

Feladat: Keressük meg az f függvény értékei között a k szám p -edik előfordulását!

Átfogalmazás: Keressük azt az értéket, ahol a φ függvény először veszi fel a p értéket, ahol φ definíciója:

$$\varphi : [a - 1, b] \rightarrow \mathbb{N}_0$$

$$\varphi(a - 1) = 0, \forall i \in [a, b] : \varphi(i) = F(i, \varphi(i - 1)), \text{ ahol } F : [a, b] \times \mathbb{N}_0 \rightarrow \mathbb{R} \text{ definíciója:}$$

$$F(i, x) = \begin{cases} x & , \text{ ha } f(i) \neq k \\ x + 1 & , \text{ ha } f(i) = k \end{cases}$$

Paraméteres visszavezetés (k, p).

Specifikáció:

$$A = \mathbb{Z} \times \mathbb{Z} \times \mathbb{N} \times \mathbb{N} \times \mathbb{Z} \times \mathbb{L}$$

$$B = \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$$

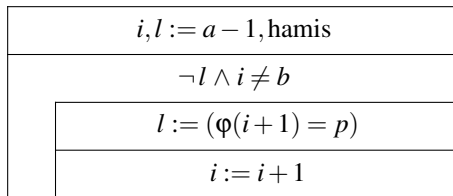
$$Q = (a = a' \wedge b = b' \wedge a \leq b + 1 \wedge k = k' \wedge p = p')$$

$$R = (Q \wedge l = (\exists j \in [a..b] : \varphi(j) = p) \wedge l \rightarrow (i \in [a..b] \wedge \varphi(i) = p \wedge f(i) = k))$$

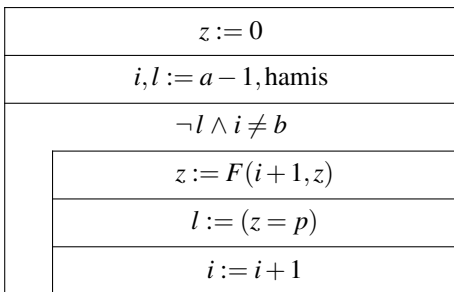
\Leftrightarrow (hiszen, ha $f(i) = k$, akkor éppen először veszi fel p értékét φ)

$$(Q \wedge l = (\exists j \in [a..b] : \varphi(j) = p) \wedge l \rightarrow (i \in [a..b] \wedge \varphi(i) = p \wedge \forall j \in [a..i - 1] : \varphi(j) \neq p))$$

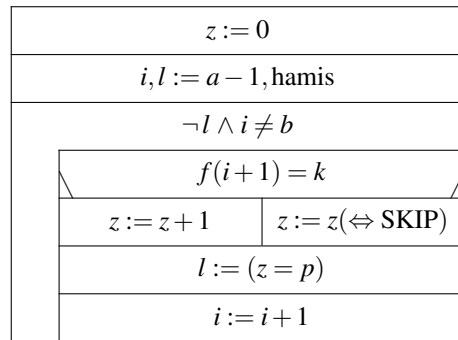
feladat		lin. ker. 2.8
a	\leftrightarrow	m
b	\leftrightarrow	n
$\varphi(i) = p$	\leftrightarrow	$\beta(i)$



Felhasználva a rekurzív függvény változóval való helyettesítésének programtranszformációs módszerét φ -re:



Ha ismerjük az elágazással definiált függvény kiszámításának programozási tételét, akkor ebből megkaphatjuk a megoldó programot, ha azt F -re alkalmazzuk:



Az egyik félévben ZH feladat volt, hogy a szöveget tartalmazó adott x vektorban keressük meg a harmadik szó kezdetét (ha van benne három szó). A szavakat vesszők (akár több is egymás mellett) választják el.

