SZEKVENCIÁLIS PROGRAMOK SZINTÉZISE

Levezetés

- Hoare-Dijkstra-Gries módszere (levezetési szabályok)
- Felülről lefele történő finomítási eljárás: minden lépésben egy rész-specifikácót vizsgálunk, amelyhez vagy tudunk közvetlenül megoldó (teljesen helyes) programot rendelni, vagy felbontjuk részekre valamelyik programszerkezet mentén.

Stratégia

- Visszalépéses (visszaugrásos) keresés
- Döntéseket hozunk, amelyek következtében fokozatosan ölt testet a program. Egy döntés mindig egy programrésszel kapcsolatos kérdésre kapott válasz:
  - Összetett program szerkezet vagy elemi?
  - Ha elemi, akkor az üres-e vagy melyik értékadás?
  - Ha összetett, akkor szekvencia, elágazás vagy ciklus?
  - Ha szekvencia, akkor mi a köztes állítás?
  - Ha elágazás, akkor mik a feltételek?
  - Ha ciklus, akkor mi az invariánsa, mi a termináló függvénye?
    - célba ér-e? – hogyan csökkenthető a termináló fügvénye? – invariáns-e az invariáns? – kezdetben jó-e az invariáns?
- Minden döntést a levezetési szabályokkal ellenőrzünk.
- Ha valamikor egy szabály nem teljesül, akkor visszalépünk (visszaugrunk) a nem teljesülő szabályt érintő döntési pontra és másik döntést hozunk.

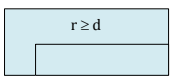
1. Példa

- Számoljuk ki csak összeadással és kivonással egy természetes szám egy másik természetes számmal vett osztási maradékát!
- $A = \mathbb{N} \times \mathbb{N} \times \mathbb{N}$
  $n \quad d \quad r$
  $Q = (n = n' \wedge d = d' \wedge d > 0)$
  $R = (Q \wedge d \mid (n-r) \wedge r < d)$

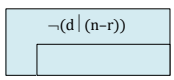
1. Példa folytatása

- Legyen ciklus! (ciklus lev. szab.)
- Válasszunk invariánst! Az $I \wedge \neg\alpha \Rightarrow R$ érdekében $I \wedge \neg\alpha = R$.
- $R = (Q \wedge d \mid (n-r) \wedge r < d)$

1. $I = (Q \wedge d \mid (n-r))$

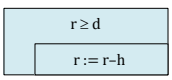


1. $I = (Q \wedge r < d)$



1. Példa folytatása

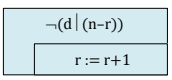
- $I = (Q \wedge d \mid (n-r))$



- r-t kéne d alá csökkenteni $t = r$

2. $I \wedge \alpha \Rightarrow t > 0$ azaz
   $I \wedge r \geq d \Rightarrow r > 0$
   ami igaz, mert Q miatt $d > 0$
3. Ha $r := r-h$, ahol $0 < h \leq r$, akkor $I \wedge \alpha \wedge t = t_0 \Rightarrow lf(S_0, t < t_0)$ hisz $lf(S_0, r < t_0) = r-h < t_0$ ami $r = t_0$ miatt igaz

- $I = (Q \wedge r < d)$

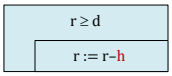


- $r \in [0..d-1]$-t végigpróbáljuk $t = d-r$

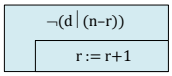
2. $I \wedge \alpha \Rightarrow t > 0$ azaz
   $I \wedge \alpha \Rightarrow d-r > 0$
   ami igaz, mert I miatt $r < d$
3. Ha $r := r+1$, akkor $I \wedge \alpha \wedge t = t_0 \Rightarrow lf(S_0, t < t_0)$ hisz $lf(S_0, d-r < t_0) = d-r-1 < t_0$
   ami $d-r = t_0$ miatt igaz

1. Példa folytatása

- $I = (Q \wedge d \mid (n-r))$



- $I = (Q \wedge r<d)$



4. $I \wedge \alpha \Rightarrow lf(S_0, I)$
   $lf(S_0, I) = I^{r \leftarrow r-h} =$
   $=(Q \wedge d \mid (n-r+h))$
   - $I \Rightarrow Q$
   - $I \Rightarrow d \mid (n-r)$ és
     ha $h=i*d$ ahol $i \in \mathbb{N}$ $(0<h \leq r)$,
     akkor $d \mid (n-r+i*d))$,
     például lehet $h=d$ $(i=1)$

