# Visszalépéses keresés

Legyenek $U_1, U_2, ..., U_n$ tetszőleges véges halmazok. Legyen

$$U = U_1 \times U_2 \times ... \times U_n$$

Legyenek $\rho_0, \rho_1, \rho_2, ..., \rho_n$ állítások $U$-n:

$$\rho_i : U \to \mathbb{L}, \quad i = 0, 1, 2, ..., n, \quad \rho_0 = TRUE,$$

és amelyekre

$$\rho_{i+1} \Rightarrow \rho_i, \quad i = 0, 1, 2, ..., n-1$$

Legyen $\rho = \rho_n$.

Legyen $u', u'' \in U$, ahol

$$u' = (u'_1, u'_2, ..., u'_n) \quad \text{és} \quad u'' = (u''_1, u''_2, ..., u''_n)$$

Tegyük fel, hogy a $\rho_i$ csak az első $i$ változótól függ, azaz

$$\begin{aligned}&\forall i \in [1..n] : \forall u' \in U : \forall u'' \in U :\\ &(\forall j \in [1..i] : u'_j = u''_j) \to (\rho_i(u'_1, u'_2, ..., u'_i, u'_{i+1}, ..., u'_n) \leftrightarrow \rho_i(u''_1, u''_2, ..., u''_i, u''_{i+1}, ..., u''_n))\end{aligned}$$

$$\boxed{\text{Keressünk olyan } u \in U \text{, amelyre } \rho(u)!}$$

A feladat megoldását nehezíti, hogy az $U$ halmaz nem rendezett.

Legyen $|U_i| = \sigma_i, \quad i = 1, 2, ..., n$, ekkor léteznek a

$$\phi_i : [0, \sigma_i - 1] \to U_i, \quad i = 1, 2, ..., n,$$

kölcsönösen egyértelmű leképezések. Legyen

$$\phi_j(i) = u_i^{(j)}, \quad i \in [0, \sigma_j - 1], \quad j = 1, 2, ..., n$$

Tekintsük a $\Sigma_k = [0, \sigma_1 - 1] \times [0, \sigma_2 - 1] \times ... \times [0, \sigma_k - 1]$, $k = 1, 2, ..., n$, halmazokat. Legyen $\Sigma = \Sigma_n$.

Vezessük be a $\Sigma$ halmazon $\overline{\rho}_i$ állításokat a következő módon:

$$\overline{\rho}_i(i_1, i_2, ..., i_n) = \rho_i(u_{i_1}^{(1)}, u_{i_2}^{(2)}, ..., u_{i_n}^{(n)}), \quad i = 0, 1, 2, ..., n$$

Legyen $\overline{\rho} = \overline{\rho}_n$

Az eredeti feladat ekvivalens a következő feladattal:

$\boxed{\text{Keressünk olyan } i \in \Sigma \text{ , amelyre } \overline{\rho}(i)!}$

Ha a $\Sigma$ halmazt valamilyen formában rendezetté tesszük, akkor a feladat a **lineáris keresés 2.8** tételére vezethető vissza. Könnyen ellenőrízhetők a következő összefüggések:

$$\rho(u) \leftrightarrow \neg\exists m \in [1, n] : \neg\rho_m(u)$$

illetve

$$\overline{\rho}(u) \leftrightarrow \neg\exists m \in [1, n] : \neg\overline{\rho}_m(i)$$

A $\rho(u) \leftrightarrow$, illetve a $\overline{\rho}(u)$ állítások ilyen átfogalmazását első közelítésban azért célszerű választani, hogy a kitranszformálásuk közvetlen visszavezethető legyen a lineáris keresés tételére. Mind látni fogjuk, a későbbiekben célszerű lesz inkább az állításoknak a

$$\rho(u) \leftrightarrow \forall m \in [1, n] : \rho_m(u)$$

illetve

$$\overline{\rho}(u) \leftrightarrow \forall m \in [1, n] : \overline{\rho}_m(u)$$

érvényes formulák alapján származtatott ekvivalens alakját alkalmazni, ezek kitranszformálásához azonban külön tétel alkalmazására lesz szükség.

