PROGRAMOZÁSMÓDSZERTAN
4. ELŐADÁS'2004
(VÁZLAT)

1. A TÉTELEK SZIGNATÚRÁJÁRÓL

1.1. Formája és célja

Induljunk ki egy ismert tételből:

Megszámolás(H^*, $\mathcal{F}(H,\mathbb{L})$):$\mathbb{N}$

Szignatúra
Milyen absztrakt paraméterei vannak (*értelmezési tartomány*);
milyen absztrakt eredményt állít elő (*értékkészlet*).
Csak az alaphalmazt jelöljük; anonim módon.

Be: $N\in\mathbb{N}$, $X\in H^*$, $T:H\to\mathbb{L}$
Ki: $Db\in\mathbb{N}$
Ef: … ez most nem érdekes …
Uf: ... ez most nem érdekes …
Def: … ez most nem érdekes …

Specifikáció
A paraméterek *nev*et kapnak.
A *sorozatok hossz*ának –ha érdekes– itt rendelünk önálló (függő/független) változót.

Alg:
. . . *ez most nem érdekes* . . .

Algoritmus

A tételekről ezt a szignatúra-adta „kevés" információt arra használhatjuk föl, hogy „megsaccoljuk" egy adott feladat megoldására mely **tételek** valamilyen **kombinációja** képzelhető el. A „megsaccolni" ige arra utal, hogy ha azt nem is tudjuk előre garantálni: valóban mely tételek lesznek nyerők, az biztos, hogy **milyenek zárhatók ki eleve**.

1.2. A tételek csoportjai és a szignatúra

A legjellegzetesebb szempont a csoportosításra: a benne szereplő sorozatok száma.

Értelmezési tartománybeli sorozatok száma	→	Értékkészletbeli sorozatok száma	Tétel
1	→	0 [az érték egy skalár]	Sorozatszámítás Eldöntés Kiválasztás Keresés Megszámolás Maximumkiválasztás

1	→	1	Másolás Kiválogatás Rendezés(ek)
1	→	sok	Szétválogatás
sok	→	1	Metszet Egyesítés Összefuttatás

1.3. A tételek összeépítéséről előzetesen

1.3.1. Első példafeladat

Egy egyetem hallgatóinak adatai (név, szak, évfolyam, átlag) alapján adjuk meg a hallgatók közül kik voltak a legjobbak az egyes évfolyamokban!

1.3.2. Megfontolás

A feladat specifikációjában ***egy sorozat***ból kell kiindulni, s ***egy sorozatot***/5-elemű ***skalár***t[1] kell előállítani.

'***Egy*** sorozat → ***egy*** sorozat' esetén szóba jöhető tételek:

- másolás Hossz(Spec.Be)≠Hossz(Spec.Ki) ⇒👎
- rendezés Hossz(Spec.Be)≠Hossz(Spec.Ki) ⇒👎
- kiválogatás 👍

Akkor legyen a '**Kiválogatás**'?!?

Döntésünk még nem tűnik véglegesnek, hiszen ***nem létezik egyértelmű (univerzális) T tulajdonság***. (Pontosabban *5* hasonló tulajdonság van: x-évfolyambeli, x=1..5) Arra viszont rávilágít, hogy az eredménysorozat minden elemét „azonos logika" szerint kell előállítani. Figyeljünk most erre a „logikára"! Minden „legjobb" a bemenő sorozat egy adott részsorozatára: az adott évfolyamhoz tartozó hallgatókra vonatkozik. Tehát ha tudnánk is „mennyi a legjobb", hogyan korlátozzuk a megfelelő részsorozatra? Tehát, hogy szűrjük ki a bemenő sorozat megfelelő részét? Az előbbi tételcsoportos okoskodást újból végig víve már bizonyosra vehető konzekvenciaként fogalmazhatjuk meg a tételt: **kiválogatás**. A „legjobbság" valami **maximumkiválasztás**-félére utal, amit éppen valamelyik évfolyambelire kell alkalmazni.

