PROGRAMOZÁSMÓDSZERTAN
2. ELŐADÁS'2004
(VÁZLAT)

1. SPECIFIKÁCIÓ

1.1. Alapvető matematikai jelölések

Lásd http://izzo.inf.elte.hu/szlavi honlapon a „Prtetel.doc"-ban (vagy a Prtetel.pdf-ben)!

H halmaz	• tetszőleges (véges) halmaz a szokásos halmaz műveletekkel és konstansokkal ($\in,\notin,\subset,\subseteq,\cup,\cap,\setminus,\emptyset$, Számosság)
$\mathbb{L}$	• *logikai értékek* halmaza: {Igaz,Hamis} a szokásos műveletekkel ($\neg,\wedge,\vee,\Leftrightarrow,\Rightarrow$), kvantorokkal ($\forall,\exists$) és individuumváltozókkal, halmazokkal [1]
$\mathbb{N}$	• *természetes számok* halmaza {0, 1, ...} a szokásos műveletekkel (+,-,*,Div,Mod)
$\mathbb{Z}$	• *egész számok* halmaza {.., -1, 0, 1, ...} a szokásos műveletekkel (+,-,*,Div,Mod)
$\mathbb{R}$	• *valós számok* halmaza a szokásos műveletekkel (+,-,*,/, ^)
$\mathbb{C}$	• *karakterek* halmaza {…, ' ',…, 'A',…,'z', …}
$\mathbb{S}$	• *szövegek* halmaza
H-Sorozat	• H-beli elem N-esek ($N\in\mathbb{N}$), amelyen alapműveletként értelmezzük az *i. elem* ($i\in\{1..N\}$) kiválasztását: s_i:= az S sorozat i. eleme, valamint az *eleme* relációt: $x\in S$, ha $\exists i$: $x=s_i$, és a *sorozathossz* függvényt, ekkor Hossz(S)=N
H^N halmaz $N\in\mathbb{N}$	• H-beli N-*elemű sorozatok* halmaza
H^* halmaz	• H-beli *véges sorozatok* halmaza, azaz $H^* = \cup_{(i=0..\infty)} H^i$
$()\in H^*$ *üres sorozat*	• S=() $\Leftrightarrow$ Hossz(S)
[x..y][2] *index-intervallum* $x,y\in\mathbb{N}$	• [x..y]:={x,x+1,...,y-1,y}
I[x..y] *index-sorozat*	• I[x..y]:=$(x,x+1,...,y-1,y)\in\mathbb{N}^{y-x}$, ha $y\geq x$ I[x..y]:=(), egyébként
$\mathcal{F}$(H,S) *függvényhalmaz*	• $\mathcal{F}$(H,S):={ f : f:H→S }

[1] Pl. „**∃i∈H: T(i)**" logikai kifejezésben az i – individuumváltozó, a H – az individuumhalmaz. A kifejezést így értjük: „**létezik olyan H-beli i elem, amelyre teljesül a T predikátum**".

[2] Helyenként ugyanilyen értelemben használjuk a [x,y] rövidebb jelölést. A balról/jobbról nyíltságra a hagyományos matematikai zárójeleket használjuk.

