Lista típuskonstrukció

Szlávi Péter
ELTE Informatika Szakmódszertani Csoport
szlavi@ludens.elte.hu
http://izzo.inf.elte.hu/~szlavi

Tartalomjegyzék

1 A lista algebrai specifikációja

1.1 Lista-műveletek

Típus Lista(Elem):

Asszociált műveletek:

Üres: Lista
Üres?(Lista): Logikai
ElemÉrték(Lista): Elem ∪ {NemDef}
ElemMódosít(Lista,Elem): Lista ∪ {NemDef}
Elsőre(Lista): Lista ∪ {NemDef}
Következőre(Lista): Lista ∪ {NemDef}
BeszúrMögé(Lista,Elem): Lista ∪ {NemDef}
BeszúrElejére(Lista,Elem): Lista ∪ {NemDef}

1.1 Lista-műveletek (folytatás)

Kihagy(Lista): Lista ∪ {NemDef}
UtolsóE?(Lista): Logikai ∪ {NemDef}

Kiegészítő műveletek:

BeszúrElé(Lista,Elem): Lista ∪ {NemDef}
ElemSzám(Lista): Egész

A kétirányú lista további műveletei:

Előzőre(Lista): Lista ∪ {NemDef}
Utolsóra(Lista): Lista ∪ {NemDef}
BeszúrElé(Lista,Elem): Lista ∪ {NemDef}
ElsőE?(Lista): Logikai ∪ {NemDef}

1.1 Lista-műveletek (folytatás)

A ciklikus lista:

olyan *egy-* vagy *kétirányú lista*, amelynek az utolsó elemét az első követi, az elsőt pedig az utolsó előzi meg. Így kevesebb műveletre van szükség a megvalósításához.

A gyűrű:

olyan *ciklikus lista*, amelynek nincs kitüntetett első eleme.

1.1 Lista-műveletek (folytatás)

További lehetséges műveletek:

Első(Lista): (Lista × Elem) U {NemDef}
Következő(Lista): (Lista × Elem) U {NemDef}
Utolsó(Lista): (Lista × Elem) U {NemDef}
Előző(Lista): (Lista × Elem) U {NemDef}
Egymásután(Lista,Lista): Lista

Axiómák

l,l',l'':Lista(Elem)
e,f:Elem
h,h':Egész

1.2 Lista-axiómák

1° Az Üres lista üres.
l=Üres ⇒ Üres?(l)

2° Üres listának nincs eleme.
Üres?(l) ⇔ ElemSzám(l)=0

3° Az üres listának nincs aktuális, első, ... utolsó eleme.
ElemÉrték(Üres)=NemDef ∧
ElsőE?(Üres)=NemDef ∧ Első(Üres)=NemDef ∧
Következő(Üres)=NemDef ∧
Előző(Üres)=NemDef ∧ Utolsó(Üres)=NemDef ∧
UtolsóE?(Üres)=NemDef

4a° Nincs elsőt megelőző és nincs utolsót követő eleme a listának.
Előző(Elsőre(l))=NemDef ∧
Következő(Utolsóra(l))=NemDef

1.2 Lista-axiómák (folytatás)

4b° A ciklikus listában az elsőt megelőző az utolsó, az utolsót követő pedig az első.
Előző(Elsőre(l))=Utolsó(l) ∧
Következő(Utolsóra(l))=Első(l)

5° A lista Utolsóra művelettel kiválasztott eleme az ElemSzám-adik, az 1. elemét az Elsőre művelet választja ki.
UtolsóE?(Utolsóra(l)) ∧ ElsőE?(Elsőre(l)) ∧
h=ElemSzám(l)-1 ⇒
Következőre^h(Elsőre(l))=Utolsóra(l) ∧
0≤h<ElemSzám(l)-1 ⇒
nem UtolsóE?(Következőre^h(Elsőre(l))) ∧
0<h≤ElemSzám(l)-1 ⇒
nem ElsőE?(Következőre^h(Elsőre(l)))

1.2 Lista-axiómák (folytatás)

6° A BeszúrMögé művelet bővíti a listát az adott elemmel. Az új elem az 'aktuális' mögé kerül. Üres listába elsőként kerül be az új elem. Az új válik aktuálissá. Ha nincs 'aktuális' elem, akkor a művelet eredménye nem definiált.