4. $I \wedge \alpha \Rightarrow lf(S_0, I)$
   $? \; Q \wedge r<d \wedge \neg(d \mid (n-r)) \Rightarrow$
   $(Q \wedge r+1<d) \; ?$

1. Példa befejezése

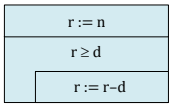
$I = (Q \wedge d \mid (n-r))$

5. $Q \not\Rightarrow I$, tehát $Q \not\Rightarrow lf(\textbf{while}\ \alpha\ \textbf{do}\ S_0\ \textbf{od}), R)$

Legyen S inkább szekvencia, amelynek második tagja az eddig levezetett ciklus, közbeeső állítása: I

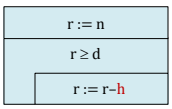
1. Egyrészt $I \Rightarrow I$, tehát $I \Rightarrow lf(\textbf{while}\ \alpha\ \textbf{do}\ S_0\ \textbf{od}), R)$
2. Másrészt $Q \Rightarrow I^{r \leftarrow n} = lf(r:=n, I)$

Így teljesül (szekv. lev. szab.), hogy $Q \Rightarrow lf(S, R)$



1. Példa másik megoldása

- $I = (Q \wedge d \mid (n-r))$



részfeladat

- $A = \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$
  $n \quad d \quad r \quad h$
  $Q' = (Q \wedge d \mid (n-r) \wedge r \geq d)$
  $R' = (Q' \wedge h=i*d \wedge i \in \mathbb{N}$
  $\wedge h \leq r \wedge (i+1)*d>r)$

- $I \wedge \alpha \Rightarrow lf(S_0, I) = (Q \wedge d \mid (n-r+h))$
  ha $h=i*d$ ahol $i \in \mathbb{N}$ $(0<h \leq r)$,
  akkor $d \mid (n-r+i*d))$,
  Keressük meg a lehető legnagyobb $i*d$-t!

Részfeladat egyszerűsítése

- $A = \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$
  $n \quad d \quad r \quad h$
  $Q' = (Q \wedge d \mid (n-r) \wedge r \geq d)$
  $R' = (Q' \wedge h=i*d \wedge i \in \mathbb{N}^+ \wedge h \leq r \wedge (i+1)*d>r)$

1. $A = \mathbb{N} \times \mathbb{N} \times \mathbb{N}$
   $d \quad r \quad h$
   $Q' = (d=d' \wedge r=r' \wedge r \geq d)$
   $R' = (Q' \wedge d \mid h \wedge h \leq r \wedge h+d>r)$

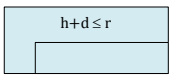
2. $A = \mathbb{N} \times \mathbb{N} \times \mathbb{N}$
   $d \quad r \quad h$
   $Q' = (d=d' \wedge r=r' \wedge r \geq d)$
   $R' = (Q' \wedge d \mid h \wedge h \leq r \wedge 2*h>r)$

Részfeladat megoldásai

- Legyen ciklus! (ciklus lev. szab.)
- Válasszunk invariánst! Az $I \wedge \neg\alpha \Rightarrow R$ érdekében $I \wedge \neg\alpha = R$.

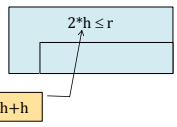
- $R' = (Q' \wedge d \mid h \wedge h \leq r \wedge h+d>r)$

1. $I' = (Q' \wedge d \mid h \wedge h \leq r)$



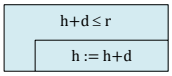
- $R' = (Q' \wedge d \mid h \wedge h \leq r \wedge 2*h>r)$

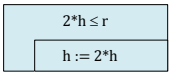
1. $I' = (Q' \wedge d \mid h \wedge h \leq r)$



Részfeladat megoldásainak folytatása

- $I' = (Q' \wedge d \mid h \wedge h \leq r)$





- h-t kéne növelni, amíg $h \leq r$, azaz $t = r-h$

2. $I \wedge \alpha \Rightarrow t>0$
   $I' \wedge h+d \leq r \Rightarrow r-h>0$
3. Ha $h := h+d$, akkor
   $I \wedge \alpha \wedge t=t_0 \Rightarrow lf(S_0, t<t_0)$
   $lf(S_0, r-h<t_0) = r-(h+d)<t_0$
4. $I \wedge \alpha \Rightarrow lf(S_0, I)$
   $I' \wedge h+d \leq r \Rightarrow I'^{h \leftarrow h+d} =$
   $(Q' \wedge d \mid (h+d) \wedge h+d \leq r)$