Azaz, jelöljük

az egyetemi hallgatók sorozatát: `E`-vel (melynek van pl. `E.Átlag`-gal jelölt komponense),

a legjobbak sorozatát: `L`-lel,

így az algoritmus tételfüggvényekkel kifejezve:

`L`$_i$`:=`***Maximumkiválasztás*****(*****Kiválogatás*****(**`E,Évfolyambeli(.,i)`[2]**)**`,≤`$_{Átlag}$**)** `(i=1..5)`

ahol

`Évfolyambeli:` ℕ×ℕ → 𝕃

`Évfolyambeli(x,e):=(x=e)` [3]

Elképzelhető lenne így is:

`Évfolyambeli(i,e):=(E`$_i$`.Évfolyam=e)`

[1] Mert, hogy előre ismert és kicsiny számú az eredmény. Így nagy a kísértés, hogy „egységnek" érezzük.

[2] `Évfolyambeli(.,e)` azt hangsúlyozza, hogy ebben a szituációban a kétváltozós függvény valójában csak az első változójától függ, a második rögzített: 1 vagy 2 vagy … vagy 5 (`e`) értékű.

$\leq_{\text{Átlag}} : \mathbb{N}\times\mathbb{N} \to \mathbb{L}$	vagy	$\leq_{\text{Átlag}} : \mathbb{R}\times\mathbb{R} \to \mathbb{L}$
`≤Átlag(i,j):=Ei.Átlag ≤ Ej.Átlag`		`≤Átlag(ái,áj):=ái ≤ áj` [4]

Konkretizáljunk, s közben ellenőrizzük a minimális elfogadhatóságot: az összeilleszthetőséget!

Kiválogatás$(\mathbf{H}^*, \mathcal{F}(\mathbf{H},\mathbb{L})):\mathbb{N}^*$ vagy $\mathbf{H}^*$	***Maximumkiválasztás***$(\mathbf{H}^*, \mathcal{F}(\mathbf{H}\times\mathbf{H},\mathbb{L})):\mathbb{N}$ vagy $\mathbf{H}$
$\mathbf{H}^*$=(**Név×Szak×Évfolyam×Átlag**)*∍E – *a hallgatók adatai* *A továbbiakban jelöljük a struktúrát így:* **HallgatóiAdat**	
$\mathcal{F}(\mathbf{H},\mathbb{L})$∍`Évfolyambeli(.,e)` függvények – *megfelelő évfolyambeli-e*?: l. fent	
$\mathbb{N}^*$ vagy **HallgatóiAdat*** – *a megfelelő hallgatók indexei, vagy maguk a hallgatók*	$\mathbf{H}^*=\mathbb{N}^*$ vagy **HallgatóiAdat*** – *a megfelelő hallgatók indexei, maguk a hallgatók*
	$\mathcal{F}(\mathbf{H}\times\mathbf{H},\mathbb{L})=\mathcal{F}(\mathbb{N}\times\mathbb{N},\mathbb{L})\ni\leq_{\text{Átlag}}$ rendezési reláció vagy $=\mathcal{F}($**HallgatóiAdat×HallgatóiAdat**$,\mathbb{L})\ni$ $\leq_{\text{Átlag}}$ rendezési reláció – *kisebb-e*? l. fent
	$\mathbb{N}$ vagy **HallgatóiAdat** – *a legjobb átlagú hallgató indexe, vagy a hallgató maga*

Formálisan tehát elképzelhető a fenti tételkombináció.

Megjegyzések:

1. Arra persze figyelnünk kell, hogy a Maximumkiválasztás tétel válaszindexe az **eredeti** sorozatra vonatkozzék!
2. Mivel a Maximumkiválasztás **nem** működik **üres** sorozatra (l. MaximumKiválasztás.Ef), ezért törődni kell ezzel is (még ha a konkrét feladatban nem túl valószínű az előfordulása).
3. A tételek kombinálása általában **nem** eredményez **optimális** algoritmust, de reményt ad a **helyességre**. (Az optimalizálásról lesz még szó: a „Programtranszformációk…” előadásban.)