$f=ElemÉrték(l) \Rightarrow \exists l'=BeszúrMögé(l,e):$
$ElemSzám(l')=ElemSzám(l)+1 \wedge$
$ElemÉrték(l')=e \wedge Előző(l').\underline{Elem}=f \wedge$
$Üres?(l) \Rightarrow ElemSzám(BeszúrMögé(l,e))=1 \wedge$
$ElemÉrték(BeszúrMögé(l,e))=e \wedge$
$nem\ Üres?(l) \wedge ElemÉrték(l)=NemDef \Rightarrow$
$BeszúrMögé(l,e)=NemDef$

A BeszúrElé művelet ennek analógiájára fogalmazható meg.

1.2 Lista-axiómák (folytatás)

7° Kihagyni az 'aktuális' elemet lehet, az 'aktuális' a következő lesz. Ha nincs az 'aktuális' elem kijelölve, akkor nem lehet kihagyni sem.

$\exists ElemÉrték(l) \Rightarrow \exists l'=Kihagy(l):$
$ElemSzám(l')=ElemSzám(l)-1 \wedge$
$(Üres?(l') \vee ElemÉrték(l')=Következő(l).Elem)$
$ElemÉrték(l)=NemDef \Rightarrow Kihagy(l)=NemDef$

Állítás: ha az utolsó az 'aktuális', akkor a kihagyás után nem definiált lesz az 'aktuális'.

$UtolsóE?(l) \Rightarrow ElemÉrték(Kihagy(l))=NemDef$

Biz.:

$UtolsóE?(l) \Rightarrow \exists ElemÉrték(l)$
$7° \Rightarrow \exists l'=Kihagy(l):(Üres?(l') \vee$
$ElemÉrték(l')=Következő(l).Elem)$
(1) $Üres?(l') \wedge 3° \Rightarrow ElemÉrték(Kihagy(l))=NemDef$
(2) $ElemÉrték(l')=Következő(l).Elem \wedge 9° \wedge 4a° \Rightarrow$
$ElemÉrték(l')=NemDef$

1.2 Lista-axiómák *(folytatás)*

8° Két egymásután illesztett lista listát alkot. Elemei az eredeti kettő elemei lesznek, az eredeti sorrendben.

$h=ElemSzám(l) \wedge h'=ElemSzám(l') \wedge l''=Egymásután(l,l') \Rightarrow$

$\forall k \in [0,h): ElemÉrték(Következőre^k(Elsőre(l'')))=$

$=ElemÉrték(Következőre^k(Elsőre(l))) \wedge$

$\forall k \in [0,h'): ElemÉrték(Következőre^{k+h}(Elsőre(l')))=$

$=ElemÉrték(Következőre^k(Elsőre(l')))$

1.2 Lista-axiómák *(folytatás)*

9° A következő elem az, amelyet a következőre lépés után aktuálisként érzékelünk, az előző az, amelyet az előzőre lépés után aktuálisként érzékelünk, ...

$ElemÉrték(l) \neq NemDef \Rightarrow$

$Következő(l).Elem = ElemÉrték(Következőre(l)) \wedge$

$Előző(l).Elem = ElemÉrték(Előzőre(l)) \wedge$

$Első(l).Elem = ElemÉrték(Elsőre(l)) \wedge$

$Utolsó(l).Elem = ElemÉrték(Utolsóra(l))$

2 A lista típuskonstrukció specifikációja

2.1 A lista exportmodulja

Meg kell gondolni az operátorok ef/uf-ét az axiómák alapján!

ExportModul Lista(**Típus** TElem):

Eljárás Üres(**Változó** l:Lista)

Függvény Üres?(**Konstans** l:Lista): Logikai

Függvény ElemÉrték(**Változó** l:Lista): TElem

Eljárás ElemMódosít(**Változó** l:Lista, **Konstans** e:TElem)

Eljárás Elsőre(**Változó** l:Lista)

Eljárás Következőre(**Változó** l:Lista)

2.1 A lista exportmodulja *(folytatás)*

Eljárás BeszúrMögé(**Változó** l:Lista, **Konstans** e:TElem)

Eljárás BeszúrElejére(**Változó** l:Lista, **Konstans** e:TElem)

Eljárás Kihagy(**Változó** l:Lista)

Függvény UtolsóE?(**Változó** l:Lista): Logikai

Függvény ElemSzám(**Konstans** l:Lista): Egész

Eljárás BeszúrElé(**Változó** l:Lista, **Konstans** e:TElem)

Függvény Hibás?(**Változó** l:Lista): Logikai

2.1 A lista exportmodulja *(folytatás)*

Infix Operátor Azonos?(**Konstans** l1,l2:Lista):Logikai
Másnéven l1=l2

Infix Operátor LegyenEgyenlő(**Változó** l1:Lista,
Konstans l2:Lista)
Másnéven l1:=l2

Operátor Ki(**Konstans** l:Lista)
Másnéven Ki: l

Operátor Be(**Változó** l:Lista)
Másnéven Be: l

Modul vége.