Tegyük fel, hogy $\Sigma$ valamilyen formában már rendezett és a keresésben már eljutottunk valamely $i$ pontig és $\overline{\rho}(i)$ nem teljesül. Akkor az $\overline{\rho}(i)$ fenti előállításából következik, hogy egyérteműen létezik olyan $j \in [1..n]$ index, amelyre $\overline{\rho}_j(i)$ nem teljesül, de

$$\forall k \in [1..j-1] : \overline{\rho}_k(i)$$

Figyelembe véve a

$$\rho_{i+1} \Rightarrow \rho_i, \quad i = 0, 1, 2, ..., n-1$$

tulajdonságokat, következik, hogy a $\overline{\rho}(i')$ biztosan nem teljesül, ha $i$ és $i'$ az első $j$ komponensben nem tér el. Rendezzük általában a $\Sigma_k$ halmazokat úgy, ha $i', i'' \in \Sigma_k$,

$$i' = (i'_1, i'_2, ..., i'_k) \quad \text{és} \quad i'' = (i''_1, i''_2, ..., i''_k)$$

akkor $\quad i' \leq i'' \quad$ akkor és csak akkor, ha $\quad \forall j \in [1, k] : (\forall l \in [1, j-1] : i'_l = i''_l) \rightarrow i'_j \leq i''_j$

Könnyen látható, hogy ezen rendezés mellett azok az elemek, amelyekről biztosan tudjuk, hogy ott a $\overline{\rho}$ értéke hamis, egy zárt intervallumot alkotnak $i$-től kezdve, ezért átléphetők a keresésben és meg tudjuk adni, azt az elemet is, ahol a keresést folytatni kell, feltéve, hogy ilyen elem még létezik.

Természetesen maradhatnánk az eredeti feladatnál is, mivel az $U$ halmaz is rendezhető, például a $\Sigma$ halmazban így bevezetett rendezés alapján. Mégis indokolt az állapottér transzformáció, mivel a $\Sigma$ halmaz rendezettsége közvetlenül az $N_0$ halmazon értelmezett rendezésre épül, amelyre viszont általában lehet hivatkozni a programokban.

Feltételezzük, hogy a $\rho_i, i = 0, 1, 2, ..., n,$ feltételek megengedettek.

Tekintsük az $I$ halmazt, amelyet úgy kapunk, hogy a $\Sigma$ halmazt a $-1$ (extra) elemmel kiegészítjük, amely minden $\Sigma$-beli elemnél kisebb.

$$I = \{-1\} \cup \Sigma \quad \text{és} \quad \forall i \in \Sigma : -1 \leq i$$

Legyen $i \in I$. Jelölje $i^+$ és $i^-$, az $i$ után közvetlen következő, ill. az $i$-t közvetlen megelőző elemet a $\Sigma_k$-ban a bevezetett rendezés szerint feltéve, hogy ilyen elem létezik.

Alkalmazzuk a **lineáris keresés 2.8** tételt a következő visszavezetéssel:

$$m \sim -1, \quad n \sim (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_n - 1), \quad \gamma \sim \overline{\rho},$$

ahol $\overline{\rho}$ kitranszformálandó feltétel a fenti formula alapján, szintén a **lineáris keresés 2.8** tétel alkalmazásával.

Mindezeknek megfelelően a keresett program a következő:

| |
|---|
| $i, u := -1, \downarrow$ |
| $\neg u \wedge i < (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_n - 1)$ |
| $\quad m, l := 0, \downarrow$ |
| $\quad m < n \wedge \neg l$ |
| $\quad\quad l := \neg\overline{\rho}_{m+1}(i^+)$ |
| $\quad\quad m := m + 1$ |
| $\quad u := \neg l$ |
| $\quad i := i^+$ |

Ha kihasználjuk a beágyazott **lineáris keresés 2.8** tétel által szolgáltatott plusz információt, és a $\overline{\rho}_i, \quad i = 0, 1, 2, ..., n$, rendszernek a bevezető részben deklarált tulajdonságait, akkor a keresést az előzőekben mondottak alapján lényegesen meggyorsíthatjuk. Ha egy adott $i^+$ keresési pontnál $\overline{\rho}(i^+)$ nem teljesül, akkor pontosan meg tudjuk mondani azt az intervallumot $i^+$-tól kezdődően, amelyben $\overline{\rho}$ biztosan nem teljesül..

