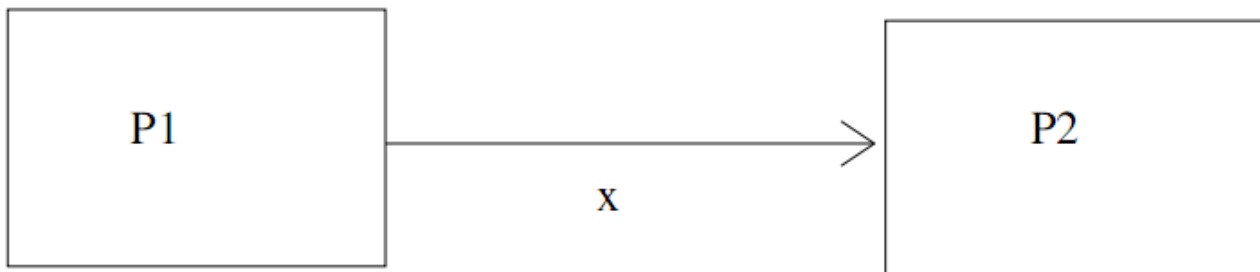
Adatcsatorna tétele

Bevezetés – Csatornaváltozók használata

A párhuzamos és elosztott rendszereket folyamathálózatok formájában írjuk le. A folyamatok üzenetek segítségével kommunikálnak egymással. Ezeket az üzeneteket egyirányú *csatornákon* keresztül juttatja el a feladó a címzett(ek)nek. Az üzenetküldés aszinkron, a feladó általában rögtön folytatja a tevékenységét az üzenet csatornára való elhelyezése után – az üzenet megérkezését a feladó tehát nem várja meg.

A csatornák sor típusú változóként viselkednek, átmenetileg képesek tárolni az elküldött, de még nem fogadott üzeneteket. Minden csatornához két sor típusú változó tartozik: az egyik a csatornán várakozó üzeneteket tartalmazza (a csatorna aktuális állapota), a másik a csatorna története.

A csatorna története minden olyan üzenetet tartalmaz (beérkezési sorrendben), amelyik valaha rákerült a csatornára. A történetváltozót egy felülvonással különböztetjük meg a csatorna állapotát jelző változótól (*mivel én ilyet nem tudok, ezért $x.h$ -val jelölöm egy x csatorna történetét*). A gyakorlatban a történetváltozó tárolása nehezen, vagy egyáltalán nem megoldható, ugyanis a történetváltozó mérete idővel minden korlátot meghaladhat. Csatornaváltozókat ezért csak a specifikációban használhatunk, a konkrét programban nem.



A fenti ábrán (Horváth Zoltán tanár úr ábrája) a $P1$ és $P2$ folyamatokat összekötő x csatorna látható. A $P1$ folyamat üzeneteket helyezhet el x -en, a $P2$ folyamat pedig ezeket az üzeneteket olvashatja ki belőle.

Az alábbi műveletek tartoznak a csatorna típusú változókhoz:

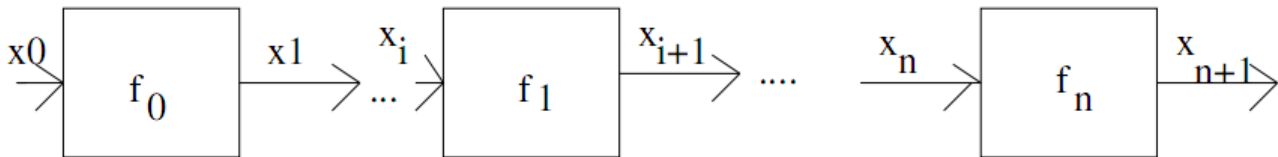
- üzenetküldés ($P1$): $x := hiext(x,e)$, vagy röviden: $x := x;e$
- üzenet eltávolítása ($P2$): $x := lorem(x)$, ha $x \neq \diamond$
- csatorna inicializálása: $x := \diamond$
- üzenet olvasása ($P2$): $x.lov$, ha $x \neq \diamond$
- lekérdezés, hogy a csatorna üres-e: $x = \diamond$
- lekérdezés, hogy hány üzenet várakozik a csatornán: $length(x)$, vagy $|x|$.