1.3.3. Második példafeladat

Egy egyetem hallgatóinak adatai (név, szak, évfolyam, átlag) alapján adjuk meg a hallgatók közül kik voltak a legjobbak!

---

[3] Az x helyébe valamely `Ei.Évfolyam`-t képzelhetjük.

[4] Egy lehetséges hívása: `Ei.Átlag≤Ej.Átlag`, azaz `ái` helyébe `Ei.Átlag` és `áj` helyébe `Ej.Átlag` helyettesítendő.

1.3.4. Megfontolás

Az előző feladathoz képest az eltérés az, hogy az *összes* legjobbat (ha több azonos átlagú van) kell évfolyamtól függetlenül kikeresni.

A feladat specifikációjában ***egy sorozat*ból** kell kiindulni, s ***egy sorozatot*** kell előállítani. A számba jövő tételek közül megint a **kiválogatás** marad fenn, amely egymagában megint nem jelentheti a megoldást. Miért nem?

A legjobbságról (megint) a **maximumkiválasztás**ra (mégpedig annak elemértékű, 2. variánsára) asszociálhatunk, hogy végül is az alábbi megoldásvázlatot kapjuk:

```
L:=Kiválogatás(E,Legjobb)
```

ahol

```
Legjobb: ℝ → 𝕃
Legjobb(x):=( x=Maximumkiválasztás(E,≤Átlag) ) [5]
```

A formális ellenőrzést most nem részletezem.

Végezetül hangsúlyozni szeretném, hogy a fenti gondolatmenetnek semmi köze a formális levezetéshez, csupán azt illusztrálja, hogy miképpen lehet ötletekhez jutni pusztán a tételek szignatúrájából a megfelelő megoldás megsejtéséhez!

2. TOVÁBBI PROGRAMOZÁSI TÉTELEK

2.1. Szétválogatás tétele

Szétválogatás$(H^*,\mathcal{F}_i(H,\mathbb{L})(i=1..K)):(H^*)^K$

Be: $N\in\mathbb{N},\ X\in H^*,\ K\in\mathbb{N},\ T_i:H\to\mathbb{L}\ (i=1..K)$

Ki: $Db_i\in\mathbb{N},\ Y_i\in H^*\ (i=1..K)$

Ef: $N=Hossz(X)\ \wedge\ K\geq 2\ \wedge\ T_i$ *H-particionálása* $(i=1..K)$

Uf: $Db_i=\sum_{j=1}^{N}\chi(T_i(x_j))\ \wedge\ Y_i\in H^{Db_i}\ \wedge\ Y_i\subseteq X\ \wedge\ \forall j\in[1..Db_i]:\ T_i(y_{ij})\ \ (i=1..K)$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Konstans K:Egész(???)
Függvény T(Konstans i:1..K, x:TH):Logikai
  ... feladatfüggő számítás: x Ti tulajdonságú-e? ...
Függvény Vége.
Típus TDbk=Tömb(1..MaxN:Egész)
      THkk=Tömb(1..MaxN:THk)  [sorozatok sorozata]
```

[5] Világos, hogy `x` helyébe az `Ei.Átlag`-k kerülnek.