2. $I \wedge \alpha \Rightarrow t>0$
   $I' \wedge 2*h \leq r \Rightarrow r-h>0$
3. Ha $h := 2*h$, akkor
   $I \wedge \alpha \wedge t=t_0 \Rightarrow lf(S_0, t<t_0)$
   $lf(S_0, r-h<t_0) = r-2*h<t_0$
4. $I \wedge \alpha \Rightarrow lf(S_0, I)$
   $I' \wedge 2*h \leq r \Rightarrow I'^{h \leftarrow 2*h} =$
   $(Q' \wedge d \mid 2*h \wedge 2*h \leq r)$

Részfeladat megoldásainak befejezése

I′ = (Q′∧ d | h ∧ h≤r)

5. Q′⇏I′, tehát Q′ ⇏ lf(**while** α′ **do** S′$_0$ **od**) , R′)

Legyen S′ inkább szekvencia, amelynek második tagja az eddig levezetett ciklus, közbeeső állítása: I′

1. Egyrészt I′ ⇒ I′, tehát I′ ⇒ lf(**while** α′ **do** S′$_0$ **od**) , R′)
2. Másrészt Q′⇒I′$^{h\leftarrow d}$, mert Q′-ben d≤r, azaz Q′⇒lf(h:=d, I)

Így teljesül (szekv. lev. szab.), hogy Q′ ⇒ lf(S′, R′)

h := d
h+d ≤ r
h := h+d

h := d
2*h ≤ r
h := 2*h

1. Példa megoldásának három változata

r := n
r ≥ d
r := r–d

Nem lett gyorsabb, mint az első megoldás

r := n
r ≥ d
h := d
h+d ≤ r
h := h+d
r := r–h

r := n
r ≥ d
h := d
2*h ≤ r
h := 2*h
r := r–h

1. Példa 3. változatának javítása

r := n
r ≥ d
h := d
2*h ≤ r
h := 2*h
r := r–h

I′′=I′ ∧ s=2*h
t′ = r–h

r := n
r ≥ d
h, s := d, 2*d
s ≤ r
h, s := s, 2*s
r := r–h

2. Példa

- Számoljuk ki csak összeadással és kivonással két természetes szám szorzatát!
- A = ℕ × ℕ × ℕ
  x y z
  Q = (x=x′ ∧ y=y′)
  R = (z = x′*y′)

2. Példa folytatása

- Legyen ciklus! (ciklus lev. szab.)
- Válasszunk invariánst! Az I∧¬α⇒R érdekében I⇐R.
- R = (z = x′*y′) = (z = $\Sigma_{i=1..x'}$ y′)

1/a.
I = (y=y′ ∧ z + x*y = x′*y′)

2/a.
I = (Q ∧ 1≤k≤x+1 ∧ ∧ z = $\Sigma_{i=1..k-1}$ y)

1/b.
I = (z + x*y = x′*y′)

2/a.
I = (y=y′ ∧ z = $\Sigma_{i=x+1..x'}$ y)

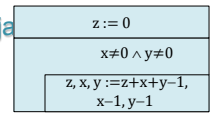
2. Példa 1/a. programja

z := 0
x≠0
z, x:=z+y, x–1

I = (y=y′ ∧ z + x*y = x′*y′)

I∧¬α⇒R: α = x≠0
I∧α⇒t>0: t = x
I∧α∧t=t_0⇒ lf(S_0,t<t_0): S_0: x := x–1
I∧α⇒lf(S_0, I): I ∧ x≠0⇒lf(x:=x–1, I)= I$^{x\leftarrow x-1}$ ∧ x>0 =
= (z + (x–1)*y = x′*y′ ∧ x>0) =
= (z – y + x*y = x′*y′ ∧ x>0)
z+y
Q⇏I: de Q⇒I$^{z\leftarrow 0}$= (0 + x*y = x′*y′)