$\le_H \in \mathcal{F}(H\times H,\mathbb{L})$ rendezés *rendezési reláció (infix jelölésű bináris függ.)* *ha lehet: $\le_H$ helyett $\le$*	• ha $\forall a\in H$: $a\le a$ (reflexív), $\forall a,b,c\in H$: $a\le b$ és $b\le c \Rightarrow a\le c$ (tranzitív), $\forall a,b\in H$: $a\le b$ és $b\le a \Rightarrow a=b$ (antiszimmetrikus) $\forall a,b\in H$: $a\le b$ vagy $b\le a$ (teljes)
H halmaz *RendezettHalmaz$_{\le}$(H)*	• ha $\forall h,j\in H$: $h\le j$ vagy $h\ge j$ (ha H véges $\Rightarrow \exists$max, min eleme)
$S\in H^*$ sorozat *RendezettSorozat$_{\le}$(S)*	• ha RendezettHalmaz(H) és $\forall i\in[1..N]$: $s_i\le s_{i+1}$
$S\in H^*$ sorozat *HalmazFölsorolás(S)*	• ha $\forall i,j\in[1..N]$: $i\ne j \Rightarrow s_i\ne s_j$
X&Y *konkatenáció*ja X és Y sorozatoknak	• $X\&Y:=(x_1,\ldots,x_N,y_1,\ldots,y_M)$, ahol N=Hossz(X) és M=Hossz(Y)
$X\subseteq Y$ X *részsorozata* Y-nak	• $\forall y\in Y$: $\exists x\in X$: $y=x$ és $i,j\in[1..\text{Hossz}(Y)]$: $i<j$: $y_i,y_j\in Y \Rightarrow$ $\exists k,l\in[1..\text{Hossz}(X)]$: $k<l$: ($y_i=x_k$ és $y_j=x_l$)
R\r sorozat $r\in H$, $R\in H^N$: HalmazFölsorolás(R)	• ha $r\in R \Rightarrow R\backslash r\in H^{N-1}$ és $R=R_1\&(r)\&R_2 \Rightarrow R\backslash r=R_1\&R_2$ ha $r\notin R \Rightarrow R\backslash r=R$
$\mathcal{P}$(R) az $R\in H^N$ *Permutációhalmaza*	• $\mathcal{P}(R):=\{S\in H^N : N=1 \Rightarrow S=R,$ $N>1 \Rightarrow S=s\&S'$ és $s\in R$ és $S'\in\mathcal{P}(R\backslash s)\}$
X,Y a Z *felbontása* (X,Y,Z sorozatok)	• ha Z=X&Y
$T_i:H\to\mathbb{L}$ $i\in[1..K]$ H-*partícionálása*	• $\forall x\in H$: $\vee_{(i=1..K)} T_i(x)$=Igaz és ($T_i(x)\wedge T_j(x)$=Hamis ($i\ne j$))

Definíció: Az $F:H^*\to S^*$ feladat *elemenként feldolgozható*, $\forall Z\in H^*$ $\forall$ X,Y Z-felbontásra igaz, hogy F(X)&F(Y)=F(Z).

Lemma: Ha az $F:H^*\to S^*$ feladat elemenként feldolgozható, akkor $\exists f:H\to S$ függvény, hogy $f(h_i)=s_i$ $i\in[1..N]$.

1.2. Egy példa

Feladat:

Egy egyetem hallgatóinak adatai (név, szak, évfolyam) alapján adjuk meg, van-e első éves informatika szakos hallgató!

Specifikáció:

Bemenet: $N\in\mathbb{N}$, *Hallgatók*$\in$(*Név*$\times$*Szak*$\times$*Évfolyam*)*,
Név=$\mathbb{S}$, *Szak*={*Info, Mat, Fiz, …*[3]}, *Évfolyam*=$\mathbb{N}$

[3] A „…” jelentése: itt egy felsorolás lenne, amit most nem folytatunk (formálisan persze kellene, mivel ki nem található)! Figyelem: így csak egy szak rendelhető a hallgatóhoz! Nagy baj ez?

Kimenet: $Vane \in \mathbb{L}$

Előfeltétel: $Hossz(Hallgatók)=N \wedge$

$\forall i \in [1..N]: Hallgatók_i.Név \neq '' \wedge Hallgatók_i.Évfolyam \in [1..5]$

Utófeltétel: $Vane = \exists i \in [1..N]: Hallgatók_i.Szak=Info \wedge Hallgatók_i.Évfolyam=1$

1.3. Specifikáció és programmodell

Állapottér: (bemenő és kimenő, valamint belső) adatok absztrakt halmaza, amelynek egy-egy vetületét („dimenzióját”) alkotják a változók.

Feladat: egy binér reláció az állapottér (-kettős) felett. Értve ez alatt: a *bemeneti* és a hozzá rendelt *kimeneti* állapot kettősének halmazát. Az állapottér kezdetben (bemenetként) elfogadható elemeinek halmazát jelöli ki az *előfeltétel*. (Amelyek kielégítik az előfeltételt, azok tartoznak a tényleges bemeneti halmazba.) Az elérendő végállapotot az *utófeltétel* rögzíti az állapothalmazban (a bemeneti állapot függvényében).

Specifikáció: a feladat formális meghatározása.

A **program** egy kiszámítási sorozatot definiál, amely meghatározza, hogy a számítás milyen állapottér „útvonalon” haladva jusson el a végeredményhez. Vagyis a *program* mint függvény (programfüggvény) egy *állapottér-transzformátor*.

Feladat- és **programspecifikáció** kapcsolata: a programspecifikációt úgy kapjuk a feladatspecifikációból, hogy **gyengítjük az előfeltételt** („többet engedünk meg”) és **szigorítjuk az utófeltételt** („egyértelműsítve az esetlegesen meglévő nemegyértelműségeket”).