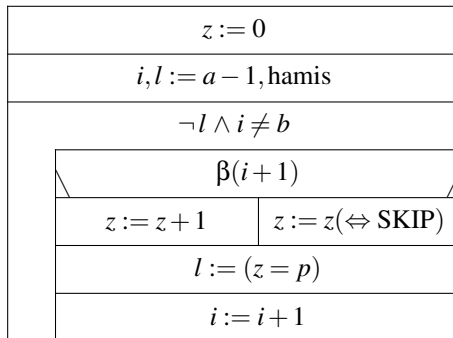
Ha az előző feladatra egy kicsit absztraktabban tekintünk, akkor ezt a ZH feladat megoldást is megkapjuk. Írjunk egy olyan programot, ami egy logikai β függvény értékei között a p -edik igaz előfordulást keresi! Ennek a programnak a megalkotása után a $\beta(i)$ -t $f(i) = k$ -val helyettesítve kaphatjuk az előző feladatot, tehát ez valóban általánosítás.

Az előző feladat megoldóprogramjában az $f(i+1) = k$ -t kell $\beta(i+1)$ -re cserélni nyilván, a hozzá tartozó visszavezetés és rekurzív függvény is analóg módon megkapható, ha minden $f(i) = k$ jellegű vizsgálatot $\beta(i)$ -vel helyettesítünk a megoldás során.

$$\varphi : [m-1, n] \rightarrow \mathbb{N}_0$$

$\varphi(m-1) = 0, \forall i \in [m, n] : \varphi(i) = F(i, \varphi(i-1))$, ahol $F : [m, n] \times \mathbb{N}_0 \rightarrow R$ definíciója:

$$F(i, x) = \begin{cases} x & , \text{ ha } \neg \beta(i) \\ x+1 & , \text{ ha } \beta(i) \end{cases}$$



Nézzük most a ZH feladat specifikációját:

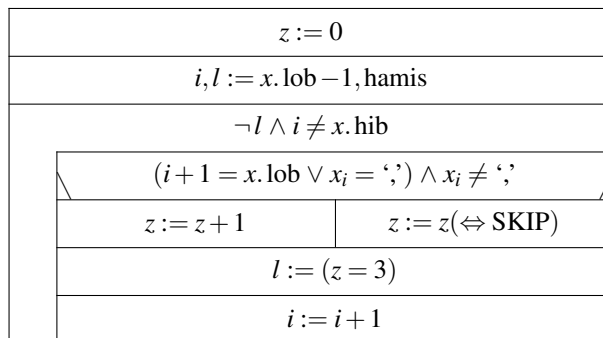
$$\mathbb{V} = \text{vect}(\mathbb{Z}, \text{Ch})$$

$$A = \mathbb{V} \times \mathbb{Z} \times \mathbb{L}$$

$$B = \mathbb{V}$$

$$Q = (x = x')$$

$R = (Q \wedge l = (\exists j \in [x.\text{lob}..x.\text{hib}] : \varphi(j) = 3) \wedge l \rightarrow (i \in [x.\text{lob}..x.\text{hib}] \wedge \varphi(i) = 3 \wedge \forall j \in [x.\text{lob}..i-1] : \varphi(j) \neq 3))$, ahol a φ -ben szereplő $\beta(i)$ most az, hogy $(i = x.\text{lob} \vee x_{i-1} = ',') \wedge x_i \neq ','$, ugyanis akkor kezdődik valahol új szó, ha betű az aktuális karakter és előtte vagy a vektor eleje vagy egy vessző van. Az is kiolvasható az utófeltételből, hogy az m -et $x.\text{lob}$ -bal, az n -et $x.\text{hib}$ -bel, a p -t pedig 3-mal helyettesítettük, így a megoldóprogram:



A struktogram csak olyan ún. lusta kiértékelés mellett helyes, ahol az x vektor lob indexénél korábbi elem hibás lekérdezésére már nem kerül sor a vele egy diszjunkcióban szereplő $i+1 = x.\text{lob}$ igaz volta miatt. Természetesen ha a mi számítógépünk nem lustán értékeli ki, akkor az elágazásfeltételt ki kell transzformálnunk és ezt a viselkedést egy külön elágazással megvalósíthatjuk.