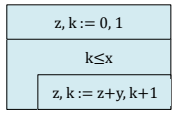
2. Példa 1/b. programja



I = (z + x*y = x'*y')

I∧¬α⇒R: α = x≠0 ∧ y≠0
I∧α⇒t>0: t = x+y
I∧α∧t=t$_0$⇒ If(S$_0$,t<t$_0$): S$_0$: x, y := x−1, y−1
I∧α⇒If(S$_0$, I): I ∧ x≠0 ∧ y≠0⇒If(x,y:=x−1,y−1, I)=
= I$^{x,y←x-1,y-1}$ ∧ x>0 ∧ y>0 =
= (z + (x−1)*(y−1) =x'*y' ∧ x>0 ∧ y>0)
z+x+y−1
Q⇏I: de Q⇒I$^{z←0}$= (0 + x*y = x'*y')

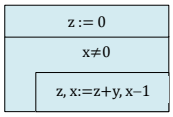
2. Példa 2/a. programja



I = (Q ∧ 1≤k≤x+1 ∧ z=Σ$_{i=1..k-1}$ y)

I∧¬α⇒R: α = k≤x
I∧α⇒t>0: t = x+1−k
I∧α∧t=t$_0$⇒ If(S$_0$,t<t$_0$): S$_0$: k := k+1
I∧α⇒If(S$_0$, I): I ∧ k≤x ⇒If(k:=k+1, I)= I$^{k←k+1}$ =
= (Q ∧ 1≤k+1≤x+1] ∧ z=Σ$_{i=1..k}$ y)
z+y
Q⇏I: de Q⇒I$^{z,k←0,1}$=
= (Q∧1≤1≤x+1∧0=Σ$_{i=1..0}$ y)

2. Példa 2/b. programja



I = (y=y' ∧ z = Σ$_{i=x+1..x'}$ y)

I∧¬α⇒R: α = x≠0
I∧α⇒t>0: t = x
I∧α∧t=t$_0$⇒ If(S$_0$,t<t$_0$): S$_0$: x := x−1
I∧α⇒If(S$_0$, I): I ∧ x≠0⇒If(x:=x−1, I)=I $^{x←x-1}$ ∧ x>0 =
= (y=y' ∧ z = Σ$_{i=x..x'}$ y ∧ x>0)
z+y
Q⇏I: de Q⇒I$^{z←0}$= (y=y' ∧ 0 = Σ$_{i=x+1..x'}$ y)

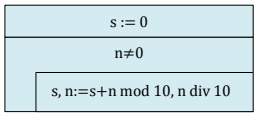
3. Példa

- Számoljuk ki természetes szám decimális számjegyeinek összegét!
- A = ℕ × ℕ
  n s
  Q = (n=n')
  R = (s = r(n'))

  ahol r(0) = 0
  r(n) = r(n div 10) + (n mod 10) (n>0)

- r(n') = r(n) + (... + ... + ...)^{s} = r(0) + (...)^{s}
- I = (s + r(n) = r(n'))

3. Példa folytatása



- Legyen ciklus!
- Válasszunk invariánst! Az I∧¬α⇒R érdekében I⇐R.

I = (s + r(n) = r(n'))
I∧¬α⇒R: α = n≠0
I∧α⇒t>0: t = n
I∧α∧t=t$_0$⇒ If(S$_0$,t<t$_0$): S$_0$: n := n div 10
I∧α⇒If(S$_0$, I): I ∧ n≠0⇒If(n:=n div 10, I)= I$^{n←n\ div\ 10}$
= (s + r(n div 10) = r(n'))
s+ n mod 10
Q⇏I: de Q⇒I$^{s←0}$= (0 + r(n) = r(n'))

4. Példa

- Számoljuk ki az n alatt a k binomiális együtthatót!
- A = ℕ × ℕ × ℕ
  n k b
  Q = (n=n' ∧ k=k' ∧ k∈[0..n])
  R = (b = $\binom{n'}{k'}$)

  ahol $\forall n \in \mathbb{N}: \binom{n}{0} = \binom{n}{n} = 1$

  $\forall n \geq 2, \forall k \in [1..n-1]: \binom{n}{k} = \binom{n}{k-1} \frac{n-k+1}{k}$

- $\binom{n'}{k'} = \binom{n'}{k} * (... * ... * ...)^{b} = \binom{n'}{0} * (...)^{b}$

4. Példa folytatása

b := 1
k≠0
b, k:=b*(n−k+1)/k, k − 1

- Legyen ciklus!
- Válasszunk invariánst! Az $I \wedge \neg\alpha \Rightarrow R$ érdekében $I \Leftarrow R$.