2. ALGORITMUS ÉS ADAT – ’STRUKTÚRASZERINTI FELDOLGOZÁS’ ELV

2.1. Az elv

Absztraktadat	Algoritmikus nyelvi adatfogalom	⇒	Algoritmus
Elemi	Skalár	⇒	Értékadás (függvénnyel), eljárás
Direktszorzat	Rekord	⇒	Rekord-mező szerinti utasítás-szekvencia
Iteráció	Sorozatfélék (pl. tömb)	⇒	Ciklus, rekurzió
Unió	Alternatív rekord	⇒	Elágazás az alternatíva feltételei szerint

A fenti példában: az N és a Vane elemi (skalár), a Hallgatók iteráció (pl. tömb), mégpedig a Név, a Szak és az Évfolyam halmazok direktszorzataként adódó rekord.

2.2. Hogyan használható az elv föl?

Az 1.2.-beli példa „vezérlő” (azaz legbonyolultabb lévén, a legtöbb „gondot okozó”) adatszerkezete a *Hallgatók*. Ez *iteráció*, tehát sejthető, hogy a feldolgozás legbonyolultabb része egy *ciklus* lesz.

3. A STRUKTURÁLT PROGRAMOZÁS ELVEI

3.1. Felülről-lefelé tervezés és lépésenkénti finomítás

A feladat **egészét** tekintjük először, s így bontjuk föl néhány (3-7) **részfeladat**ra. A részfeladatok **megengedett programstruktúrák** komponenseiként alkotják a feladat egészének megoldását. A dekomponáláskor a részfeladatokat a szokásos *elő- és utófeltételekkel specifikáljuk*. (Ezek az elő- és utófeltételek természetesen „kellően” illeszkednek egymáshoz.)
Az egyes részfeladatokat külön-külön oldjuk meg az előbbi szerint.

3.2. Megengedett programstruktúrák

Transzformáció (értékadás jobboldalán függvénnyel; eljárás)
Elágazások (`Ha…akkor...különben…`; `Ha…akkor…`)
Ciklus (`Ciklus amíg …`)

3.3. Példa

Az 1.2.-beli példa lebontásának vázlata:

```
Program = Beolvasás(N,Hallgatók)
          Vane:=Számítás(N,Hallgatók)
          Kiírás(Vane)
Beolvasás(…) =...
Számítás(…) =
          Ciklus … N-re
            HallgatóFeldolgozás(Hallgatók(i))
          Ciklus vége
Kiírás(…) =...
```

4. ALGORITMIKUS NYELV RÉSZLETEI

4.1. A program legfelsőbb szintje

```
Program PrNév:
  Globális adatok definíciója, deklarációja
  (Típusdefiníciók, konstans- és változó-deklarációk)
  A végrehajtás legfelsőbb szintje
Program vége.[4]
```

Természetesen a végrehajtórészben szerepelnek azoknak az eljárásoknak és függvényeknek a hívásai, amelyeket a lépésenkénti finomítás során a tervezéskor kaptunk.

4.2. Az alprogramok definiálása

A felülről-lefelé haladva finomodó tevékenységek definiálása:

[4] Az algoritmus (és majd a kód) szerkezetet meghatározó kulcs-szavait **vastagítva** kiemeljük.

```
Eljárás EljNév(form.param.):
  lokális adatok definíciója, deklarációja
  A végrehajtás utasításai
Eljárás vége.
Függvény FvNév(form.param.): ÉrtékTípus
  lokális adatok definíciója, deklarációja
  A végrehajtás utasításai
  FvNév:=kifejezés
Függvény vége.
```

4.3. Példa

Be: $N \in \mathbb{N}$, *Hallgatók* $\in$ *(Név×Szak×Évfolyam)**, *Név*=$\mathbb{S}$, *Szak={Info,Mat,Fiz,…}*, *Évfolyam*=$\mathbb{N}$
Ki: *Vane* $\in \mathbb{L}$

```
Program Egyetem:
  Típus
    TNév=Szöveg
    TSzak=(Info, Mat, Fiz, …)
    TÉvfolyam=Egész
    THallgató=Rekord(
                  Név:TNév
                  Szak:TSzak
                  Évfolyam:TÉvfolyam)
  Konstans
    MaxN:Egész(1000)
  Típus
    THallgatók=Tömb(1..MaxN:THallgató)
  Változó
    N:Egész
    Hallgató:THallgatók
    Vane:Logikai
  Beolvasás(N,Hallgatók)
  Vane:=Feldolgozás(N,Hallgatók)
  Kiírás(Vane)
Program vége.
Eljárás Beolvasás(Változó n:Egész, h:THallgatók):
  Be: n [0≤n≤MaxN]
  Be: h(1..N) [h(1..N).Név≠'' és
               h(1..N).Évfolyam∈[1..5]]
Eljárás vége.
```