Adatcsatorna tétele

Adott egy $F = f_n \circ \dots \circ f_0$ függvénykompozíció és egy $D = \langle d_1, \dots, d_m \rangle$ sorozat, az argumentumok. A feladat az, hogy mindegyik d argumentumra meghatározzuk $F(d)$ -t, azaz $f_n(\dots(f_0(d))\dots)$ -t. Ez a tétel akkor alkalmazható hatékonyan, ha az adott feladatban $m \gg n$, azaz az argumentumok száma sokkal nagyobb, mint a függvénykomponensek száma. Illetve feltesszük azt, hogy az egyes függvénykomponensek kiszámításához szükséges idő lényegében azonos.

A tételben alkalmazott módszer alapötlete az, hogy az egyes függvénykomponensek kiszámítását egy-egy külön processzorra bízjuk. Először az első processzor megkapja az első adatot (d_1 -et), majd erre kiszámítja $f_0(d_1)$ -et. Ezt a kiszámított eredményt adja tovább az első a második processzornak, amely kiszámítja $f_1(f_0(d_1))$ -et, közben pedig az első processzor megkapja d_2 -t és kiszámolja $f_0(d_2)$ -t. n lépés elteltével az n -edik processzor is elkezd dolgozni ($f_n(\dots(f_1(f_0(d_1)))\dots)$ -et számolja) - az adatcsatorna „megtelik”. Ez azt jelenti, hogy innentől kezdve minden lépésben mindegyik processzor dolgozik és minden lépésben megkapjuk a soron következő d -re $F(d)$ -t (értelmszerűen mindig az n -edik processzortól kapjuk az új eredményt).

Ha a kommunikációs költségeket figyelmen kívül hagyjuk, arra a következtetésre juthatunk, hogy n processzossal egy m adatból álló argumentum-sorozatra $n+m$ lépésben elő tudjuk állítani a végeredményt. Ugyanez 1 processzossal $m \cdot n$ lépésbe telne (m darab adatra egyesével az összes f_i -t kellene meghatározni egymás után). Mivel $n \ll m$, ezért az $(n+m)/(n \cdot m)$ képletben a számlálóbeli n -et elhagyhatjuk, majd m -mel egyszerűsítve megkapjuk, hogy n processzossal kb. n -szer gyorsabban végzünk.



Adatcsatorna ábrázolása (Horváth Zoltán tanár úr ábrája).
Az x_0 a fő programszámból, az x_{n+1} a fő programszámba vezető csatorna.

A formális specifikáció során először csak annyit fogalmazunk meg, hogy egy F összetett függvény értékét kell elemenként meghatározni a D sorozatra. A D sorozat elemei az x_0 csatornán vannak kezdetben, új adat nem érkezik futás közben és fixpontban az eredmény, az $F(D)$ sorozat az x_{n+1} sorozat történetében található meg. Azért csak a történetre kötjük meg ezt, mert nem feltétlenül szeretnénk, hogy a főprogram megvárja az összes eredmény kiszámítását – lényegesen hatékonyabb, ha minden kiszámított eredményt rögtön átadunk a főprogramnak. Így csak azt kötjük ki, hogy a futás során az eredmény minden tagja előfordult már az x_{n+1} sorozatban.

$$A = \begin{matrix} Ch(a) & \times & Ch(a) & \times & Ch(a) & \times & Ch(a) \\ x_0 & & x_0.h & & x_{n+1} & & x_{n+1}.h \end{matrix}$$

$$B = \begin{matrix} Ch(a) & \times & Ch(a) & \times & Ch(a) & \times & Ch(a) \\ x'_0 & & x'_0.h & & x'_{n+1} & & x'_{n+1}.h \end{matrix}$$

$$Q ::= (x_0 = x_0.h = x'_0 = x'_0.h = D \wedge x_{n+1} = x_{n+1}.h = x'_{n+1} = x'_{n+1}.h = \langle \rangle)$$

$$Q \in INIT_h (h ::= x'_0 x'_0.h x'_{n+1} x'_{n+1}.h) \quad (8.24)$$

$$FP_h \Rightarrow x_{n+1}.h = F(x'_0.h) = F(D) \quad (8.25)$$

$$Q \in TERM_h \quad (8.26)$$

$$(x_0.h = x'_0.h = D) \in inv_h \text{ a teljes rendszerre} \quad (8.27)$$

A 8.24 kikötés szerint az x_0 és története kezdetben tartalmazza a D sorozatot. A 8.27 kikötés szerint az x_0 sorozat története nem is változhat, tehát újabb elem nem kerülhet x_0 csatornára. 8.24 szerint az x_{n+1} csatorna kezdetben üres. Fixpontban a 8.25 kikötés alapján megköveteljük, hogy x_{n+1} története $F(D)$ legyen. A 8.26 kikötés azt mondja ki, hogy a programnak mindenképpen terminálnia kell. A megoldás előállításához bővítjük az állapotteret (a fentebb látható adatcsatorna-ábrát alapul véve) az x_2, \dots, x_n csatornaváltozókkal és történetváltozókkal. A specifikációt finomítjuk a *Fixpontfinomítás tétele* alapján (8.28, 8.29), illetve variánsfüggvényt vezetünk be (8.29).

