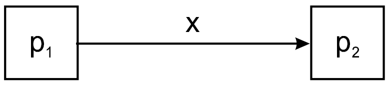
Adatcsatorna

Horváth Zoltán előadásai alapján

1 Csatornaváltozók

1.1 Csatornaváltozók

x csatornaváltozó



A változón értelmezett műveletek:

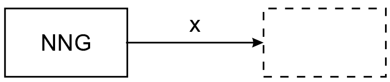
$$x : sor(a)$$
$$x :=<>$$
$$x := \text{lorem}(x)$$
$$x := \text{hiext}(x, e)$$
$$x.\text{dom} =: |x|$$
$$x.\text{lov}$$

Legyen x csatornaváltozó $\overline{x}$ az x csatorna története (csak a specifikációban és a verifikáció során használjuk).

$$lf(x :=<>, R) = R^{x \leftarrow <>, \overline{x} \leftarrow <>}$$
$$lf(x := \text{lorem}(x), R) = R^{x \leftarrow \text{lorem}(x)}$$
$$lf(x : \text{hiext}(x, e)) = R^{x \leftarrow x;e, \overline{x} \leftarrow \overline{x};e}$$

1.2 NNG (Natural Number Generator)

Az NNG természetes számokat generál, és az x nevű csatornára kell kitennie őket. Kezdetben $\overline{x}$ üres.



$$Ch := seq(\mathbb{N})$$
$$A = \underset{x}{Ch} \times \underset{\overline{x}}{Ch}$$
$$B = \underset{x'}{Ch} \times \underset{\overline{x}'}{Ch}$$

(1.) $Q : (x = x' = \overline{x} = \overline{x}' =<>) \in \mathrm{INIT}_{x',\overline{x}'}$

(2.) $P : (\exists i : i > 0 : \overline{x} \leq< 1, \ldots, i >) \in \mathrm{INV}_{x',\overline{x}'}$

(3.) $\forall k \in \mathbb{N}_0 : |\overline{x}| = k \hookrightarrow_{x',\overline{x}'} |\overline{x}| = k + 1$

$$Ch := seq(\mathbb{N})$$
$$A = \underset{x}{Ch} \times \underset{\overline{x}}{Ch} \times \underset{i}{\mathbb{N}_0}$$
$$B = \underset{x'}{Ch} \times \underset{\overline{x}'}{Ch}$$

(1.) és (3.) ugyanaz

(2.) $P : ((i > 0 \to \overline{x} =< 1, \ldots, i >) \wedge (i = 0 \to \overline{x} =<>)) \in \mathrm{INV}_{x'\overline{x}'}$

Legyen a megoldóprogram:

$$\mathbb{S} := (i := 0, i := i + 1 \parallel x : \mathrm{hiext}(i + 1))$$

Az invariáns tulajdonság (2.) bizonyítása:

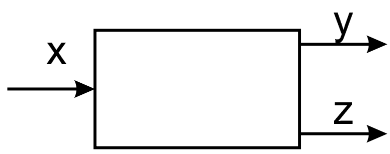
$$sp(s_0, Q) \Rightarrow P :$$
$$(i > 0 \to \overline{x} =< 1, \ldots, i >) \wedge (i = 0 \to \overline{x} =<>)$$
$$P \Rightarrow lf(S, P) :$$
$$lf(S, P) = (i + 1 > 0 \to \overline{x} = \overline{x} : \mathrm{hiext}(i + 1) =< 1, \ldots, i + 1 >) \wedge$$
$$(i + 1 = 0 \to \overline{x} =<>)$$

A haladási tulajdonság (3.) bizonyítása:

$$\forall k : |\overline{x}| = k \hookrightarrow_S |\overline{x}| = k + 1$$
$$\forall k : |\overline{x}| = k \mapsto_S |\overline{x}| = k + 1$$
$$\rhd_S : |\overline{x}| = k \Rightarrow lf(s_1, |\overline{x}| = k \vee |\overline{x}| = k + 1)$$
$$\exists s \in S : |\overline{x}| = k \Rightarrow lf(s, |\overline{x}| = k + 1)$$

1.3 Elágazás (fork)

Az x csatornáról a Fork az adatokat az y és a z csatornára adja ki.