$I = (n=n' \wedge b*\binom{n}{k} = \binom{n}{k'})$

$I \wedge \neg\alpha \Rightarrow R$: $\alpha = k \neq 0$

$I \wedge \alpha \Rightarrow t>0$: $t = k$

$I \wedge \alpha \wedge t=t_0 \Rightarrow lf(S_0, t<t_0)$: S_0: $k := k - 1$

$I \wedge \alpha \Rightarrow lf(S_0, I)$: $I \wedge k \neq 0 \Rightarrow lf(k:= k-1, I) = I^{k \leftarrow k-1} \wedge k > 0$

$= (b * \binom{n}{k-1} = \binom{n}{k'} \wedge k > 0)$

$b * \frac{n-k+1}{k}$

$Q \Rightarrow I$: de $Q \Rightarrow I^{b \leftarrow 1} = (n=n' \wedge 1*\binom{n}{k} = \binom{n}{k'})$

Ciklus invariáns utófeltétel gyengítésével

- Elhagyás
  - Pl. : maradékos osztás: n, d, r : ℕ , d≠0
  - $d \mid (n-r) \wedge r<d \quad \rightsquigarrow \quad d \mid (n-r)$
- Intervallum szűkítése
  - Pl. : szorzás: x, y, z : ℕ (k : ℕ új változó)
  - $z=x'*y' = \Sigma_{i=1..y'} x' \rightsquigarrow z=\Sigma_{i=1..k} x'$
- Kiszámítás particionálása (rekurzív függvény)
  - Pl. : szorzás: x, y, z : ℕ
  - $z=x'*y' \quad \rightsquigarrow \quad z + x*y = x'*y'$
    $x*y = (x-1)*(y-1) + (x+y-1)$ [x, y := x − 1, y − 1]
    $x*y = (x-1)*y + (x+y-1)$ [x:=x − 1]
  - Pl. : Decimális számjegyek összege: n, d : ℕ
  - $d= r(n') \quad \rightsquigarrow \quad d + r(n) = r(n')$
    $r(n) = r(n \text{ div } 10) + (n \bmod 10)$ [n:=n div 10]

5. Példa

- Rendezünk növekedő sorrendbe a valós számokat tartalmazó tömböt!
- $A = \mathbb{R}^n$
  x
  $Q = (x=x')$
  $R = (sort(x[1..n]) \wedge x \in perm(x'))$

ahol

$sort(x[1..i])$: $\forall j \in [1..i-1]: x[j] \leq x[j+1]$
$x \in perm(x')$: x az x' egy permutációja
$x[a..b] \leq x[c..d]$: $\forall i \in [a..b]\ \forall j \in [c..d]: x[i] \leq x[j]$

$x[a..b] = \begin{cases} <x[a], \dots, x[b]> & \text{ha } 1 \leq a \leq b \leq n \\ <> & \text{ha } a > b \end{cases}$

5. Példa 1.megolda

i := 2
i≤n

- Legyen ciklus!
- Válasszunk invariánst!
  Gyengítsük az R-t az 1..n intervallum szűkítésével.

$I = (i \in [2..n+1] \wedge sort(x[1..i-1]) \wedge x \in perm(x'))$ i:ℤ

$I \wedge \neg\alpha \Rightarrow R$: $\alpha = i \leq n$

$I \wedge \alpha \Rightarrow t>0$: $t = n+1-i$

$I \wedge \alpha \wedge t=t_0 \Rightarrow lf(S_0, t<t_0)$: S_0: $i := i+1$

$I \wedge \alpha \Rightarrow lf(S_0, I)$: $I \wedge i \leq n \Rightarrow lf(i:=i+1, I) = I^{i \leftarrow i+1}$
$= (i+1 \in [2..n+1] \wedge sort(x[1..i]) \wedge x \in perm(x'))$
Beilleszt(x[i], x[1..i −1])

$Q \Rightarrow I$: de $Q \Rightarrow I^{i \leftarrow 2}$

5. Példa 1. folytatása: ciklusmag

- A ciklus magja legyen szekvencia!