Ef: *Hossz(Hallgatók)=N* $\wedge$ $\forall i \in [1..N]$: *Hallgatók$_i$.Név≠''* $\wedge$ *Hallgatók$_i$.Évfolyam* $\in$ *[1..5]*

```
Függvény Feldolgozás(Konstans n:Egész
                              h:THallgatók):Logikai
  Változó
    i:Egész
  i:=1
  Ciklus amíg i≤N és
          nem (h(i).Szak=Info és h(i).Évfolyam=1)
    i:+1
  Ciklus vége
  Feldolgozás:=i≤N
Függvény vége.
```

Uf: *Vane*=$\exists i \in [1..N]$: *Hallgatók$_i$.Szak=Info* $\wedge$ *Hallgatók$_i$.Évfolyam=1*

```
Eljárás Kiírás(Konstans v:Logikai):
  Ki: v
Eljárás vége.
```

Megjegyzések:

- Érdemes megfigyelni a specifikáció és az algoritmus formális kapcsolatait, továbbá az alkalmazott névkonvenciókat.
- Mivel a specifikációban felbukkanó adatok a „főprogram” minden résztevékenysége számára elérhető (ún. globális adatok), ezért eltekinthetünk a résztevékenységeket megvalósító eljárások, ill. függvények paraméterezésétől.
- Bevezettük a programozási nyelvekben is gyakorta meglevő „inkrementálás” (növelés) műveletet. Értelmezését az alábbi: példa érzékelteti:
  `x:+y` ugyanaz, mintha `x:=x+y`
  (Persze kitalálható a jelentése a dekrement és társai műveleteknek: `x:-y, x:*y, x:/x`.)

5. PROGRAMOZÁSI TÉTELEK

5.1. A céljuk, lényegük

Gondolati mankóul szolgál a program algoritmusának felderítésében és formalizálásában. (A ***hatékonyság*** egyelőre ***nem*** szempont!)

Tételek szerkezete:

- **Feladat/program-specifikáció**
  1) Bemenet
  2) Kimenet
  3) Előfeltétel
  4) Utófeltétel
  5) Definíciók (opcionális)
- **Absztrakt algoritmus**
- **Állítás: *a specifikációt az absztrakt algoritmus kielégíti.***
- ***Bizonyítás***

5.2. A formalizmus és célja

Megállapodások:

- A tételeket mint *függvény*eket is megadjuk (a „fejsorban”, az ún. szignatúrában), amelynek argumentuma tartalmazza azokat a „kellékeket”, amelyek a szükséges bemenetei, ill. az értékét, amely halmazba képez. E jelölés célja, hogy pusztán ilyen „szintaktikai” természetű dolog is segítségünkre legyen majd a tételek összeépítésénél.
- Szívesebben adunk meg egy sorozatot H^*-beliként (H^N helyett), mivel a specifikációban legkevésbé így köti meg a kezünket (nem utal a leírás tömbre vagy más, fixhosszúságú

szerkezetre). A „tétel-függvényben” éppen e megfontolásból nem tüntetjük föl a sorozatok elemszámát szimbolizáló paramétereket.

- Az absztrakt algoritmust komplett eljárásként fogalmazzuk meg, amelyet megelőz a szükséges típusok definíciója.