A **lineáris keresés 2.8** tétel utófeltétele a következő:

$$R : \big(Q \wedge u = (\exists j \in [m..n] : \gamma(j) \wedge (u \rightarrow (i \in [m..n] \wedge \gamma(i) \wedge \forall j \in [m..i) : \neg\gamma(j))))\big)$$

Könnyen belátható, hogy igaz a tétel akkor is, ha a következő $R'$ utófeltételt tekintjük:

$$R' : \big(Q \wedge u = (\exists j \in [m..n] : \gamma(j) \wedge (u \rightarrow (i \in [m..n] \wedge \gamma(i) \wedge \forall j \in [m..i) : \neg\gamma(j))) \\ \wedge (\neg u \rightarrow i = n)\big)$$

Tehát, ha a beágyazott **lineáris keresés 2.8** tételnek megfelelő program eredményeként $l$ hamis, akkor $m = n$, ha viszont $l$ igaz, akkor $\forall j \in [1, m) : \overline{\rho}_j(i^+)$ teljesül, és $\overline{\rho}_m(i^+)$ nem teljesül.

Vezessük be $i \in \Sigma$, $i = (i_1, i_2, ..., i_n)$,valamint $k \in [1, n]$ esetén

$$\left[i_{1..k}\right] = (i_1, i_2, ..., i_k)$$

jelölést.

Ha tehát $l$ igaz, akkor

$$\overline{\rho}_{i+1} \Rightarrow \overline{\rho}_i, \quad i = 0, 1, 2, ..., n-1,$$

miatt a $\overline{\rho}$ biztosan hamis az

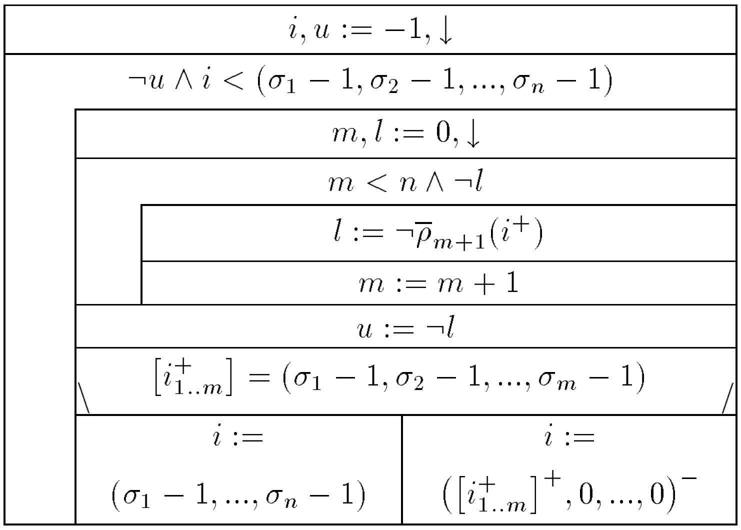
$$\left[i^+, \left[i^+_{1..m}\right]^+, 0, ..., 0\right)$$

intervallumban, ha $\left[i^+_{1..m}\right]^+$ létezik, ugyanis itt az az $\overline{\rho}_m$ biztosan hamis, mivel az index vektor első $m$ komponense változatlan és $\overline{\rho}_m(i)$, $m = 1, 2, ..., n$, a feltevés szerint $i$-nek csak az első $m$ komponensétől függ.

Hasonló meggondolaásból $\overline{\rho}$ biztosan hamis az

$$\left[i^+, (\sigma_1 - 1, ..., \sigma_n - 1)\right]$$

intervallumban, ha már $\left[i^+_{1..m}\right]^+$ nem létezik. Tehát a következő ellenőrizendő argumentum: $\left(\left[i^+_{1..m}\right]^+, 0, ..., 0\right)$ feltéve, hogy ez az argumentum létezik. Ennek megfelelően a keresés hatékonyabb formája:




mivel az elmondottak szerint a módosított ciklusmag is alkalmas választás ebben az eseben a **lineáris keresés 2.8** tételének levezetéséhez.

Könnyen belátható, hogy a **lineáris keresés 2.8** tétel igaz a következő formában is feltéve, hogy a $\gamma$ megengedett feltétel:

$Q : \left(m = m' \wedge n = n' \wedge m \leq n\right)$

$$R' : \big(Q \wedge u = (\exists j \in [m..n] : \gamma(j) \wedge (u \to (i \in [m..n] \wedge \gamma(i) \wedge \forall j \in [m..i) : \neg\gamma(j)))\wedge$$
$$\wedge\, (\neg u \to i = n)\big)$$

$$P : \big(Q \wedge i \in [m..n] \cup \{m-1\} \wedge u = (\exists j \in [m..i] : \gamma(j)) \wedge \forall j \in [m..i) : \neg\gamma(j)\big)$$

| | |
|---|---|
| $i, u := m, \gamma(m)$ | |
| $\neg u \wedge i < n$ | |
| | $i := i + 1$ |
| | $u := \gamma(i)$ |

Ezen tétel szerint az $i$-re rákövetkező előre is kiszámítható, és ezért előnyösebb a keresési probléma megoldását ezen tétere visszavezetni. Maga a levezetés alapvetően a

$$P^{i \leftarrow i+1\, u \leftarrow \gamma(i+1)} \Leftrightarrow P^{u \leftarrow \gamma(i)\, i \leftarrow i+1}$$

érvényességén múlik, ami a jelen esetben teljesül.

Ezt felhasználva a feladat megoldását a következő program adja:

| | | |
|---|---|---|
| $i := (0, 0, ..., 0)$ | | |
| $m, l := 0, \downarrow$ | | |
| $m < n \wedge \neg l$ | | |
| | $l := \neg\overline{\rho}_{m+1}(i)$ | |
| | $m := m + 1$ | |
| $u := \neg l$ | | |
| $\neg u \wedge i \neq (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_n - 1)$ | | |
| | $i := i^+$ | |
| | $m, l := 0, \downarrow$ | |
| | $m < n \wedge \neg l$ | |
| | | $l := \neg\overline{\rho}_{m+1}(i)$ |
| | | $m := m + 1$ |
| | $u := \neg l$ | |

Ha figyelembe vesszük az $u := \neg l$értékadást, az $l$ változó használatát megspórolhatjuk, és az $u := \neg l$ értékadást elhagyhatjuk, ha helyette alkalmasan magát az $u$ változót használjuk. Ennek megfelelően a $\overline{\rho}$ értéket kitranszformáló program és az $u := \neg l$ helyett a következő programot használhatjuk:

| | |
|---|---|
| $m, u := 0, \uparrow$ | |
| $m < n \wedge u$ | |
| | $u := \overline{\rho}_{m+1}(i)$ |
| | $m := m + 1$ |

Ezt a programot **KERESŐ** programnak nevezhetjük.

Ugyanezt a programot

$$m \sim 1, \quad n \sim m, \quad i \sim m, \quad \gamma(m) \sim \overline{\rho}_m()$$

visszavezetéssel is megkaphatjuk az alábbi tételből:

$Q : \big(m = m' \wedge n = n'\big)$

$$R' : \big(Q \wedge u = (\forall j \in [m..n] : \gamma(j) \wedge (\neg u \rightarrow (i \in [m..n] \wedge \neg\gamma(i) \wedge \forall j \in [m..i) : \gamma(j)))\wedge \\ \wedge\, (u \rightarrow i = n)\big)$$

$P : \big(Q \wedge i \in [m..n] \cup \{m-1\} \wedge \forall j \in [m..i) : \gamma(j) \wedge u = (\forall j \in [m..i] : \gamma(j))\big)$

<table>
<tr><td colspan="2" align="center">$i, u := m - 1, \uparrow$</td></tr>
<tr><td colspan="2" align="center">$i < n \wedge u$</td></tr>
<tr><td rowspan="2"></td><td align="center">$u := \gamma(i+1)$</td></tr>
<tr><td align="center">$i := i + 1$</td></tr>
</table>

Az utófeltétel alapján valóban $u \leftrightarrow \overline{\rho}(i)$.

Felhasználva ezt, és a következő ellenőrizendő argumentumra tett előző megállapítást, a feladat megoldását jelentő program:

<table>
<tr><td colspan="4" align="center">$i := (0, 0, ..., 0)$</td></tr>
<tr><td colspan="4" align="center">$m, u := 0, \uparrow$</td></tr>
<tr><td colspan="4" align="center">$m < n \wedge u$</td></tr>
<tr><td rowspan="2"></td><td colspan="3" align="center">$u := \overline{\rho}_{m+1}(i)$</td></tr>
<tr><td colspan="3" align="center">$m := m + 1$</td></tr>
<tr><td colspan="4" align="center">$\neg u \wedge i \neq (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_n - 1)$</td></tr>
<tr><td rowspan="6"></td><td colspan="3" align="center">$\big[i_{1..m}\big] = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1)$</td></tr>
<tr><td colspan="2" align="center">$i :=$<br>$(\sigma_1 - 1, ..., \sigma_n - 1)$</td><td align="center">$i :=$<br>$\big(\big[i_{1..m}\big]^+, 0, ..., 0\big)$</td></tr>
<tr><td colspan="3" align="center">$m, u := 0, \uparrow$</td></tr>
<tr><td colspan="3" align="center">$m < n \wedge u$</td></tr>
<tr><td rowspan="2"></td><td colspan="2" align="center">$u := \overline{\rho}_{m+1}(i)$</td></tr>
<tr><td colspan="2" align="center">$m := m + 1$</td></tr>
</table>

Az alábbi

<table>
<tr><td colspan="2">$\left[i_{1..m}\right] = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1)$</td></tr>
<tr><td>$i :=$<br>$(\sigma_1 - 1, ..., \sigma_n - 1)$</td><td>$i :=$<br>$\left(\left[i_{1..m}\right]^+, 0, ..., 0\right)$</td></tr>
</table>

programot **ELŐRELÉPŐ** programnak nevezhetjük. Az $i^+$ művelet nem megengedett, ezért ezt a programot tovább kell finomítani a következők szerint.

Az **ELŐRELÉPŐ** programban a $\Sigma_m$-ben bevezetett rendezés alapján az $\left[i_{1..m}\right]^+$ meghatározásához a következő specifikációval megadott feladatot kell megoldani:

$$\mathbf{A}: \underset{i}{\mathbf{\Sigma}}, \qquad \mathbf{B}: \underset{i'}{\mathbf{\Sigma}}, \qquad m \in [1, n] \quad \text{a feladat paramétere}$$

$$Q: \left(\left[i_{1..m}\right] = \left[i'_{1..m}\right]\right)$$

$$\begin{aligned} R: \big( & (\left[i'_{1..m}\right] = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1) \rightarrow i = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_n - 1)) \wedge \\ & \wedge ((\exists j \in [1, m] : i'_j \neq \sigma_j - 1) \rightarrow \\ & \rightarrow \forall j \in [1, m] : ((\forall k \in [j + 1, m] : i'_k = \sigma_k - 1) \wedge i'_j \neq \sigma_j - 1) \rightarrow \\ & \rightarrow (i_j = i'_j + 1 \wedge \forall k \in [j + 1, m] : i_k = 0 \wedge \forall k \in [1, j - 1] : i_k = i'_k))\big) \end{aligned}$$

Könnyen látható, hogy

$$Q \wedge \left[i_{1..m}\right] = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1) \Rightarrow R^{i \leftarrow (\sigma_1 - 1, ..., \sigma_n - 1)}$$

Legyen

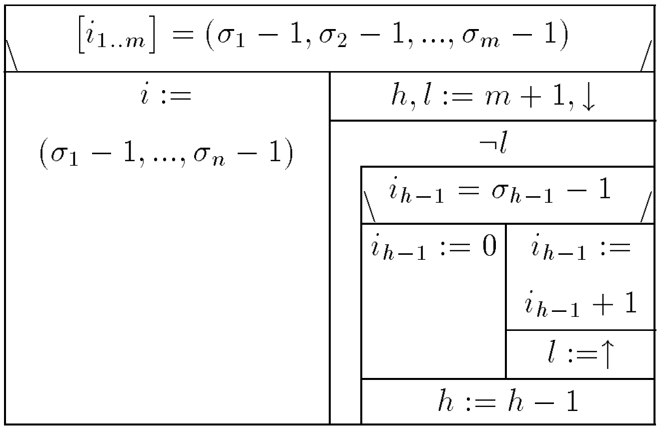
$$Q': \quad \left(Q \wedge \exists j \in [1, m] : i'_j \neq \sigma_j - 1\right)$$

és

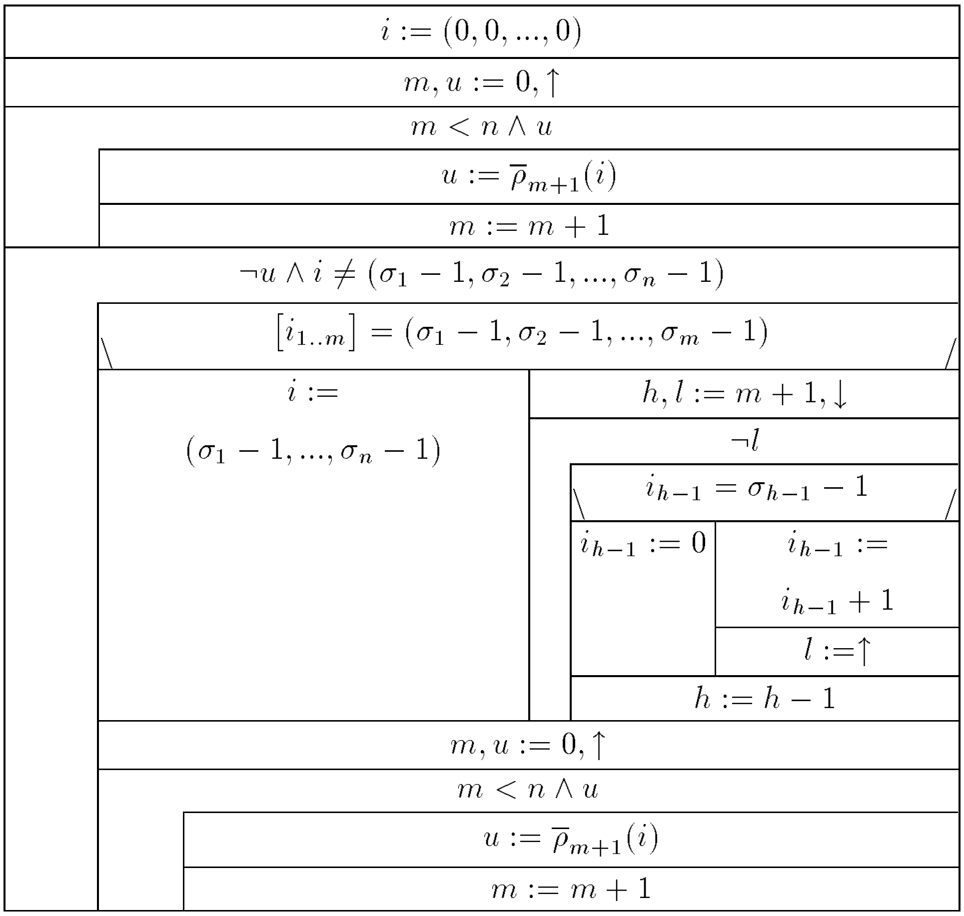
$$\begin{aligned} P: \big( & Q' \wedge h \in [0..m + 1] \wedge \\ & \wedge \forall k \in [h + 1, m] : i'_k = \sigma_k - 1 \wedge \forall k \in [h + 1, m] : i_k = 0 \quad \wedge \\ & \wedge (l \leftrightarrow \exists k \in [h, m] : i'_k \neq \sigma_k - 1) \quad \wedge \\ & \wedge (l \wedge h \neq m + 1 \rightarrow i_h = i'_h + 1) \quad \wedge \\ & \wedge (\neg l \wedge h \neq m + 1 \rightarrow i_h = 0) \quad \wedge \\ & \wedge \forall k \in [1, h) : i'_k = i_k\big) \end{aligned}$$

1. $Q' \Rightarrow P^{l,k\leftarrow m+1,\downarrow}$ teljesül.
2. $P \wedge \neg\pi \Rightarrow R$ teljesül, ha $\pi:\ l$.
3. $P \wedge \pi \Rightarrow t > 0$ teljesül, ha $t:\ k$.
4. $P \wedge \pi \wedge i'_{h-1} = \sigma_{h-1} - 1 \Rightarrow P^{h\leftarrow h-1}{}^{i_{h-1}\leftarrow 0}$
   valamint
   $P \wedge \pi \wedge i'_{h-1} \neq \sigma_{h-1} - 1 \Rightarrow P^{h\leftarrow h-1}{}^{l,i_{h-1}\leftarrow\uparrow,i_{h-1}+1}$
   teljesülnek.

A fentiek alapján a megoldást a következő program adja, amely egyúttal kitranszformálja a "rákövetkező" műveletet.



Mindezeket figyelembe véve, a kiinduló feladat megoldását a következő program adja:




Az **ELŐRELÉPŐ** program utófeltételéből következik, hogy a **KERESŐ** program

$$[i_{1..m}] \neq (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1)$$

esetén elég, ha a $[h, n]$ intervallumban keres, mivel az $[1, h - 1]$ intervallumban az $i$ index vektor komponensei nem változtak, és ebben az intervallumban $\overline{\rho}_k(i)$ állítások biztosan igazak $h - 1 < m$ miatt.

