Programozási tételek felsorolókra

Összegzés

Feladat: Adott egy *E*-beli elemeket felsoroló *t* objektum és egy $f{:}E\rightarrow H$ függvény. A *H* halmazon értelmezzük az összeadás asszociatív, baloldali nullelemes műveletét. Határozzuk meg a függvénynek a *t* elemeihez rendelt értékeinek összegét! (Üres felsorolás esetén az összeg értéke definíció szerint a nullelem: *0*).

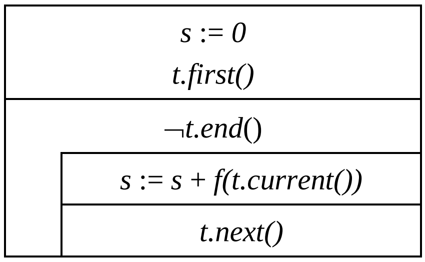
Specifikáció:

$A = (\ t{:}enor(E),\ s{:}H\ )$

$Ef = (\ t=t'\ )$

$Uf = (\ s = \sum_{e\in t'} f(e)\ )$

Algoritmus:



Számlálás

Feladat: Adott egy *E*-beli elemeket felsoroló *t* objektum és egy $\beta{:}E\rightarrow\mathbb{L}$ feltétel. A felsoroló objektum hány elemére teljesül a feltétel?

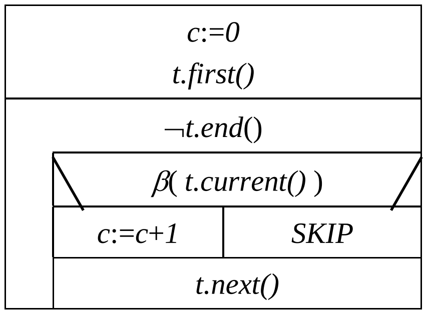
Specifikáció:

$A = (\ t{:}enor(E),\ c{:}\mathbb{N}\ )$

$Ef = (\ t=t'\ )$

$Uf = (\ c = \sum_{\substack{e\in t'\\ \beta(e)}} 1\ )$

Algoritmus:



Maximum kiválasztás

Feladat: Adott egy *E*-beli elemeket felsoroló *t* objektum és egy $f{:}E\rightarrow H$ függvény. A *H* halmazon definiáltunk egy teljes rendezési relációt. Feltesszük, hogy *t* nem üres. Hol veszi fel az *f* függvény a *t* elemein a maximális értékét?

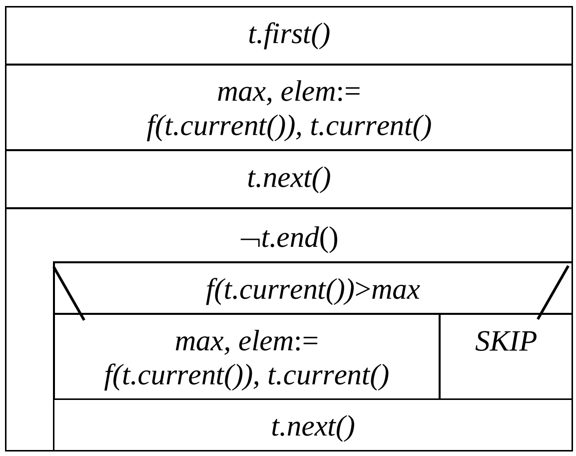
Specifikáció:

$A = (\ t{:}enor(E),\ max{:}H,\ elem{:}E\ )$

$Ef = (\ t=t' \wedge |t|>0\ )$

$Uf = (\ (max, elem) = \mathbf{max}_{e\in t'}\ f(e)\ )$

Algoritmus:



Kiválasztás

Feladat: Adott egy E-beli elemeket felsoroló t objektum és egy $\beta{:}E\to\mathbb{L}$ feltétel. Keressük a t bejárása során az első olyan elemi értéket, amely kielégíti a $\beta{:}E\to\mathbb{L}$ feltételt, ha tudjuk, hogy biztosan van ilyen.