5.3. A programozási tételek

5.3.1. Másolás tétel

Másolás$(H^*,\mathcal{F}(H,S)):S^*$

Be: $N \in N,\ X \in H^*$
Ki: $Y \in S^*$
Ef: F *elemenként feldolgozható az* $f{:}H \to S$ *függvénnyel* $\wedge$ $Hossz(X)=N$
Uf: $\forall i \in [1..N]{:}\ y_i = f(x_i)$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
      TSk=Tömb(1..MaxN:TS)
Eljárás Másolás(Konstans N:Egész, X:THk
                Változó  Y:TSk):
  Változó
    i:Egész
  Ciklus i=1-től N-ig
    Y(i):=f(X(i))
  Ciklus vége
Eljárás vége.
```

5.3.2. Sorozatszámítás tétel

Sorozatszámítás$(H^*,\mathcal{F}(H^*,H)):H$ lehetne még **Sorozatszámítás**$(H^*,H,\mathcal{F}(H,S)):S$

Be: $N \in \mathbb{N},\ X \in H^*,\ F{:}H^* \to H$
Ki: $S \in H$
Ef: $Hossz(X)=N \wedge \exists f{:}HxH \to H \wedge f_0 \in H\ :\ F((x_1..x_N))=f(F(x_2..x_N),x_1,),\ F(())=f_0 \wedge f_0$ *f-neutrális*[5]
Uf: $S=F(X)$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Sorozatszámítás(Konstans N:Egész, X:THk
                        Változó  S:TH):
  Változó
    i:Egész
```

[5] Azaz $\forall x \in H{:}\ f(f_0,x)=f(x,f_0)=x$

```
  S:=f0
  Ciklus i=1-től N-ig
  S:=f(S,X(i))
  Ciklus vége
Eljárás vége.
```

5.3.3. Eldöntés tétel

Eldöntés(H*,ℱ(H,𝕃)):𝕃

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

Ki: $Vane\in\mathbb{L}$

Ef: $Hossz(X)=N$

Uf: $Vane\equiv\exists i\in[1..N] : T(x_i)$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Eldöntés(Konstans N:Egész, X:THk
                 Változó  Vane:Logikai):
  Változó
    i:Egész
  i:=1
  Ciklus amíg i≤N és nem T(X(i))
    i:+1
  Ciklus vége
  Vane:=i≤N
Eljárás vége.
```

Megjegyzés: Hogyan származtatható e tételből az, amely utófeltétele az alábbi:

Uf: $Minde\equiv\forall i\in[1..N] : T(x_i)$

azaz, hogy „minden X-elem T-tulajdonságú-e”?

Válasz:

Uf: $Minde\equiv\neg\exists i\in[1..N] :\neg T(x_i)$

És algoritmikus folyamánya:

```
…
Eljárás Eldöntés(Konstans N:Egész, X:THk
                 Változó  Minde:Logikai):
  Változó
    i:Egész
  i:=1
  Ciklus amíg i≤N és ~~nem~~ T(X(i))
    i:+1
  Ciklus vége
  Minde:=i>N
Eljárás vége.
```

5.3.4. Kiválasztás tétel

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

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

Ki: $Sorsz \in \mathbb{N}$

Ef: $Hossz(X)=N \wedge \exists i \in [1..N] : T(x_i)$

Uf: $Sorsz \in [1..N] \wedge T(x_{Sorsz})$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás Kiválasztás(Konstans N:Egész, X:THk
                    Változó  Sorsz:Egész):
  Változó
    i:Egész
  i:=1
  Ciklus amíg nem T(X(i))
    i:+1
  Ciklus vége
  Sorsz:=i
Eljárás vége.
```

5.3.5. Keresés tétel

(Lineáris) keresés$(H^* \mathcal{F}(H,\mathbb{L}))$:$\mathbb{L} \cup \mathbb{L} \times \mathbb{N}$

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

Ki: $Vane \in \mathbb{L}, Sorsz \in \mathbb{N}$

Ef: $Hossz(X)=N$

Uf: $Vane \equiv \exists i \in [1..N] : T(x_i) \wedge Vane \Rightarrow Sorsz \in [1..N] \wedge T(x_{Sorsz})$

Alg:

```
Konstans MaxN:Egész(???)
Típus THk=Tömb(1..MaxN:TH)
Eljárás LinKeresés(Konstans N:Egész, X:THk
                   Változó  Vane:Logikai, Sorsz:Egész):
  Változó
    i:Egész
  i:=1
  Ciklus amíg i≤N és nem T(X(i))
    i:+1
  Ciklus vége
  Vane:=i≤N
  Ha Vane akkor Sorsz:=i
Eljárás vége.
```

6. SPECIFIKÁCIÓ – ALGORITMUS –KÓD

A specifikáció kapcsolata az algoritmussal, és a kóddal kérdéskörhöz érdemes a http://izzo.inf.elte.hu/szlavi honlapon a HTML-publikációk 2. hivatkozását (Specification, algorithm and program code.) fellapozni.

TARTALOM