2.1 A lista exportmodulja *(folytatás)*

A kétirányú lista további műveletei:

Eljárás Utolsóra(**Változó** l:Lista)

Eljárás Előzőre(**Változó** l:Lista)

Függvény ElsőE?(**Változó** l:Lista): Logikai

2 A lista típuskonstrukció specifikációja

2.2. A lista megvalósítási moduljai

2.2.1 Láncolt ábrázolás

```
Modul Lista(Típus TElem):
    Reprezentáció
        Típus ListaElem=Rekord(érték: TElem
                               köv: ListaElem'Mutató)
        Változó  fej,akt: ListaElem'Mutató
                 hossz: Egész
                 hiba: Logikai
```


2.2.1 Láncolt ábrázolás (folytatás)

```
Implementáció
    Eljárás Üres(Változó l:Lista):
        fej:=sehova; akt:=sehova; hossz:=0
        hiba:=Hamis
    Eljárás vége.
    Függvény Üres?(Konstans l:Lista): Logikai
        Üres?:=hossz=0
    Függvény vége.
    Függvény ElemSzám(Konstans l: Lista): Egész
        ElemSzám:=hossz
    Függvény vége.
```

2.2.1 Láncolt ábrázolás (folytatás)

```
Függvény ElemÉrték(Változó l:Lista): TElem
  Ha akt≠sehova
     akkor   ElemÉrték:=ListaElem(akt).érték
   különben hiba:=Igaz
Függvény vége.
Eljárás Elsőre(Változó l:Lista):
  Ha Üres(l) akkor hiba:=Igaz
  akt:=fej
Eljárás vége.
Eljárás Következőre(Változó l:Lista):
  Ha akt≠sehova akkor  akt:=ListaElem(akt).köv
                 különben hiba:=Igaz
Eljárás vége.
```

A függvény értéke NemDef!

2.2.1 Láncolt ábrázolás (folytatás)

```
Eljárás BeszúrMögé(Változó l:Lista,
                   Konstans e:TElem):
  Változó  új : ListaElem'Mutató
  Elágazás
    fej=sehova esetén BeszúrElejére(l,e)
    akt≠sehova  esetén Lefoglal(új, ListaElem(e,
                       ListaElem(akt).köv))
                  hiba:=új=sehova
                  Ha nem hiba akkor
                    ListaElem(akt).köv:=új
                    akt:=új; hossz:+1
                  Elágazás vége
       egyéb esetben   hiba:=Igaz
  Elágazás vége
Eljárás vége.
```

2.2.1 Láncolt ábrázolás (folytatás)

```
Eljárás BeszúrElejére(Változó l:Lista,
                      Konstans e:TElem):
  Változó  új: ListaElem'Mutató
  új:=fej; Lefoglal(fej,ListaElem(e,új))
  hiba:=fej=sehova
  Ha nem hiba akkor   akt:=fej; hossz:+1
                 különben fej:=új
Eljárás vége.
Eljárás Kihagy(Változó l:Lista):
  Változó el: ListaElem'Mutató
  Ha akt≠sehova akkor
    Ha akt=fej akkor [első elem]
      fej:=Listaelem(akt).köv
      Felszabadít(akt); akt:=fej
```


2.2.1 Láncolt ábrázolás (folytatás)

```
    különben  [nem az első elem]
  [előző elem megkeresése:]
  el:=fej
  Ciklus amíg ListaElem(el).köv≠akt
      el:=ListaElem(el).köv
  Ciklus vége
  ListaElem(el).köv:=ListaElem(akt).köv
  Felszabadít(akt); akt:=ListaElem(el).köv
 Elágazás vége
 hossz:-1
    különben [nincs aktuális elem]
 hiba:=Igaz
Elágazás vége
Eljárás vége.
```

2.2.1 Láncolt ábrázolás (folytatás)

```
Függvény UtolsóE?(Változó l:Lista): Logikai
  Ha akt≠sehova
     akkor   UtolsóE?:=ListaElem(akt).köv=sehova
   különben hiba:=Igaz
Függvény vége.
Függvény Hibás?(Változó l:Lista): Logikai
   Hibás?:=hiba; hiba:=hamis
Függvény vége.
Infix Operátor Azonos?(Konstans l1,l2:Lista):Logikai
   Másnéven l1=l2
          …
Operátor vége.
```


2.2.1 Láncolt ábrázolás (folytatás)

```
Infix Operátor LegyenEgyenlő(Változó l1:Lista,
                             Konstans l2:Lista):
   Másnéven l1:=l2
          …
Operátor vége.
Operátor Ki(Konstans l:Lista):
  Másnéven Ki: l
          …
Operátor vége.
Operátor Be(Változó l:Lista):
  Másnéven Be: l
          …
Operátor vége.
Inicializálás [minden lista-deklarációkor végrehajtandó]
  fej:=sehova; akt:=sehova; hossz:=0; hiba:=Hamis
Modul vége.
```

2 A lista típuskonstrukció specifikációja

2.2. A lista meglósítási moduljai

2.2.2 Folytonos ábrázolás

```
Modul Lista(Típus TElem):
    Reprezentáció
        Típus ListaElemek=Tömb(1..MaxSzám: TElem)
        Változó  le: ListaElemek
                 akt,hossz: 0..MaxSzám+1
                 hiba: Logikai
```

Kényelmességből, l. Következőre, Előzőre

2.2.2 Folytonos ábrázolás (folytatás)

```
Implementáció
    Eljárás Üres(Változó l:Lista):
      akt:=0; hossz:=0; hiba:=Hamis
    Eljárás vége.
    Függvény Üres?(Konstans l:Lista): Logikai
      Üres?:=hossz=0
    Függvény vége.
    Függvény ElemSzám(Konstans l: Lista): Egész
      ElemSzám:=hossz
    Függvény vége.
    Függvény ElemÉrték(Változó l:Lista): TElem
      Ha akt∈[1..hossz] akkor  ElemÉrték:=le(akt)
                       különben hiba:=Igaz
    Függvény vége.
```

A függvény értéke NemDef!

2.2.2 Folytonos ábrázolás (folytatás)

```
Eljárás Elsőre(Változó l:Lista):
  Ha hossz=0 akkor hiba:=Igaz
  akt:=1
Eljárás vége.
Eljárás Következőre(Változó l:Lista):
  Ha akt∈[1..hossz] akkor   akt:+1
                    különben hiba:=Igaz
Eljárás vége.
Eljárás Előzőre(Változó l:Lista):
  Ha akt∈[1..hossz] akkor   akt:-1
                    különben hiba:=Igaz
Eljárás vége.
```


2.2.2 Folytonos ábrázolás (folytatás)

```
Eljárás Utolsóra(Változó l:Lista):
  Ha hossz=0 akkor hiba:=Igaz
  akt:=hossz
Eljárás vége.
Eljárás BeszúrMögé(Változó l:Lista,
                   Konstans e:TElem):
  Változó i:1..MaxSzám
  Elágazás
    hossz=0 esetén
        le(1):=e; akt:=1; hossz:=1
   akt=hossz és hossz<MaxSzám esetén
        akt:+1; le(akt):=e; hossz:+1
  ...
```

2.2.2 Folytonos ábrázolás (folytatás)

```
            ...
  akt∈[1..hossz-1] és hossz<MaxSzám esetén
            akt:+1
            Ciklus i=hossz-tól akt-ig -1-esével
              le(i+1):=le(i)
            Ciklus vége
            le(akt):=e; hossz:+1
   egyéb esetben hiba:=Igaz
  Elágazás vége
Eljárás vége.
```


2.2.2 Folytonos ábrázolás (folytatás)

```
Eljárás Kihagy(Változó l:Lista):
  Változó i :1..MaxSzám
  Ha akt∈[1..hossz] akkor
                  Ciklus i=akt-tól hossz-1-ig
                    le(i):=le(i+1)
                  Ciklus vége
                  hossz:-1
              különben
                  hiba:=Igaz
  Elágazás vége
Eljárás vége.
Függvény UtolsóE?(Változó l:Lista): Logikai
  Ha akt∈[1..hossz] akkor   UtolsóE?:=akt=hossz
                    különben hiba:=Igaz
Függvény vége.
```

2.2.2 Folytonos ábrázolás (folytatás)

```
Függvény ElsőE?(Változó l:Lista): Logikai
  Ha akt∈[1..hossz] akkor   ElsőE?:=akt=1
                    különben hiba:=Igaz
Függvény vége.
Függvény Hibás?(Változó l:Lista): Logikai
  Hibás?:=hiba; hiba:=hamis
Függvény vége.
Infix Operátor Azonos?(Konstans  l1,l2:Lista):Logikai
  Másnéven l1=l2
        …
Operátor vége.
```

A függvény értéke NemDef!