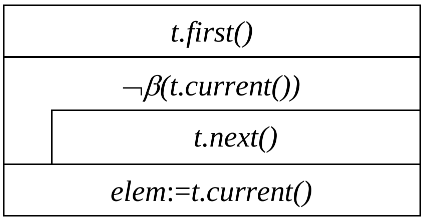
Specifikáció:

$A = (\ t{:}enor(E),\ elem{:}E\ )$

$Ef = (\ t=t' \wedge \exists i\in[1..\ |t|]{:}\ \beta(t_i)\ )$

$Uf = (\ (elem,t) = \underset{elem\in t'}{\textbf{select}}\ \beta(elem)\ )$

Algoritmus:



Lineáris keresés

Feladat: Adott egy E-beli elemeket felsoroló t objektum és egy $\beta{:}E\to\mathbb{L}$ feltétel. Keressük a t bejárása során az első olyan elemi értéket, amely kielégíti a $\beta{:}E\to\mathbb{L}$ feltételt.

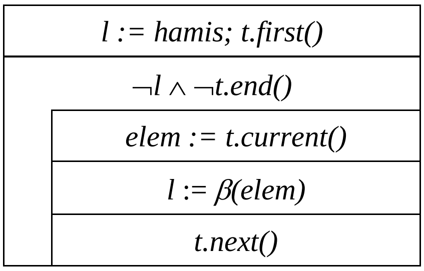
Specifikáció:

$A = (\ t{:}enor(E),\ l{:}\mathbb{L},\ elem{:}E\ )$

$Ef = (\ t=t'\ )$

$Uf = (\ (l,elem,t) = \underset{e\in t'}{\textbf{search}}\ \beta(e)\ )$

Algoritmus:



Feltételes maximumkeresés

Feladat: Adott egy E-beli elemeket felsoroló t objektum, egy $\beta{:}E\to\mathbb{L}$ feltétel és egy $f{:}E\to H$ függvény. A H halmazon definiáltunk egy teljes rendezési relációt. Határozzuk meg t azon elemeihez rendelt f szerinti értékek között a legnagyobbat, amelyek kielégítik a β feltételt.

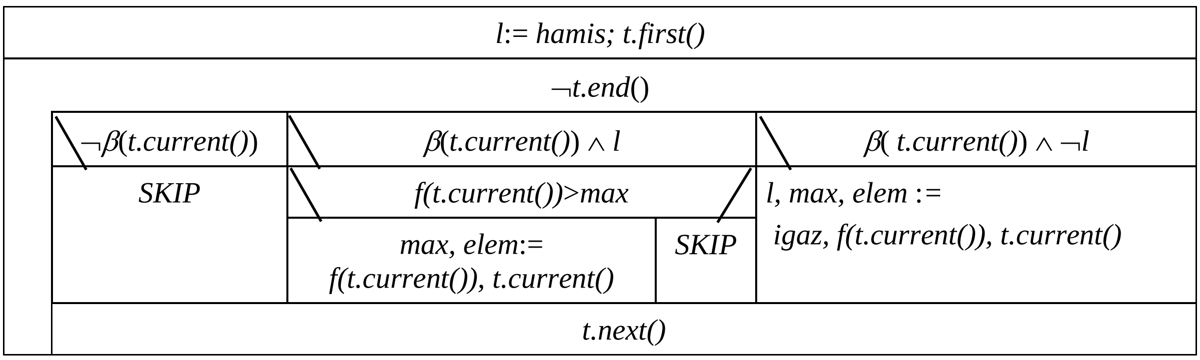
Specifikáció:

$A = (\ t{:}enor(E),\ l{:}\mathbb{L},\ max{:}H,\ elem{:}E\ )$

$Ef = (\ t=t'\ )$

$Uf = (\ (l,\ max,\ elem) = \underset{\substack{e\in t'\\ \beta(e)}}{\textbf{max}}\ f(e)\ )$

Algoritmus:



Megjegyzések:

1. A maximum keresés, a lineáris keresés, a kiválasztás nem a megtalált elem indexét, hanem a megtalált elemet adják vissza.
2. A lineáris keresésnél és kiválasztásnál az eredmények között szerepel maga a felsoroló is. Ennek az oka az, hogy ennél a két tételnél korábban is leállhat a feldolgozás, mint hogy a felsorolás véget érne, és ekkor maradnak még fel nem sorolt (fel nem dolgozott) elemek. Ezeket az elemeket további feldolgozásnak lehet alávetni, ha a felsorolót tovább használjuk. Felhívjuk azonban a figyelmet arra, hogy ha egy már korábban használt felsorolóval dolgozunk tovább, akkor nem szabad a *First()* művelettel újraindítani a felsorolást.
3. Kiválasztásnál nem kell a felsoroló által szolgáltatott értéksorozatnak végesnek lennie, hiszen ez a tétel más módon garantálja a feldolgozás véges lépésben történő leállását.
4. A programozási tételek alkalmazásakor – ha körültekintően járunk el – szabad az algoritmuson hatékonyságot javító módosításokat tenni. Ilyen például az, amikor ahelyett, hogy sokszor egymás után lekérdezzük a *t.Current()* értékét, azt annak első lekérdezésénél egy segédváltozóba elmentjük. A maximum kiválasztás illetve feltételes maximumkeresés esetén a feldolgozás eredményei között szerepel mind a megtalált maximális érték, mind pedig az elem, amelyhez a maximális érték tartozik. Konkrét esetekben azonban nincs mindig mindkettőre szükség. Például olyan esetekben, ahol a *f* függvény identitás, azaz egy elem és annak értéke megegyezik, a maximális elem és maximális érték közül elég csak az egyiket nyilvántartani az algoritmusban.
5. Nevezetes felsorolók alkalmazása esetén érdemes saját specifikációs jelöléseket bevezetni. Ilyenkor ugyanis a specifikációt nem egy absztrakt felsorolóra (t:$enor(E)$), hanem közvetlenül a feldolgozandó gyűjteményre (intervallumra, tömbre, halmazra, szekvenciális fájlra) fogalmazzuk a felsoroláshoz használt segédadatokkal.

   a) Egy <u>szekvenciális inputfájl</u> felsorolóját maga a szekvenciális inputfájl, az abból utoljára kiolvasott elem és az olvasás státusza reprezentálja. Szekvenciális inputfájlok feldolgozása esetén ehhez a megállapításhoz igazíthatjuk a specifikációs jelöléseket. Ilyenkor az állapottérben magát a szekvenciális inputfájlt vesszük fel, és az $e\in x$ (x a szekvenciális inputfájl) azt jelöli, hogy sorban egymás után ki akarjuk olvasni az x fájl (amely egy sorozat) elemeit. Ennél fogva a korábban bevezetett specifikációs jelölésekben szereplő $e\in t'$ (ahol t' a felsoroló kiinduló állapota) szimbólumot szekvenciális inputfájl bejárásakor kicserélhetjük az $e\in x'$ (x' a szekvenciális inputfájl kezdeti állapota) szimbólumra. Az $e\in x$ jelölés természetesen nem azt jelenti, hogy az utoljára kiolvasott elemet tartalmazó változót a programban is e-nek kell elnevezni, de célszerű ezt tenni.

   összegzés: $s = \sum_{e\in x'} f(e)$

   számlálás: $c = \sum_{\substack{e\in x' \\ \beta(e)}} 1$

   maximum kiválasztás: $max,elem = \boldsymbol{max}_{e\in x'} f(e)$

   feltételes maximumkeresés: $l,max,elem = \boldsymbol{max}_{\substack{e\in x' \\ \beta(e)}} f(e)$

   Láthattuk, hogy a kiválasztás és a lineáris keresés azelőtt is leállhat, hogy a felsorolás befejeződne, és ezért fontos eredménye ezen programozási tételeknek ez a be nem fejezett felsoroló is. Amennyiben az *st,e,x:read* műveletet használjuk a *x* szekvenciális inputfájl bejárására, akkor a „megkezdett” *t* felsorolót az *st, e, x* hármassal helyettesíthetjük a specifikációs jelölés baloldalán.

   lineáris keresés: $l,elem,(st,e,x) = \boldsymbol{search}_{e\in x'} \beta(e)$

   kiválasztás: $elem,(st,e,x) = \boldsymbol{select}_{elem\in x'} \beta(elem)$

Az *st* és az *e* azonban redundáns információt hordoz. Kiválasztásnál az *elem* azonos az *e*-vel, az *st* pedig biztosan *norm*, hiszen ilyenkor garantáltan találunk keresett elemet. Lineáris keresésnél *st=abnorm*, ha a keresés sikertelen (azaz *l* értéke hamis); sikeres termináláskor (ha *l* igaz) az *elem* azonos az *e*-vel, az *st* pedig biztosan *norm*. Ezért megengedjük a fenti jelölés minden olyan egyszerűsítését, ami nem megy az egyértelműség rovására. Ilyen például az alábbi:

lineáris keresés: $l,elem,x = \mathop{\boldsymbol{search}}\limits_{e \in x'} \beta(e)$

kiválasztás: $elem,x = \mathop{\boldsymbol{select}}\limits_{elem \in x'} \beta(elem)$