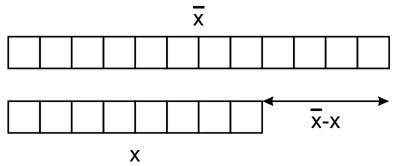


A specifikációban használni fogjuk a *split* függvényt. A split függvény definíciója:

$$
\begin{array}{ll}
 & split : Ch \times Ch \times Ch \to \mathbb{L} \\
(1.) & split(<>,<>,<>) = \uparrow \\
(2.) & split(a,b,c) \to split(a;x,b;x,c) \wedge split(a;x,b,c;x) \\
(3.) & \text{és a legszűkebb ilyen függvény}
\end{array}
$$

A feladat specifikációja:

$$
\begin{array}{ll}
 & A = \underset{x}{Ch} \times \underset{\overline{x}}{Ch} \times \underset{y}{Ch} \times \underset{\overline{y}}{Ch} \times \underset{z}{Ch} \times \underset{\overline{z}}{Ch} \\
 & B = \underset{x'}{Ch} \times \underset{\overline{x}'}{Ch} \times \underset{y'}{Ch} \times \underset{\overline{y}'}{Ch} \times \underset{z'}{Ch} \times \underset{\overline{z}'}{Ch} \\
(1.) & Q : (x = x' = \overline{x} = \overline{x}' = y = y' = \overline{y} = \overline{y}' = z = z' = \overline{z} = \overline{z}' = <>) \\
 & Q \in \mathrm{INIT}_h \\
(2.) & P : split(\overline{x} - x, \overline{y}, \overline{z}) \in \mathrm{INV}_h \qquad \overline{x} - x : \text{ ami átment a dobozon} \\
(3.) & \forall k \in \mathbb{N} : |\overline{x}| = k \hookrightarrow_h |\overline{y}| + |\overline{z}| \geq k
\end{array}
$$



A megoldó program:

$$
\mathbb{S} := \left(SKIP, \left\{ \begin{array}{l} y, x := \mathrm{hiext}(y, x.\mathrm{lov}), \mathrm{lorem}(x), \text{ ha } x \neq <> \\ \Box \\ z, x := \mathrm{hiext}(z, x.\mathrm{lov}), \mathrm{lorem}(x), \text{ ha } x \neq <> \end{array} \right\} \right)
$$

Az invariáns tulajdonság (2.) bizonyítása:

$$
\begin{array}{l}
sp(s_0, Q) \Rightarrow split(\overline{x} - x, \overline{y}, \overline{z}) : \\
s_0 = SKIP \\
Q : (x = x' = \overline{x} = \overline{x}' = y = y' = \overline{y} = \overline{y}' = z = z' = \overline{z} = \overline{z}' = <>) \Rightarrow \\
(\overline{x} - x = <> \wedge \overline{y} = <> \wedge \overline{z} = <>) \Rightarrow split(\overline{x} - x, \overline{y}, \overline{z})
\end{array}
$$

$P \Rightarrow lf(S,P) = lf(s_1,P) \wedge lf(s_2,P)$:
$lf(s_1,P) = x \neq <> \rightarrow split(\overline{x} - \text{lorem}(x), (\overline{y}; x.\text{lov}), \overline{z}) =$
$x \neq <> \rightarrow split((\overline{x} - x; x.\text{lov}), (\overline{y}; x.\text{lov}), \overline{z})$
$lf(s_2,P) = x \neq <> \rightarrow split(\overline{x} - \text{lorem}(x), \overline{y}, (\overline{z}; x.\text{lov})) =$
$x \neq <> \rightarrow split((\overline{x} - x; x.\text{lov}), \overline{y}, (\overline{z}; x.\text{lov}))$

A haladási tulajdonság (3.) bizonyítása: $|\overline{x}| = k \hookrightarrow_S |\overline{y}| + |\overline{z}| \geq k$. Tegyük fel, hogy $|\overline{x}| = k$ (k rögzített).

1. eset: $|\overline{y}| + |\overline{z}| \geq k$ ekkor készen vagyunk.
2. eset: $|\overline{y}| + |\overline{z}| < k$ és $x \neq <>$.

   Tegyük fel, hogy $|\overline{y}| + |\overline{z}| = l, l < k$.