$I \wedge \alpha \wedge t=t_0$
$I^{i \leftarrow i+1} \wedge t=t_0$
$I \wedge t<t_0$

Beilleszt(x[i], x[1..i −1])
i := i+1

- A szekvencia levezetési szabályának második feltétele teljesül:
  $I^{i \leftarrow i+1} \wedge n-i+1=t_0 \Rightarrow lf(i := i+1 , I \wedge n-i+1<t_0)$
- Specifikáljuk újra az első részt egyszerűsítve:
  $A = \mathbb{R}^n \times \mathbb{N}$
  x i
  $Q = (x=x'' \wedge i=i' \in [2..n] \wedge sort(x[1..i-1]))$
  $R = (i=i' \in [2..n] \wedge sort(x[1..i])$
  $\wedge x[1..i] \in perm(x'[1..i]) \wedge x[i+1..n]=x''[i+1..n])$

t=t_0-hoz elég, ha i nem változik

5. Példa 1. folytatása: részfeladat

- Bontsuk fel a specifikációt:
  $A = \mathbb{R}^n \times \mathbb{N} \times \mathbb{R}$
  x i a
  $Q = (x=x'' \wedge i=i' \wedge i \in [2..n] \wedge sort(x[1..i-1]))$

a := x[i]

$Q' = (x=x'' \wedge i=i' \in [2..n] \wedge sort(x[1..i-1]) \wedge a = x''[i])$

?

$R' = (i=i' \in [2..n] \wedge sort(x[1..i-1]) \wedge a = x''[i] \wedge j \in [1..i]$
$\wedge x[1..j-1]=x''[1..j-1] \wedge x[i+1..n]=x''[i+1..n]$
$\wedge \forall k \in [j..i-1]: x''[k]>a \wedge x[j+1..i] \leq x''[j..i-1]$
$\wedge (j=1 \vee x[j-1] \leq a))$

x[k+1]=x''[k]

x[j] := a

$R = (i=i' \in [2..n] \wedge sort(x[1..i])$
$\wedge x[1..i] \in perm(x''[1..i]) \wedge x[i+1..n]=x''[i+1..n])$

5. Példa 1. befejezés: részfeladat

- Legyen ciklus!
- Válasszunk invariánst! Hagyjunk el az R'-ből!

j := i
j≠1 ∧ x[j−1]>a
x[j], j :=x[j−1], j−1

I' = (i=i' ∧ j∈[1..i] ∧ x[1..j−1]=x'[1..j−1] ∧ x[i+1..n]=x'[i+1..n]
∧ ∀k∈[j..i−1]: x'[k]>a ∧ x[k+1]=x'[k] ∧ a = x'[i]) j:ℤ

I∧¬α⇒R: α = (j≠1 ∧ x[j−1]>a)

I∧α⇒t>0: t = j

I∧α∧t=t_0⇒ lf(S_0,t<t_0): S_0: j:= j −1

I∧α⇒lf(S_0, I): I'∧(j≠1 ∧ x[j−1]>a) ⇒ lf(j:=j−1 ,I')=$I'^{j\leftarrow j-1}$
= (i=i' ∧ j−1 ∈[1..i] ∧ x[1..j−2]=x'[1..j−2] ∧ x[i+1..n]=x'[i+1..n]
∧ ∀k∈[j−1..i−1]: x'[k]>a ∧ x[k+1]=x'[k] ∧ a = x'[i])

Q⇏I: de Q⇒$I'^{j\leftarrow i}$ ↑ x[j] ← x[j−1]

5. Példa 2. megold

- Legyen ciklus!
- Válasszunk invariánst! Gyengítsük az R-t az 1..n intervallum szűkítésével, de picit erősítsük is!

i := 1
i≤n−1

I = (i∈[1..n] ∧ sort(x[1.. i−1]) ∧ x∈perm(x')∧ x[1..i−1]≤x[i..n])

I∧¬α⇒R: α = i≤n−1 i:ℤ

I∧α⇒t>0: t = n−i

I∧α∧t=t_0⇒ lf(S_0,t<t_0): S_0: i := i +1

I∧α⇒lf(S_0, I): I∧ i≤n−1⇒lf(i:=i+1, I)= $I^{i\leftarrow i+1}$
= (i+1∈[1..n] ∧ sort(x[1..i]) ∧ x∈perm(x') ∧ x[1..i]≤x[i+1..n])

Q⇏I: de Q⇒$I^{i\leftarrow 1}$

5. Példa 2/a. folytatása: ciklusmag

I = (i∈[1..n] ∧ sort(x[1.. i−1]) ∧ x∈perm(x')∧ x[1..i−1]≤x[i..n])

- A ciklus magja legyen szekvencia!