```
Eljárás Szétválogatás(Konstans N:Egész,X:THk, K:Egész
                      Változó  Db:TDbk,Y:THkk):
  Változó
    i:Egész
  Db(1..K):=0
  Ciklus i=1-től N-ig
    j:=MelyikTIgaz(X(i),K)
    Db(j):+1; Y(j,Db(j)):=X(i)
  Ciklus vége
Eljárás vége.
Függvény MelyikTIgaz(Konstans e:TH, K:Egész): Egész
  Változó
    i:Egész
  i:=1
  Ciklus amíg i≤K és nem T(i,e)
    i:+1
  Ciklus vége
  MelyikTIgaz:=i
Függvény vége.
```

Megjegyzések:

1. Természetesen értelmes változat: a szétválogatás *indexsorozat*okra vonatkozó is.
2. Egyszerűsíthető némileg a kétfelé szétválogatás (K=2) algoritmusa (hisz a $T_2=\neg T_1$). Érdekes alesetei:
   a. Két sorozatba,
   b. Egy sorozatba (elől a T-, hátul a nem-T-tulajdonságúak),
   c. Helyben.

Ez utóbbi eset gyakran előfordul, emiatt foglalkozunk részletesen.

SzétválogatásHelyben$(H^* \mathcal{F}(H,\mathbb{L})):(H^*)$

)-: Vegyük észre: a „helybenség”-et a függvényszerű leírás nem támogatja! :-(

Be: $N\in\mathbb{N}$, $X\in H^*$, $T:H\to\mathbb{L}$

Ki: $Db\in\mathbb{N}$ *[*$X\in H^*$ *nem jelenik meg explicite, helyette* $X'\in H^*$*-t fogunk írni]*

Ef: $N=Hossz(X)$

Uf: $Db=\sum_{i=1}^{N}\chi(T(x_i)) \wedge X'\in\mathcal{P}(X) \wedge \forall i\in[1..Db]: T(x'_i)$

Megjegyzés:

$\mathcal{P}(X)$ az X permutációhalmazát jelöli; továbbiakban is alkalmazandó megállapodás lesz, hogy a „*helybenség*” (azaz, ha ugyanazon adat a bemeneten is és a kimeneten is megjelenik) kifejezésére használjuk az *aposztróf* jelet.

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
```

```
Eljárás SzétválogatásHelyben(Konstans N:Egész , X:THk
                             Változó Db:Egész, X:THk):
  Változó
    i:Egész
    h:TH
  Db:=0
  e:=1; u:=N; h:=X(e)
  Ciklus amíg e<u
    Ciklus amíg e<u és nem T(X(u)) [T-tulajdonságú keresése hátul]
      u:-1
    Ciklus vége
    Ha e<u akkor X(e):=X(u); e:+1
    Ciklus amíg e<u és T(X(e)) [nem T-tulajdonságú keresése elöl]
      e:+1
    Ciklus vége
    Ha e<u akkor X(u):=X(e); u:-1
  Ciklus vége
  X(e):=h
  Ha T(X(e)) akkor Db:=e különben Db:=e-1
Eljárás vége.
```

Megjegyzés:

1. spórolhatnánk 2 felesleges feltételvizsgálatot, ha a **második belső ciklust és az őt követő elágazást** az első feltételes utasítás akkor-ágához illesztenénk.
2. Érdemes meggondolni a *program*specifikáció utófeltételét!

2.2. Egyesítés tétele

Egyesítés$(H^*, H^*) : H^*$

Be: $N \in \mathbb{N}$, $X \in H^*$, $M \in \mathbb{N}$, $Y \in H^*$

Ki: $Db \in \mathbb{N}$, $Z \in H^*$

Ef: $N=Hossz(X) \wedge HalmazFölsorolás(X) \wedge M=Hossz(Y) \wedge HalmazFölsorolás(Y)$

Uf: $Db = N + \sum_{i=1}^{M} \chi(y_i \notin X) \wedge Z \in H^{Db} \wedge HalmazFölsorolás(Z) \wedge \forall i \in [1..Db]:(z_i \in X \vee z_i \in Y)$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Egyesítés(Konstans N:Egész,X:THk, M:Egész,Y:THk
                  Változó  Db:Egész,Z:THk):
  Változó
    i:Egész
```