   Állítás 1. $|\overline{y}| + |\overline{z}| = l \wedge x \neq <> \mapsto_S |\overline{y}| + |\overline{z}| = l + 1$

   Megjegyzés: Az adott feltételek mellet az $\mapsto_S$ tulajdonságból tranzitivitással már következik: $|\overline{y}| + |\overline{z}| < k \wedge x \neq <> \hookrightarrow_S |\overline{y}| + |\overline{z}| \geq k$.

   Bizonyítás. Biztosítja tulajdonság:

$$|\overline{y}| + |\overline{z}| = l \wedge x \neq <> \Rightarrow$$
$$lf(S, (|\overline{y}| + |\overline{z}| = l \wedge x \neq <>) \vee (|\overline{y}| + |\overline{z}| = l + 1))$$

   Ugyanis:

$$lf(s_1, (|\overline{y}| + |\overline{z}| = l \wedge x \neq <>) \vee (|\overline{y}| + |\overline{z}| = l + 1)) =$$
$$((|\overline{y} + |\overline{z}| = l \wedge x \neq <>) \vee$$
$$(|\overline{y}| + |\overline{z}| = l + 1))^{x \leftarrow \text{lorem}(x), \overline{y} \leftarrow \overline{y}; x.\text{lov}, \overline{z} \leftarrow \overline{z}} =$$
$$= (|\overline{y}; x.\text{lov}| + |\overline{z}| = l \wedge \text{lorem}(x) \neq <>) \vee$$
$$(|\overline{y}; x.\text{lov}| + |\overline{z}| = l + 1)$$

   Tehát $P \Rightarrow lf(s_1, (|\overline{y}| + |\overline{z}| = l \wedge x \neq <>) \vee (|\overline{y}| + |\overline{z}| = l + 1))$.
   Hasonlóan igazolható, hogy: $P \Rightarrow lf(s_2, (|\overline{y}| + |\overline{z}| = l \wedge x \neq <>) \vee (|\overline{y}| + |\overline{z}| = l + 1))$.
   Így teljesül: $P \Rightarrow lf(S, (|\overline{y}| + |\overline{z}| = l \wedge x \neq <>) \vee (|\overline{y}| + |\overline{z}| = l + 1))$.
   Haladási tulajdonság: Az olvasóra bízzuk annak belátását, hogy

$$\exists s_i \in S : (|\overline{y}| + |\overline{z}| = l \wedge x \neq <>) \Rightarrow lf(s_i, |\overline{y}| + |\overline{z}| = l + 1).$$

   Például s_1 megfelel, de ugyanígy jó s_2 is. □

Az állítás következményeként: $|\overline{y}|+|\overline{z}| = l \wedge x \neq <> \hookrightarrow_S |\overline{y}|+|\overline{z}| = l+1$.

(a) Ha $l+1 \geq k$, akkor készen vagyunk.

(b) Ha $l+1 < k$, akkor

$$|\overline{y}| + |\overline{z}| = l+1 \wedge x \neq <> \hookrightarrow_S |\overline{y}| + |\overline{z}| = l+2.$$

i. Ha $l+2 \geq k$, akkor készen vagyunk.

ii. Ha $l+2 < k$, akkor tovább ...

1.4 Multiplexer

Az x és az y csatornákról bejövő adatokat a z-re kell továbbítani anélkül, hogy adat veszne el, vagy új adatokat generálnánk.

A feladat specifikációja:

$$A = \underset{x}{Ch} \times \underset{\overline{x}}{Ch} \times \underset{y}{Ch} \times \underset{\overline{y}}{Ch} \times \underset{z}{Ch} \times \underset{\overline{z}}{Ch}$$

$$B = \underset{x'}{Ch} \times \underset{\overline{x}'}{Ch} \times \underset{y'}{Ch} \times \underset{\overline{y}'}{Ch} \times \underset{z'}{Ch} \times \underset{\overline{z}'}{Ch}$$

(1.) $Q : (x = x' = \overline{x} = \overline{x}' = y = y' = \overline{y} = \overline{y}' = z = z' = \overline{z} = \overline{z}' = <>)$

$Q \in \mathrm{INIT}_h$

(2.) $P : split(\overline{z}, \overline{z}_x, \overline{z}_y) \in \mathrm{INV}_h$, ahol $\overline{z}_x := \overline{x} - x, \overline{z}_y := \overline{y} - y$

(3.) $\forall k, l \in \mathbb{N} : |\overline{x}| \geq k \hookrightarrow_h (|\overline{z}_x| \geq k) = (|\overline{x} - x| \geq k)$

$|\overline{y}| \geq l \hookrightarrow_h (|\overline{z}_y| \geq l) = (|\overline{y} - y| \geq l)$

A megoldó program:

$$\mathbb{S} := \left(SKIP, \left\{ \begin{array}{l} x, z := \mathrm{lorem}(x), \mathrm{hiext}(z, x.\mathrm{lov}), \text{ ha } x \neq <> \\ \square \\ y, z := \mathrm{lorem}(y), \mathrm{hiext}(z, y.\mathrm{lov}), \text{ ha } y \neq <> \end{array} \right\} \right)$$

Az invariáns tulajdonság bizonyítása:

$$
\begin{aligned}
&sp(s_0, Q) \Rightarrow P: \\
&s_0 = SKIP \\
&sp(s_0, Q) = \\
&(x = x' = \overline{x} = \overline{x}' = y = y' = \overline{y} = \overline{y}' = z = z' = \overline{z} = \overline{z}' =<>) \Rightarrow \\
&(\overline{z} =<> \wedge x =<> \wedge \overline{x} =<> \wedge y =<> \wedge \overline{y} =<>)) \Rightarrow \\
&split(\overline{z}, \overline{x} - x, \overline{y} - y)
\end{aligned}
$$

$$
\begin{aligned}
&P \Rightarrow lf(S, P): \\
&lf(S, P) = \\
&((x \neq <>) \rightarrow split(z; x.\text{lov}, (\overline{x} - x); x.\text{lov}, \overline{y} - y) \wedge (x =<>) \rightarrow P) \wedge \\
&((y \neq <>) \rightarrow split(z; y.\text{lov}, \overline{x} - x, (\overline{y} - y); y.\text{lov}) \wedge (y =<>) \rightarrow P))
\end{aligned}
$$

A haladási tulajdonság bizonyítása: $\forall k \in \mathbb{N} : |\overline{x}| = k \hookrightarrow_S |\overline{x} - x| \geq k$.

1. Ha $|\overline{x} - x| \geq k$, akkor készen vagyunk.
2. Ha $|\overline{x} - x| < k$, akkor tegyük fel, hogy $|\overline{x} - x| = l, l < k$. Azt kellene belátnunk, hogy $|\overline{x} - x| = l \hookrightarrow_S |\overline{x} - x| = l + 1$, ha $x \neq <>$. Ez teljesül, ha $|\overline{x} - x| = l \mapsto_S |\overline{x} - x| = l + 1$ igaz.

$$
\begin{aligned}
&|\overline{x} - x| = l \Rightarrow lf(S, |\overline{x} - x| = l \vee |\overline{x} - x| = l + 1) \\
&|\overline{x} - x| = l \Rightarrow |\overline{x} - \text{lorem}(x)| = l \vee |\overline{x} - \text{lorem}(x)| = l + 1 \\
&|\overline{x} - x| = l \Rightarrow lf(s, |\overline{x} - x| = l + 1) = (|\overline{x} - \text{lorem}(x)| = l + 1)
\end{aligned}
$$

Azt kell még belátnunk, hogy $\forall l \in \mathbb{N} : |\overline{y}| = l \hookrightarrow_S |\overline{y} - y| = l + 1$. Ez teljesül, ha $\forall l \in \mathbb{N} : |\overline{y}| = l \mapsto_S |\overline{y} - y| = l + 1$ igaz. Tegyük fel, hogy $|\overline{y}| = m$.

1. Ha $|\overline{y} - y| \geq l$, akkor készen vagyunk.
2. Ha $|\overline{y} - y| < l$, akkor hasonlóan járhatunk el, mint x esetében.

2 Adatcsatorna tétel

2.1 A feladat

Egy $n+1$ komponensből álló összetett függvény értékét szeretnénk kiszámolni m helyen.

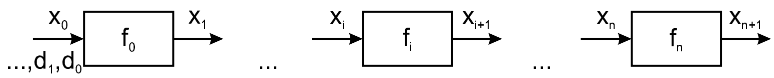
$$D =< d_0, d_1, \ldots, d_m >$$
$$F(D) =< F(d_0), F(d_1), \ldots, F(d_m) >$$
$$F = f_n \circ f_{n-1} \circ \cdots \circ f_1 \circ f_0$$

Feltesszük, hogy m lényegesen nagyobb mint n, és a rendelkezésünkre áll $\Theta(n)$ processzor.

Például a szinusz vagy a koszinusz függvény ilyen módon számolható ki, ha a hatványsoruk n-edik részletösszegével közelítjük őket.

2.2 A megoldási elv

Minden egyes processzor a függvénykompozíció egy-egy összetevőjét fogja kiszámolni.



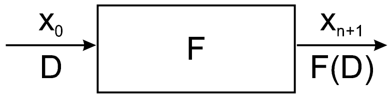
Az elért gyorsulás (speedup) Az egy processzor által elvégzett számítást tekintjük egy időegységnek ($f_i(d)$). Így a futási idő:

- 1 processzor esetén: $\Theta(m * n)$,
- n processzor esetén: $\Theta(n + m)$.

Tehát a gyorsulás:

$$\frac{c_1(m*n)}{c_2(n+m)} = \frac{c_1}{c_2} * \frac{m*n}{n+m} \approx \frac{m*n}{m} * c = c * n$$

2.3 Formális specifikáció



$$A = \underset{x_0}{Ch(a)} \times \underset{\overline{x}_0}{Ch(a)} \times \underset{x_{n+1}}{Ch(b)} \times \underset{\overline{x}_{n+1}}{Ch(b)}$$

$$B = \underset{x'_0}{Ch(a)} \times \underset{\overline{x}'_0}{Ch(a)} \times \underset{x'_{n+1}}{Ch(b)} \times \underset{\overline{x}'_{n+1}}{Ch(b)}$$

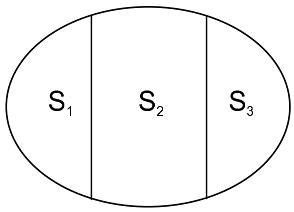
(1.) $Q : (x_0 = \overline{x}_0 = D = x'_0 = \overline{x}'_0 \wedge x_{n+1} = \overline{x}_{n+1} = x'_{n+1} = \overline{x}'_{n+1} = <>)$

$Q \in \mathrm{INIT}_{\underbrace{x'_0 \overline{x}'_0 x'_{n+1} \overline{x}'_{n+1}}_{D}}$

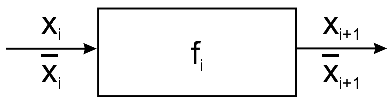
(2.) $\mathrm{FP}_D \Rightarrow \overline{x}_{n+1} = D(\overline{x}'_0) = F(D)$

(3.) $Q \hookrightarrow_D \mathrm{FP}_D$

(4.) $x'_0 = \overline{x}'_0 = D \in \mathrm{inv}_D$



2.4 Finomítás



(5.) $\mathrm{inv}_D : \forall i \in [0..n] : f_i(\overline{x}_i - x_i) = \overline{x}_{i+1}$

(6.) $\mathrm{FP}_D \Rightarrow (\forall i \in [0..n] : x_i = <>)$

Most bebizonyítjuk, hogy a (4.), (5.) és (6.) feltételekből már következik a (2.). Azt kell megmutatnunk, hogy $\mathrm{FP}_D \Rightarrow \overline{x}_{n+1} = F(\overline{x}'_0)$. Ezt teljes indukcióval fogjuk megtenni.

Állítás 2. *$\forall i \in [0..n] : f^i(\overline{x}_0) = \overline{x}_{i+1}$, ahol $f^i \overset{def}{=} f_i \circ \cdots \circ f_0$.*

Az állítás következtében $i = n$ esetén: $f^n(\overline{x}_0) \overset{(4.)}{=} F(D) = \overline{x}_{n+1}$.