$$FP_h \Rightarrow \forall i \in [0..n] : x_i = \Diamond \quad (8.28)$$

$$\forall i \in [0..n] : (f_i(x_i.h - x_i) = x_{i+1}.h) \in inv_h \quad (8.29)$$

$$\text{variánsfüggvény: } (|x_0|, \dots, |x_n|) \quad (8.30)$$

A 8.30 pontban bevezetett variánsfüggvényt a következőképpen kell értelmezni: az egyes csatornák elemeszámból álló rendezett n -es elemeit helyiértékekkel súlyozzuk, így egy $n+1$ alapú számrendszerbeli számot kapunk.

A fixpontfinomítás tétele alapján belátjuk, hogy az új specifikáció finomítása az eredetinek.

Megmutatjuk, hogy: $(8.28) \wedge (8.29) \wedge (8.27) \Rightarrow (8.25)$.

Jelölje f^i az első $i+1$ függvénykomponens kompozícióját: $f^i ::= f_i \circ \dots \circ f_0$. Teljes indukcióval belátható, hogy $(8.28) \wedge (8.29) \wedge (8.27) \Rightarrow (x_{i+1}.h) = f_i(D)$. Nyilvánvaló, hogy ha ezt beláttuk, akkor abból n -re is következik az állítás.

Tétel: (Adatcsatorna tétel)

Az alábbi program megfelel a finomított specifikációnak

$$S: \left(\bigparallel_{i=1}^n x_i := \Diamond, \right. \\ \left. \bigwedge_{i=0}^N x_i, x_{i+1} := \text{lorem}(x_i), \text{hiext}(x_{i+1}, f_i(x_i.\text{lov})), \text{ ha } x \neq \Diamond \right\}$$

Elágazás (Fork)

Az elágazás egy olyan folyamat, melynek egy bemenő és két kimenő csatornája van.



(H. Z. tanár úr ábrája)

A fenti ábrán látható elágazástól a következő 3 feltétel teljesülését követeljük meg:

- adat ne vesszen el, azaz minden, ami rákerül a bemenetre, az kerüljön feldolgozásra,

- a kimenő csatornákon csak olyan adat jelenjen meg, amelyik a bemenő csatornán érkezett, azaz legyen a külső zajoktól mentes,
- ha érkezik adat, akkor az előbb-utóbb kerüljön feldolgozásra.

Vezessünk be egy függvényt, mely segítségével könnyebben tudjuk majd specifikálni az elágazást:

$$split : Ch \times Ch \times Ch \rightarrow L$$

A *split* függvényt induktívan definiáljuk:

$$split(\diamond, \diamond, \diamond) = igaz$$

$$split(a, b, c) \rightarrow split(a; x, b; x, c) \wedge split(a; x, b, c; x).$$

Legyen *split* a fenti két tulajdonsággal rendelkező függvények közül a legkisebb igazsághalmazú. (Ez egy fontos kikötés, hiszen pl. az azonosan igaz függvény is rendelkezik ezen tulajdonságokkal.)

$$A = Ch \times Ch \times Ch \times Ch \times Ch \times Ch$$

$x \quad y \quad z \quad x.h \quad y.h \quad z.h$

$$B = Ch \times Ch \times Ch \times Ch \times Ch \times Ch$$

$x' \quad y' \quad z' \quad x'.h \quad y'.h \quad z'.h$

- 1) $Q = (x = y = z = x.h = y.h = z.h = x' = y' = z' = x'.h = y'.h = z'.h)$
 $Q \in INIT_h$
- 2) $P = (split(x.h-x, y.h, z.h)) \in inv_h$
- 3) $\forall k \in N : |x.h| \geq k \hookrightarrow_s |y.h| + |z.h| \geq k$

Az invariáns megköveteli, hogy ne vesszenek el adatok és zajmentes legyen a működés. A haladási feltétel nem követel meg „pártatlanságot” abban a tekintetben, hogy mindkét kimenő csatornára kerüljön adat.

A megoldóprogram:

$$S : (SKIP, \{x, z := lorem(x), hiext(z, x.lov), \text{ ha } x \neq \diamond \square \\ x, y := lorem(x), hiext(y, x.lov), \text{ ha } x \neq \diamond \})$$

- 2.) $(Q \Rightarrow lf(s_0, P)) \Leftrightarrow (P \Rightarrow P)$. mert $split(\diamond - \diamond, \diamond, \diamond) = split(\diamond, \diamond, \diamond) = igaz$

Belátjuk, hogy $P \Rightarrow lf(S, P) = lf(s_1, P) \wedge lf(s_2, P)$

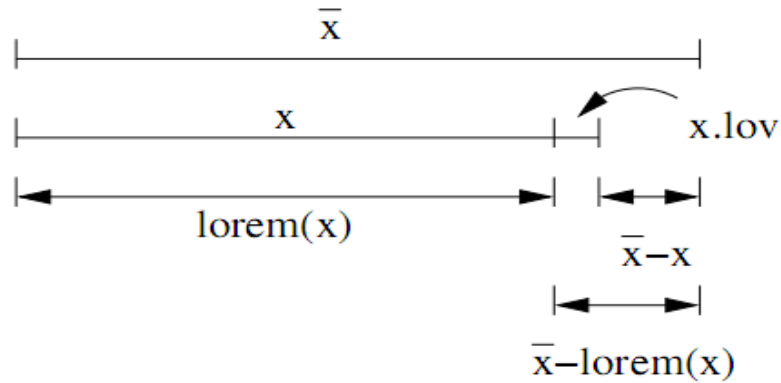
Csak $lf(s_1, P)$ -t bizonyítjuk, $lf(s_2, P)$ -t ugyanígy kell kiszámolni.

$$P \Rightarrow lf(s_1, P) = (x \neq \diamond \rightarrow P^{y \leftarrow hiext(y, x.lov), y.h \leftarrow y.h; x.lov, x \leftarrow lorem(x)}) \wedge (x = \diamond \rightarrow P) = \\ = (x \neq \diamond \rightarrow split(x.h-lorem(x), y.h; x.lov, z.h)) \wedge (x = \diamond \rightarrow P)$$

Be kell látni, hogy:

$$x.h - lorem(x) = x.h-x; x.lov$$

Az összefüggés teljesülését az alábbi ábrán jól lehet látni (Horváth Zoltán tanár úr ábrája):



A *split* definícióját alkalmazva azt kapjuk, hogy ha P igaz volt, akkor igaz is marad.

3.) *A haladási feltétel nincs bizonyítva a jegyzetben, szóval azt, gondolom, nem kéri.*

Forrás: A kidolgozás teljes egészében Horváth Zoltán tanár úr elektronikus jegyzete alapján készült: <http://people.inf.elte.hu/hz/parh/jegyzet.ps>

A természetes számok generátora is része a tételnek. Lényege, hogy egy csatornára folyamatosan, növekvő rendben el állítjuk a természetes számokat.
Ld. a jegyzet 108. oldalán.

Legyenek feltétel:

- Def: $\llbracket \ell(S, R) \rrbracket = \{a \in D_p(s) \mid p(s)(a) \subseteq R\}$

Tulajdonságok feltételre vonatkozóan:

- $\ell(s, \uparrow) = \uparrow$
- $\ell(s, \downarrow) = \downarrow$
- $P \Rightarrow R \rightarrow \ell(s, P) \Rightarrow \ell(s, R)$
- $\ell(s, P) \wedge \ell(s, R) \Rightarrow \ell(s, P \wedge R)$
- $\ell(s, P) \vee \ell(s, R) \Rightarrow \ell(s, P \vee R)$
- $\ell(s, R) = p(s) \circ R$

Tulajdonságok indukciós programra:

- $\ell(s, \uparrow) = \uparrow$
- $\ell(s, \downarrow) = \downarrow$
- $\ell(s, P) \wedge \ell(s, R) = \ell(s, P \wedge R)$
- $\ell(s, P) \vee \ell(s, R) \Rightarrow \ell(s, P \vee R)$
- $P \Rightarrow R \rightarrow \ell(s, P) \Rightarrow \ell(s, R)$
- $\ell(s, P) = \bigwedge_{i=1}^{\infty} \ell(s_i, P)$

D_S :

- Def: $P D_S Q \equiv P \wedge \neg Q \Rightarrow \ell(s, P \vee Q)$

Tulajdonságok:

- $\uparrow D_S P$
- $\downarrow D_S P$
- $P D_S \uparrow$

- Stabilitás: $P D_S \downarrow \equiv P \Rightarrow \ell(s, P) \equiv P \text{ stabil}$

- Jelölés gyengítése:
$$\frac{P D_S Q, Q \Rightarrow R}{P D_S R}$$

- Implikációs következmény:
$$\frac{P \Rightarrow Q}{P D_S Q}$$

- Diszjunktivitás:
$$\frac{P D_S R, Q D_S R}{(P \vee Q) D_S R}$$

- Monotonitás:
$$\frac{P D_S R, Q D_S R}{(P \wedge Q) D_S R}$$

$$\text{Stabilitás metrika: } \frac{P \vdash_S Q, K \text{ stabil}}{(P \wedge K) \vdash_S (Q \wedge K)}$$

-2-

\vdash_S :

- Def: $P \vdash_S Q \equiv P \vdash_S Q \wedge \exists \phi \in S: P \wedge \neg Q \Rightarrow \phi(s, Q)$

- Függvények:

$$\bullet \downarrow \vdash_S P$$

$$\bullet P \vdash_S \uparrow$$

- Implikáció hűtése:

$$\frac{P \Rightarrow Q}{P \vdash_S Q}$$

- Jobb oldal gyengítése:

$$\frac{P \vdash_S Q, Q \Rightarrow R}{P \vdash_S R}$$

- Hangsúlyosítás:

$$\frac{P \vdash_S Q, Q \vdash_S R}{(P \wedge Q) \vdash_S R}$$

- Stabilitás metrika:

$$\frac{P \vdash_S Q, K \text{ stabil}}{(P \wedge K) \vdash_S (Q \wedge K)}$$

\hookrightarrow_S :

- Def: $P \hookrightarrow_S Q \equiv P \vdash_S^{\text{tel}} Q$

$$\bullet P \vdash_S Q \rightarrow P \hookrightarrow_S Q$$

$$\bullet \frac{P \hookrightarrow_S Q, Q \hookrightarrow_S R}{P \hookrightarrow_S R}$$

$$\bullet \frac{\forall i \in \{1, \dots, n\}: P_i \hookrightarrow_S Q, \bigvee_{i=1}^n P_i = P}{P \hookrightarrow_S Q}$$

• Grah. ereh!

- Implikáció hűtése:

$$\frac{P \Rightarrow Q}{P \hookrightarrow_S Q}$$

- Jobb oldal gyengítése:

$$\frac{P \hookrightarrow_S Q, Q \Rightarrow R}{P \hookrightarrow_S R}$$

- Bal oldal erősítése:

$$\frac{P \hookrightarrow_S Q, R \Rightarrow P}{R \hookrightarrow_S Q}$$

- Függvények:

$$\bullet \downarrow \hookrightarrow_S P$$

$$\bullet P \hookrightarrow_S \uparrow$$

- Stabilitási metrikus:
$$\frac{P \hookrightarrow_S Q, K \text{ stabilis}}{(P \wedge K) \hookrightarrow_S (Q \wedge K)}$$

PSP-tétel:
$$\frac{P \hookrightarrow_S Q, R \supset_S B}{P \wedge R \hookrightarrow_S (Q \wedge R) \vee B}$$

Függvény:
$$f_S = \bigwedge_{i=1}^n (\pi_i \rightarrow (x = F_i(x)))$$

$$(\forall i \in \{1, \dots, n\}) : \pi_i = x := F_i(x) \wedge \pi_i$$

Függvény attribútumok:

- általános: $A \Rightarrow \pi \rightarrow p \equiv A \wedge \pi \Rightarrow p$

- D5-re: $P \wedge \neg Q \Rightarrow \pi \rightarrow (p \vee q)^S \wedge \neg \pi \rightarrow (p \vee q) \equiv P \wedge \neg Q \wedge \pi \Rightarrow (p \vee q)^S$

Terminális: $Q \in \text{TERM}_S \equiv Q \hookrightarrow_S f_S$

Variáns-függvény:

$t: A \rightarrow \mathbb{Z}$ variáns függvény

$P \wedge \neg Q \Rightarrow t > 0$

$P \wedge \neg Q \wedge (t = m) \hookrightarrow_S (P \wedge t < m) \vee Q$

$P \hookrightarrow_S Q$