```
  Z:=X; Db:=N  [feltöltés X elemeivel]
  Ciklus j=1-től M-ig
    i:=1
    Ciklus amíg i≤N és Y(j)≠X(i)
      i:+1
    Ciklus vége
    Ha i>N akkor Db:+1; Z(Db):=Y(j)
  Ciklus vége
Eljárás vége.
```

Megjegyzések:

1. Észre vehető, hogy a belső ciklus (és „környéke”) egy Eldöntés tételt formál, míg a külső ciklus (és környéke) egy Kiválogatás tételt; ill. a törzs első sora egy triviális másolás tételalkalmazás.
2. Gondolja ezt meg úgy, hogy megpróbálja a fenti algoritmust a korábbiakban taglalt *tételfüggvény*es leírási móddal újrafogalmazni!
3. Könnyen belátható az alábbi állítás: *helyes végállapotban Z⊆X&Y is teljesül.* (Akár a programspecifikáció részeként is fölfoghatnánk.)
4. Miért nem lenne helyes a specifikáció kimeneti részében a „Db∈ℕ, Z∈H^{N+M}”?

2.3. Metszetképzés tétele

Metszet(H^*,H^*):H^*

Be: $N\in\mathbb{N}$, $X\in H^*$, $M\in\mathbb{N}$, $Y\in H^*$

Ki: $Db\in\mathbb{N}$, $Z\in H^*$

Ef: $N=Hossz(X)$ ∧ *HalmazFölsorolás(X)* ∧ $M=Hossz(Y)$ ∧ *HalmazFölsorolás(Y)*

Uf: $Db=\sum_{i=1}^{N}\chi(x_i\in Y)$ ∧ $Z\subseteq X$ ∧ $\forall i\in[1..Db]: z_i\in Y$

Állítás: *helyes végállapotban Z-re teljesülni fog: HalmazFölsorolás(Z)*

Biz.: (indirekt)

Tfh. ¬ *HalmazFölsorolás(Z)* $\Rightarrow$ **$\exists k,l: k<l \wedge z_k=z_l$.** **(1)**

Helyes végállapotban: $Z\subseteq X \Rightarrow \forall i: \exists j: z_i=x_j, \Rightarrow$ **$k: \exists j: z_k=x_j, l: \exists j': z_l=x_{j'} \wedge$**

$k<l \Rightarrow j<j'$ (2)

(1) ∧ **(2)** $\Rightarrow j\neq j' \wedge x_j=x_{j'}$

Ez ellentmond az Ef-beli elvárással: HalmazFelsorolás(X).

♦

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Metszet(Konstans N:Egész,X:THk, M:Egész,Y:THk
                Változó  Db:Egész,Z:THk):
  Változó
    i:Egész
```

```
  Db:=0
  Ciklus i=1-től N-ig
    j:=1
    Ciklus amíg j≤M és X(i)≠Y(j)
      j:+1
    Ciklus vége
    Ha j≤M akkor Db:+1; Z(Db):=X(i)
  Ciklus vége
Eljárás vége.
```

Megjegyzések:

1. Észre vehető, hogy a belső ciklus (és „környéke") egy Eldöntés tételt formál, míg a külső ciklus (és környéke) egy Kiválogatás tételt.
2. Gondolja ezt meg úgy, hogy megpróbálja a fenti algoritmust a korábbiakban taglalt *tételfüggvény*es leírási móddal újrafogalmazni!
3. Miért nem lenne helyes a specifikáció kimeneti részében a „$Db \in \mathbb{N}$, $Z \in H^{Min(N,M)}$"?

2.4. Összefuttatás tétele

Összefuttatás$(H^*, H^* \mathcal{F}(H \times H, \mathbb{L})) : H^*$

Be: $N \in \mathbb{N}$, $X \in H^*$, $M \in \mathbb{N}$, $Y \in H^*$ ($\leq : H \times H \to \mathbb{L}$ *rendezés*)

Ki: $Db \in \mathbb{N}$, $Z \in H^*$

Ef: $N = Hossz(X) \wedge M = Hossz(Y) \wedge$

$HalmazF\ddot{o}lsorol\acute{a}s(X) \wedge HalmazF\ddot{o}lsorol\acute{a}s(Y) \wedge RendezettSorozat_{\leq}(X) \wedge RendezettSorozat_{\leq}(Y)$[6]

Uf: $Db = N + \sum_{i=1}^{M} \chi(y_i \notin X) \wedge Z \in H^{Db} \wedge HalmazF\ddot{o}lsorol\acute{a}s(Z) \wedge RendezettSorozat_{\leq}(Z)$