Bizonyítás. 1. $i = 0$-ra $f^0(\overline{x}_0) \overset{?}{=} \overline{x}_1$. Írjuk fel az (5.) és a (6.) pont állítását $i = 0$-ra:

$$\left.\begin{array}{l} (5.) : f_0(\overline{x}_0 - x_0) = \overline{x}_1 \\ (6.) : x_0 = <> \end{array}\right\} f_0(\overline{x}_0) = \overline{x}_1$$

2. Indukciós lépés: tegyük fel, hogy $f^{i-1}(\overline{x}_0) = \overline{x}_i$. Ezután használjuk megint az (5.) és (6.) pontokat:

$$\left.\begin{array}{l}(5.) : f_i(\overline{x}_i - x_i) = \overline{x}_{i+1} \\ (6.) : x_i =<>\end{array}\right\} f_i(\overline{x}_i) = \overline{x}_{i+1}$$

Most használjuk az indukciós feltételt:

$$f_i(f^{i-1}(\overline{x}_0)) = \overline{x}_{i+1}.$$

A függvény definíciója alapján ez azt jelenti, hogy:

$$f^i(\overline{x}_0) = \overline{x}_{i+1}.$$

$\square$

2.5 Variánsfüggvény

Használjuk a $v : A \mapsto \mathbb{Z}$ variánsfüggvénynek az $|x_0|, |x_1|, \ldots, |x_n|$ értékek helyiértékesen súlyozott kombinációját, legyen

$$v(a) := \sum_{i=0}^{n} |x_i| * m^{n-i}.$$

2.6 A megoldó program

$$\begin{array}{ll}\mathbb{S} = & \Big(\mathop{\|}\limits_{i=1}^{n} x_i :=<>, \\ & \quad \Big\{ \mathop{\square}\limits_{i=0}^{n} x_i, x_{i+1} := \\ & \quad\; \text{lorem}(x_i), (x_{i+1}; f_i(x_i.\text{lov})), \text{ ha } x_i \neq<> \\ & \quad \Big\}\Big)\end{array}$$

2.7 A megoldás helyességének vizsgálata

Először számoljuk ki a program fixpontját (ϕ_S):

$$\phi_S = (\forall i \in [0..n] : x_i \neq<> \rightarrow x_i = \text{lorem}(x_i) \wedge$$
$$x_{i+1} = x_{i+1}; f_i(x_i.\text{lov})) = (\forall i \in [0..n] : x_i =<>)$$

1. A (3.) pont belátásához az variánsfüggvény tételét használjuk fel. Könnyen belátható, hogy

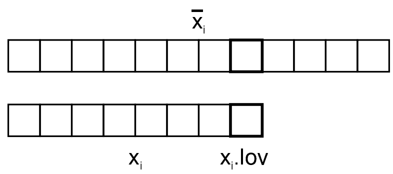
$$\forall i : \neg\phi_S \wedge x_i \neq <> \wedge t = t' \mapsto_s t \leq t' - 1 \vee \phi_S.$$

2. Az (5.) állítás invariáns voltát igazoljuk. Ez az állítás tulajdonképpen $n+1$ állítás konjunkciója: $P = \bigwedge_{i=0}^{n} f_i(\overline{x}_i - x_i) = \overline{x}_{i+1}$. Bontsuk fel a feltételt $n+1$ darab tényezőre: $P_0 = f_0(\overline{x}_0 - x_0) = \overline{x}_1, \ldots, P_{n-1} = f_n(\overline{x}_n - x_n) = \overline{x}_{n+1}$.

   Azt kell megmutatni, hogy $sp(s_0, Q) \Rightarrow P$ és $P \Rightarrow lf(S, P)$. Az utóbbi állítás igazolásához elég, ha azt látjuk be, hogy $\forall i \in J : P \Rightarrow lf(s_i, P)$.

   Tehát az a kérdés, hogy $P_1 \wedge \cdots \wedge P_j \wedge \cdots \wedge P_n \Rightarrow lf(s_i, P_1 \wedge \cdots \wedge P_n)$ igaz-e. Könnyen észrevehető, hogy s_i csak a P három tagját érinti, azokat, amelyekben x_i illetve x_{i+1} szerepel.

$$\begin{aligned}
&lf(s_i, P) = (x_i \neq <> \rightarrow \cdots \wedge \\
&\wedge f_{i-1}(\overline{x}_{i-1} - x_{i-1}) = \overline{x}_i \wedge \\
&\wedge f_i(\overline{x}_i - \mathrm{lorem}(x_i)) = \overline{x}_{i+1}; f_i(x_i.\mathrm{lov}) \wedge \\
&\wedge f_{i+1}(\overline{x}_{i+1}; f_i(x_i.\mathrm{lov}) - x_{i+1}; f_i(x_i.\mathrm{lov})) = \overline{x}_{i+2} \wedge \\
&\wedge \ldots) \\
&\wedge (x_i = <> \rightarrow P)
\end{aligned}$$



   Az átalakítások során felhasználtuk azt, hogy $\overline{x}_i - \mathrm{lorem}(x_i) = (\overline{x}_i - x_i); x_i.\mathrm{lov}$, így $f_i(\overline{x}_i - \mathrm{lorem}(x_i)) = f_i(\overline{x}_i - x_i); f_i(x_i.\mathrm{lov})$. Tehát $P \Rightarrow lf(s_i, P)$.

3 Nyitott specifikáció

Kiegészítés az unió viselkedési relációjához.

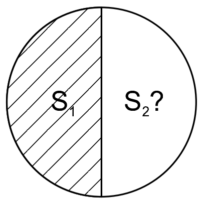
Tétel 1. *Tegyük fel, hogy*

(*) $x \geq k \rhd_{S_1 \cup S_2} \downarrow$
$S_1 := (SKIP, x := x + 1)$
$P : x \leq 5$
$Q : x > 5.$

Ekkor

$$P \hookrightarrow_{S_1 \cup S_2} Q.$$

Megjegyzés: (*) azt jelenti, hogy x értéke nem csökken az unióban. S_1 ismert, S_2 ismeretlen, S_2-re nézve implicit kikötést jelent (*).



Ezért nyitott specifikáció a módszer neve (tehát nem a feladatra vonatkozik a specifikáció megnevezés).

Bizonyítás. Könnyen ellenőrizhető, hogy

$$x \geq k \mapsto_{S_1} x > k.$$

Az unió viselkedési relációjáról szóló tétel alapján:

$$x \geq k \rhd_{S_2} \downarrow .$$

A gyengítési tétel miatt

$$x \geq k \rhd_{S_2} x > k.$$

Tehát $x \geq k \mapsto_{S_1 \cup S_2} x > k$, így $x \geq k \hookrightarrow_{S_1 \cup S_2} x > k$.

Vagyis $x \geq 3 \hookrightarrow_S x > 3$ teljesül, azaz $x \geq 3 \hookrightarrow_S x \geq 4$, ugyanígy $x \geq 4 \hookrightarrow_S x > 4$ teljesül, azaz $x \geq 4 \hookrightarrow_S x \geq 5$ is igaz. $\square$

4 Aszinkronitás tétel

4.1 Szinkron kommunikáció

Az információ úgy cserélhető ki, hogy a küldőnek és a fogadónak találkoznia kell, pl. Ada taszkok.

Most adjunk át adatokat x-ből az $y(i)$-kbe $(i = 1, \ldots, n)$. Az absztrakt program kommunikációs része így néz ki:

$$[komm] \begin{cases} \overset{n}{\underset{i=1}{\|}} y(i) := x, & \text{ha } \bigwedge_{i=1}^{n} y(i) = \underline{\emptyset} \wedge x \neq \underline{\emptyset} \\ \overset{n}{\underset{i=1}{\|}} x := \underline{\emptyset}, & \text{ha } \bigwedge_{i=1}^{n} y(i) = \underline{\emptyset} \wedge x \neq \underline{\emptyset} \end{cases}$$

Teljesülnek az alábbi tulajdonságok:

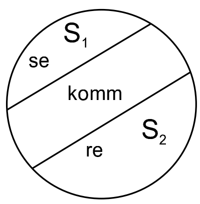
$$\begin{array}{ll} [se]: & x \neq \underline{\emptyset} \text{ stabil (küldő oldal)} \\ [re(i)]: & y(i) = \underline{\emptyset} \text{ stabil (fogadó oldal)} \end{array}$$

Ennek egy speciális esete, ha csak két folyamat van:

$$se \rightarrow re$$

Az absztrakt program kommunikációs része:

$$x, y := \underline{\emptyset}, x, \text{ ha } x \neq \underline{\emptyset} \wedge y = \underline{\emptyset}$$



4.2 Aszinkron kommunikáció

Aszinkron kommunikációra példa az e-mailen keresztül történő kommunikáció. A folyamatok csatornaváltozón vagy ,,pufferváltozón" keresztül kommunikálnak, így valójában szinkron kommunikációt használnak fel.

Például legyen u a pufferváltozó:

$$\begin{array}{ll} [se]: & x, u := \underline{\emptyset}, x, \text{ ha } x \neq \underline{\emptyset} \wedge u = \underline{\emptyset} \\ [re]: & u, y := \underline{\emptyset}, u, \text{ ha } u \neq \underline{\emptyset} \wedge y = \underline{\emptyset} \end{array}$$

Általában u sorozat típusú változó a memóriában.

4.3 Aszinkronitás tétel

A következőkben egy programtranszformációt mutatunk be, ami tulajdonképpen a programinverzió inverze.

Tétel 2. *Feltételek: Legyen az F absztrakt program a következő:*

$$\begin{array}{l} \{ \\ \quad \begin{array}{ll} \ldots & \\ [se]: & x := x \oplus d, \textit{ ha } bu \parallel \ldots \\ [re]: & vs, x := f(vs, x), g(x), \textit{ ha } b(x) \wedge bv \parallel \ldots \\ \ldots & \end{array} \\ \}, \end{array}$$

ahol $g(x)$ végzi az információ elvonását, $b(x)$ jelenti azt, hogy van új információ, x például egy csatornaváltozó és a $\oplus$ művelet mondjuk a hiext. A programban a bu és a bv változók értéke nem függ x-től.

1. $b(x) \Rightarrow b(x \oplus d)$,
2. $\forall d : b(x) \Rightarrow f(vs, x) = f(vs, x \oplus d)$ *és*
3. $b(x) \Rightarrow (g(x \oplus d) = g(x) \oplus d)$.

Állítás: A fenti feltételek teljesülése esetén az F programot át lehet alakítani a következő G programra:

$$\begin{array}{l} \{ \\ \quad \begin{array}{ll} \ldots & \\ [se]: & x, c := x \oplus d, \mathit{hiext}(c, d), \textit{ ha } bu \parallel \ldots \\ [re]: & vs, x, y := f(vs, y), g(x), g(y), \textit{ ha } b(y) \wedge bv \parallel \ldots \\ [ch]: & c, y := \mathit{lorem}(c), y \oplus c.\mathit{lov}, \textit{ ha } c.\mathit{dom} \neq 0 \\ \ldots & \end{array} \\ \} \end{array}$$

És ez a G program lényegében ekvivalens *F-fel.*

Megjegyzések:

1. $x \leftrightarrow y \oplus c$ invariáns.
2. A „lényegében ekvivalens” kifejezés bővebben azt jelenti, hogy
   (a) ha $P \rhd_F Q$, akkor $P \rhd_G Q$,
   (b) ha $P \hookrightarrow_F Q$, akkor $P \hookrightarrow_G Q$,

(c) ha $P\mathrm{inv}_F(Q)$, akkor $P\mathrm{inv}_G(Q)$.

3. Vegyük észre, hogy az $\mapsto$ tulajdonságok hiányoznak. Esetleg G ritkábban használ fel információt, azaz késleltetés lép fel.

Bizonyítás. (nélkül) □

[1]

[1]Ezt a vázlatot a 2001-2002. tanév 1. félévében Horváth Zoltán által tartott előadások alapján készítettem. Ha valamilyen hibát találnál benne, akkor írj egy e-mailt légy szíves: stuntman@inf.elte.hu, Durányik Ádám.

5 Tételjegyzék

1. Feladat fogalma
2. Párhuzamos absztrakt program
3. Az absztrakt program viselkedési relációja
4. A megoldás fogalma
5. Levezetési szabályok (finomítási tételek)
6. Asszociatív művelet eredményének kiszámítása
7. Csatornaváltozók, NNG
8. Elágazás
9. Multiplexer
10. Adatcsatorna tétel
11. Unió
12. Unió és az állapottér részhalmazai
13. Szuperpozíció
14. Szekvencia
15. Nyitott specifikáció
16. Kommunikáció, aszinkronitás tétel