Gyakran előfordul, hogy egy már előre olvasott szekvenciális inputfájlra kell egy programozási tételt alkalmazni, azaz amikor a feldolgozandó elemek közül az első már az *e* segédváltozóban van, a többi pedig az *x* szekvenciális inputfájlban. Ilyenkor nem csak az *x* elemeit, hanem előtte még az *e* tartalmát is fel kell sorolnunk. Az algoritmusban ez csak annyit jelent, hogy a ciklust nem előzi meg a *First()* (szekvenciális inputfájlnál a *st,e,x:read*) művelet, specifikációban (példaként az összegzést és a kiválasztást adjuk meg) pedig az alábbi jelölést használjuk, ahol az előre olvasás eredményeként beolvasott elemet az *e'*, az olvasás utáni fájlt pedig az *x'* jelöli:

összegzés: $s = \sum\limits_{e \in (e',x')} f(e)$

kiválasztás: $elem,x = \mathop{\boldsymbol{select}}\limits_{elem \in (e',x')} \beta(elem)$

b) Egy *h* halmaz felsorolóját maga a *h* halmaz reprezentálja. Ilyenkor a specifikációnál $e \in t'$ (ahol t' a felsoroló kiinduló állapota) szimbólum helyett az $e \in h'$ (h' a halmaz kezdeti állapota) szimbólumot írhatjuk. Jelentése: vegyük sorban egymás után a halmaz elemeit.

c) Indexelhető gyűjtemények (vektor, mátrix, sorozat, stb.) esetén a felsorolót a gyűjtemény és az azon végigvezetett index (mátrixoknál indexpár) reprezentálják. Tulajdonképpen ilyenkor közvetlenül nem is a gyűjteményben tárolt értékeket, hanem azok indexeit soroljuk fel, hiszen egy indexhez bármikor hozzárendelhető az általa megjelölt érték. Ilyenkor például egy maximum kiválasztásnál az *f* függvény sohasem identitás, mert az *f* rendeli az indexhez (a felsorolt elemhez) a gyűjtemény megfelelő értékét. A specifikációkban szereplő *elem* ilyenkor egy indexet tartalmaz, ezért az alábbiakban *ind*-ként (mátrixok esetén dupla indexként: *ind, jnd*) jelenítjük meg. Ezen megfontolások miatt használhatjuk a korábbi fejezetek specifikációs jelöléseit, amelyből az is látható, hogy a korábbi intervallumos programozási tételek a felsorolós tételek speciális esetei.

összegzés: $s = \sum\limits_{i=m}^{n} f(v[i])$

számlálás: $c = \sum\limits_{\substack{i=m \\ \beta(v[i])}}^{n} 1)$

maximum kiválasztás: $(max, ind) = \mathop{\boldsymbol{max}}\limits_{i=m}^{n} f(v[i])$

feltételes maximumkeresés: $(l, max, ind) = \mathop{\boldsymbol{max}}\limits_{\substack{i=m \\ \beta(v[i])}}^{n} f(v[i]))$

lineáris keresés: $(l, ind) = \mathop{\boldsymbol{search}}\limits_{i=m}^{n} \beta(v[i])$

kiválasztás: $ind = \mathop{\boldsymbol{select}}\limits_{i \geq m} \beta(v[i])$

Már többször felhívtuk a figyelmet arra, hogy a keresés és kiválasztás előbb leállhat, mint maga a felsorolás. Vektorok esetén a még fel nem dolgozott elemek az *ind* index után állnak, ezért azok külön jelölésére nincs szükség.

d) Az $n \times m$-es mátrixokra bevezetett specifikációs jelölések (standard felsorolás esetén) csak abban térnek el a vektorokétól, hogy indexpárokat tartalmaznak.

összegzés:	$s = \sum_{i,j=1,1}^{n,m} f(a[i,j])$
számlálás:	$c = \sum_{\substack{i,j=1,1 \\ \beta(a[i,j])}}^{n,m} 1$
maximum kiválasztás:	$(max, ind, jnd) = \mathop{\boldsymbol{max}}_{i,j=1,1}^{n,m} f(a[i,j])$
feltételes maximumkeresés:	$(l, max, ind, jnd) = \mathop{\boldsymbol{max}}_{\substack{i,j=1,1 \\ \beta(a[i,j])}}^{n,m} f(a[i,j])$
lineáris keresés:	$(l, ind, jnd) = \mathop{\boldsymbol{search}}_{i,j=1,1}^{n,m} \beta(a[i,j])$
kiválasztás:	$(ind, jnd) = \mathop{\boldsymbol{select}}_{i \geq 1, j=1}^{m} \beta(a[i,j])$