2.2.2 Folytonos ábrázolás (folytatás)

```
Infix Operátor LegyenEgyenlő(Változó l1:Lista,
                             Konstans l2:Lista):
  Másnéven l1:=l2
        …
Operátor vége.
Operátor Ki(Konstans l:Lista):
  Másnéven Ki: l
        …
Operátor vége.
Operátor Be(Változó l:Lista):
  Másnéven Be: l
        …
Operátor vége.
Inicializálás
    akt:=0; hossz:=0; hiba:=Hamis
Modul vége.
```

3 Alkalmazási példák

3.1 Kiválogatás tétel újrafogalmazása listára

3.2 Összefuttatás tétel újrafogalmazása listára

3.3 Beszúrásos rendezés

3.1 Kiválogatás

Típus TLista=Lista(TElem) [a rövidség kedvéért]
Függvény Ttul(Konstans e:TElem): Logikai
…egy TElem típusú e adatra vonatkozó predikátum…
Függvény vége.

Eljárás Kiválogatás(**Konstans** l:TLista, **Változó** t1:TLista):
Üres(tl); Elsőre(l) [a tl-lel „szinkron” számláló nem kell!]
Ciklus amíg nem UtolsóE?(l)
Ha Ttul(ElemÉrték(l)) **akkor**
BeszúrMögé(tl,ElemÉrték(l))
Elágazás vége
Következőre(l)
Ciklus vége
Ha Ttul(ElemÉrték(l)) **akkor**
BeszúrMögé(tl,ElemÉrték(l))
Eljárás vége.

Elemfeldolgozás az utolsóig.

Az utolsó elem feldolgozása

3.2 Összefuttatás

Részben megvalósítási szinten (láncoltan) fogalmazzuk meg; x és y a végén törlődik!

```
Eljárás Összefuttatás(Változó x,y,z:TLista):
  Üres(z); Elejére(x); Elejére(y)
  Ha ElemÉrték(x)≤ElemÉrték(y) akkor
     z.fej:=x.fej; z.akt:=z.fej; Következőre(x)
        különben
     z.fej:=y.fej; z.akt:=z.fej; Következőre(y)
  Elágazás vége
  Ciklus amíg x.akt≠sehova és y.akt≠sehova
   Elágazás
     ElemÉrték(x)<ElemÉrték(y) esetén
        ListaElem(z.akt).köv:=x.akt; z.akt:=x.akt
        Következőre(x)
```


3.2 Összefuttatás (folytatás)

```
        ...
     ElemÉrték(x)=ElemÉrték(y) esetén
        ListaElem(z.akt).köv:=x.akt; z.akt:=x.akt
        Következőre(x); Következőre(y)
     ElemÉrték(x)>ElemÉrték(y) esetén
        ListaElem(z.akt).köv:=y.akt; z.akt:=y.akt
        Következőre(y)
   Elágazás vége
  Ciklus vége
  Ha x.akt≠sehova akkor ListaElem(z.akt).köv:=x.akt
  Ha y.akt≠sehova akkor ListaElem(z.akt).köv:=y.akt
  Üres(x); Üres(y)
Eljárás vége.
```

3.3 Beszúrásos rendezés

```
Típus  TRendezendő=Tömb(1..Max:TElem)
       TRendezett=Lista(TElem) [növekvően]
Eljárás Rendezés(Konstans t:TRendezendő,
                 Változó l:TRendezett):
  Változó i:Egész
  Üres(l)
  Ciklus i=1-től N-ig
    Elsőre(l); e:=t(i)
    Ciklus amíg nem UtolsóE?(l) és e>ElemÉrték(l)
      Következőre(l)
    Ciklus vége
    Ha e>ElemÉrték(l) akkor BeszúrMögé(l,e)
                      különben BeszúrElé(l,e)
  Ciklus vége
Eljárás vége.
```

Programozásmódszertan 37 2005.03.08.

Megjegyzés -- Változó

A „Változó”-ság oka:

valamilyen hiba lehetősége fönnáll (üres a lista, vagy nincs kijelölt aktuális elem), s ennek visszajelzésére a „hiba” mező változhat.

Programozásmódszertan 38 2005.03.08.

Megjegyzés -- Lista x

Lista a függvény értékkészletében:

mivel a művelet elvégzése során, „bal kézről” olyan --a lista állapotát meghatározó-- jellemző (is) megváltozhat (pl. az *akt*- vagy a *hiba*-mező, amely első látásra nem várható.

Megjegyzés -- .Lista/.Elem

Az algebrai leírásban a nem aktuális elem értékét visszaadó függvények (Előző, Következő, Első, Utolsó) összetett értékének *elem*részére a '**.Elem**' mezőszelektorral hivatkozunk, a *lista*részére '**.Lista**'-val.