I∧α ∧t=t_0
$I^{i\leftarrow i+1}$ ∧t=t_0
I ∧t<t_0

?
i := i+1

- A szekvencia levezetési szabályai közül a második teljesül
- Az első rész további kifejtésre szorul. Egyszerűsítve:
  A = $\mathbb{R}^n \times \mathbb{N}$
  x i
  Q = (x=x'' ∧ i=i'∈[1..n−1])
  R = (i=i'∈[1..n−1] ∧ x[1..i−1]= x''[1..i−1]
  ∧ x[i..n]∈perm(x''[i..n]) ∧ x[i]≤x[i+1..n])

5. Példa 2/a. folytatása: részfeladat

- Legyen ciklus!
- Válasszunk invariánst! Gyengítsük az R-t az i..n intervallum szűkítésével

j := n		
j>i		
x[j−1]>x[j]		
x[j], x[j−1] := x[j−1], x[j]	SKIP	

I = (i=i'∈[0..n] ∧ x[1..i−1]= x''[1..i−1]
∧ j∈[i..n] ∧ x[i..n]∈perm(x''[i..n]) ∧ x[j]≤x[j+1..n]) j:ℤ

I∧¬α⇒R: α = j>i

I∧α⇒t>0: t = j

I∧α∧t=t_0⇒ lf(S_0,t<t_0): S_0: j := j −1

I∧α⇒lf(S_0, I): I∧ j>i ⇒lf(j:=j−1, I)= $I^{i\leftarrow j-1}$ =
(i=i'∈[0..n] ∧ x[1..i−1]= x''[1..i−1]
∧ j−1∈[i..n] ∧ x[i..n]∈perm(x''[i..n]) ∧ x[j−1]≤x[j..n])
ha x[j−1]>x[j] akkor x[j−1]↔ x[j]

Q⇏I: de Q⇒$I^{i\leftarrow n}$

5. Példa 3. megold

- Legyen ciklus!
- Válasszunk invariánst! Gyengítsük az R-t az 1..n intervallum szűkítésével, de picit erősítsük is!

i := 1
i≤n−1

I = (i∈[1..n] ∧ sort(x[1.. i−1]) ∧ x∈perm(x')∧ x[1..i−1]≤x[i..n])

I∧¬α⇒R: α = i≤n−1 i:ℤ

I∧α⇒t>0: t = n−i

I∧α∧t=t_0⇒ lf(S_0,t<t_0): S_0: i := i +1

I∧α⇒lf(S_0, I): I∧ i≤n−1⇒lf(i:=i+1, I)= $I^{i\leftarrow i+1}$
= (i+1∈[1..n] ∧ sort(x[1..i]) ∧ x∈perm(x') ∧ x[1..i]≤x[i+1..n])

ind← $\min_{j=i..n}$ x[j] ↑ x[i] ↔ x[ind]

Q⇏I: de Q⇒$I^{i\leftarrow 1}$

5. Példa 2/b. folytatása: ciklusmag

- A ciklus magja legyen szekvencia!

I∧α ∧t=t_0
I∧α ∧t=t_0 ∧ ind= $\min_{j=i..n}$ x[j]
$I^{i\leftarrow i+1}$ ∧t=t_0
I ∧t<t_0

ind := $\min_{j=i..n}$ x[j]
x[i], x[ind] := x[ind], x[i]
i := i+1

- A szekvencia levezetési szabályai (mind a három) az eddigiek alapján nyilvánvalóan teljesül
- Az első tagprogram azonban további kifejtésre szorulna, de ez egy nevezetes feladat, a minimum kiválasztás, amelynek megoldása jól ismert.

Visszavezetés

- Nevezetes algoritmus minták: programozási tételek
- A programozási tétel sémáinak kitöltése
  - Állapottér egyeztetés
    - Közös alapállapottér megválasztása
    - Típus megvalósítás
  - Változónevek
  - Állandó értékű változók konstans értékkel való helyettesítése
  - Függvények megadása