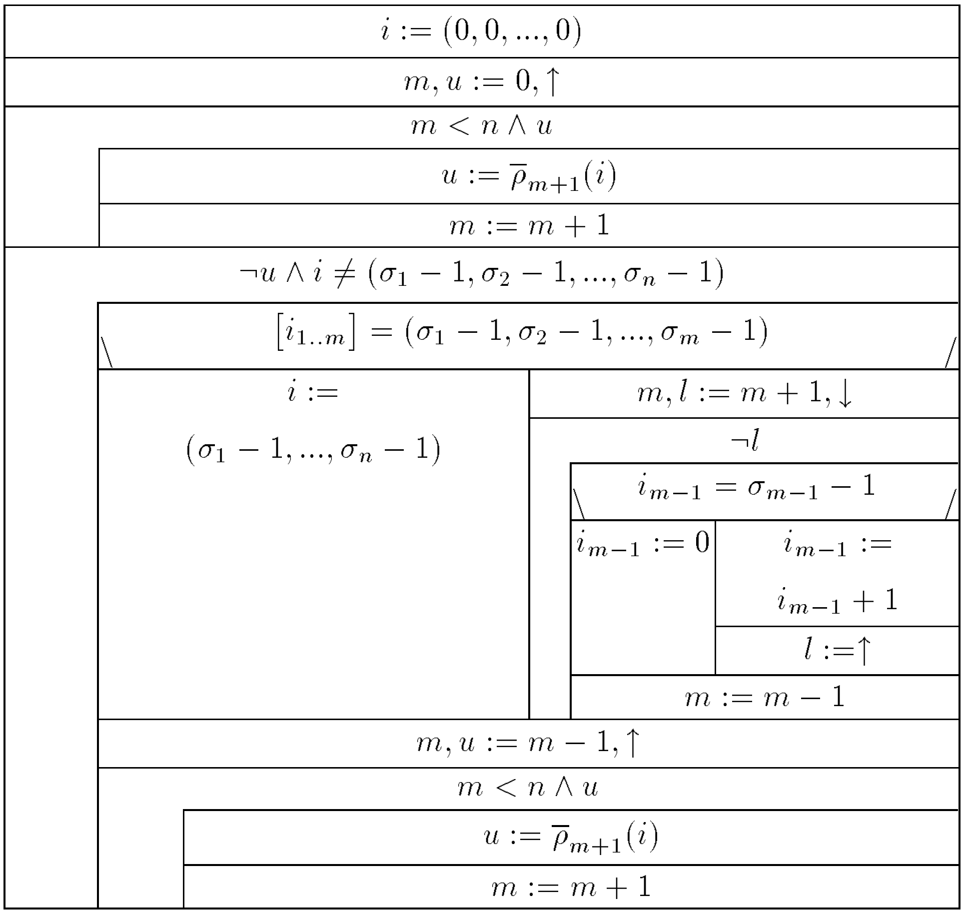
Hasonlóan, ha

$$[i_{1..m}] = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1),$$

akkor a keresést elég $m$-től indítani, hiszen ebben az esetben csak az $i_{m+1}, ..., i_n$ komponensek változtak, és a keresés az első lépésben véget fog érni.

Minden esetre $m$ a kereséshez alkalmas értéket fogja felvenni, ha az **ELŐRELÉPŐ** program belső ciklusában a $h$ váltzó helyett magát az $m$-et használjuk, és a **KERESŐ** programban a keresést az $[m, n]$ itervallumra redukáljuk.

A fentiek alapján a keresett program végleges formája a következő:


$i := (0, 0, ..., 0)$
$m, u := 0, \uparrow$
$m < n \wedge u$
$u := \overline{\rho}_{m+1}(i)$
$m := m + 1$
$\neg u \wedge i \neq (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_n - 1)$
$[i_{1..m}] = (\sigma_1 - 1, \sigma_2 - 1, ..., \sigma_m - 1)$
$i :=$
$(\sigma_1 - 1, ..., \sigma_n - 1)$
$m, l := m + 1, \downarrow$
$\neg l$
$i_{m-1} = \sigma_{m-1} - 1$
$i_{m-1} := 0$
$i_{m-1} :=$
$i_{m-1} + 1$
$l := \uparrow$
$m := m - 1$
$m, u := m - 1, \uparrow$
$m < n \wedge u$
$u := \overline{\rho}_{m+1}(i)$
$m := m + 1$