$\wedge\ \forall i \in [1..Db] : (z_i \in X \vee z_i \in Y)$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Összefuttatás(Konstans N:Egész,X:THk,M:Egész,Y:THk
                      Változó  Db:Egész,Z:THk):
  Változó
    i:Egész
  Db:=0
  i:=1; j:=1
  Ciklus amíg i≤N és j≤M
    Db:+1
    Elágazás
      X(i)<Y(j) esetén Z(Db):=X(i); i:+1
      X(i)=Y(j) esetén Z(Db):=X(i); i:+1; j:+1
      X(i)>Y(j) esetén Z(Db):=Y(j); j:+1
    Elágazás vége
  Ciklus vége
```

[6] A rendezettség „irányáról" feltesszük a **növekvőséget**.

```
  Ciklus amíg i≤N
    Db:+1; Z(Db):=X(i); i:+1
  Ciklus vége
  Ciklus amíg j≤M
    Db:+1; Z(Db):=Y(j); j:+1
  Ciklus vége
Eljárás vége.
```

Megjegyzések:

1. Észre vehető a specifikáció alapján, hogy ez nem más, **mint egyesítés rendezett sorozatokra.**
2. Az elágazás teljes eseményt ír le, így nyilván a 3. feltétel helyettesíthető az „`egyébként`” kulcs-szóval (amivel kevesebb vizsgálatot írunk elő).
3. Érdemes eltűnődni azon, hogy mennyi nyereséggel jár az „eredeti” egyesítéshez képest a rendezettség figyelembe vétele.

	Egyesítés		*Összefuttatás*	
Minimálisan	Max(N,M)	(M+1)*M/2[7]	Max(N,M)	Min(N,M)
Átlagosan	N+M/2	O(M*N)[8]	O(N+M)	2*Min(N,M)
Maximálisan	N+M	M*N	N+M	3*Min(N,M)
	Másolásszám	***Hasonlításszám***	***Másolásszám***	***Hasonlításszám***

A táblázat értelmezéséhez tudnia kell, hogy milyen feltételek húzódnak meg az egyes értékek mögött. Többször előfordul, hogy valójában nem ilyen „egyszerű” érték a pontos érték. Például függhet attól, hogy N, M közül melyik a nagyobb. Nos ilyenkor önkényes feltétellel éltem. De mivel? Gondolja meg, melyik értékhez milyen előfeltétel teljesülése szükséges!

4. Az előbbi megoldás „szépséghibája”, hogy két szemet bántó **másolással** kellett kiegészíteni, mert a „fő” ciklus idejekorán véget ért, s így nem tudta elvégezni a kitűzött feladatot. Ezt kiküszöbölhetjük, ha a sorozatok **végéhez illesztünk két azonos (eddig nem bennlevő) maximális értéket**: $+\infty$[9] (***Ef:*** *…* $\wedge$ $+\infty \notin X$ $\wedge$ $+\infty \notin Y$).

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Összefuttatás(Konstans N:Egész,X:THk,M:Egész,Y:THk
                      Változó  Db:Egész,Z:THk):
  Változó
    i:Egész
  N:+1; X(N):=+∞; M:+1; Y(M):=+∞;
  Db:=0
  i:=1; j:=1
```

[7] N=M esetén.
[8] Ordó függvény. L. a hatékonyságról szóló előadásban a 32. dián.
[9] Később bevezetjük a rendezett típusok esetére alkalmazható típusfüggvényt a hozzá tartozó legnagyobb érték jelölésre. Ez a TH típusra így néz ki: `Max'`TH.

```
  Ciklus amíg i<N vagy j<M
    Db:+1
    Elágazás
      X(i)<Y(j) esetén Z(Db):=X(i); i:+1
      X(i)=Y(j) esetén Z(Db):=X(i); i:+1; j:+1
      X(i)>Y(j) esetén Z(Db):=Y(j); j:+1
    Elágazás vége
  Ciklus vége
Eljárás vége.
```

TARTALOM: