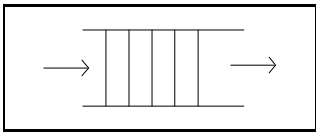
Sor típuskonstrukció



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

Tartalomjegyzék

1 A sor algebrai specifikációja

1.1 Sor-műveletek

Típus Sor(Elem):

Asszociált műveletek:

Üres: Sor
Üres?(Sor): Logikai
Tele?(Sor): Logikai
Első(Sor): Elem ∪ {NemDef}
Sorba(Sor,Elem): Sor ∪ {NemDef}
Sorból(Sor): (Sor × Elem) ∪ {NemDef}
SorHossz(Sor): Egész

1.2 Sor-axiómák

Axiómák:

1º Az Üres sor üres.
$s=Üres \Rightarrow Üres?(s) \wedge SorHossz(s)=0$

2º Az a sor, amelyben legalább egy elem van, az nem üres; a sorba tétel során a sor hossza eggyel nő.
$\neg Tele?(s) \Rightarrow \neg Üres?(Sorba(s,e)) \wedge SorHossz(Sorba(s,e))=SorHossz(s)+1$

3aº Az üres sornak nincs első eleme.
$Első(Üres)=NemDef$

3bº Az Üres sorból nem lehet kivenni elemet.
$Sorból(Üres)=NemDef$

3cº A tele sorba nem lehet további elemet betenni.
$Tele?(s) \Rightarrow Sorba(s,e)=NemDef$

1.2 Sor-axiómák (folytatás)

4º A sor első eleme a legrégebben betett elem.
Üres?(s) ∧ ¬ Tele?(s) ⇒ Első(Sorba(s,e))=e
¬ Üres?(s) ∧ ¬ Tele?(s) ⇒ Első(Sorba(s,e))=Első(s)

5º A sorból az első elemet lehet kivenni (a többi nem változik).
Üres?(s) ∧ ¬ Tele?(s) ⇒ Sorból(Sorba(s,e))=(s,e) ∧
¬ Üres?(s) ∧ ¬ Tele?(s) ⇒ Sorból(s)=(s',e) ∧
e=Első(s) ∧

Sorba(s',e')=Sorból(Sorba(s,e')).Sor

Állítás: a Sorból művelet után eggyel csökken a sor hossza.
¬ Üres?(s) ⇒ SorHossz(Sorból(s).Sor)=SorHossz(s)-1

Biz.:

2 A sor típuskonstrukció specifikációja

2.1 A sor exportmodulja

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

ExportModul Sor(**Típus** TElem):

Eljárás Üres(**Változó** s:Sor)

Függvény Üres?(**Konstans** s:Sor): Logikai

Függvény Tele?(**Konstans** s:Sor): Logikai

Függvény Első(**Változó** s:Sor): TElem

Eljárás Sorba(**Változó** s:Sor,
Konstans e:TElem)

Eljárás Sorból(**Változó** s:Sor, e:TElem)

Függvény SorHossz(**Konstans** s:Sor): Egész

2.1 A sor exportmodulja (folytatás)

Infix **Operátor** Azonos?(**Konstans** s1,s2:Sor):Logikai
Másnéven s1=s2

Infix **Operátor** LegyenEgyenlő(**Változó** s1:Sor,
Konstans s2:Sor)
Másnéven s1:=s2

Operátor Kiírás(**Konstans** s:Sor)
Másnéven Ki: s

Operátor Beolvasás(**Változó** s:Sor)
Másnéven Be: s

Függvény Hibás?(**Változó** s:Sor): Logikai

Modul vége.

2 A sor típuskonstrukció specifikációja

2.2. A sor megvalósítási moduljai

2.2.1 Láncolt ábrázolás

Modul Sor(**Típus** TElem):
Reprezentáció
Típus SorElem=**Rekord**(érték: TElem
köv: SorElem**'Mutató**)
Változó eleje,vége: SorElem**'Mutató**
hossz: Egész
hiba: Logikai

2.2.1 Láncolt ábrázolás (folytatás)

```
Implementáció
    Eljárás Üres(Változó s:Sor):
      eleje:=Sehova; vége:=Sehova; hossz:=0
      hiba:=Hamis
    Eljárás vége.
    Függvény Üres?(Konstans s:Sor): Logikai
      Üres?:=eleje=Sehova
    Függvény vége.
    Függvény Tele?(Konstans s:Sor): Logikai
      Változó ss: SorElem'Mutató
      Lefoglal(ss)
      Ha ss=Sehova akkor   Tele?:=Igaz
                   különben Tele?:=Hamis
                            Felszabadít(ss)
      Elágazás vége
    Függvény vége.
```


2.2.1 Láncolt ábrázolás (folytatás)

```
Függvény Első(Változó s:Sor): TElem
   Ha eleje≠Sehova akkor   Első:=SorElem(eleje).érték
                   különben hiba:=Igaz
Függvény vége.
Eljárás Sorba(Változó s:Sor, Konstans e:TElem):
  Változó új: SorElem'Mutató
  Lefoglal(új)
  Ha új≠Sehova akkor
        SorElem(új):=SorElem(e,Sehova)
        Ha vége≠Sehova akkor  SorElem(vége).köv:=új
                      különben eleje:=új
        vége:=új
        hossz:+1
```

2.2.1 Láncolt ábrázolás (folytatás)

```
  különben
        hiba:=Igaz
  Elágazás vége
Eljárás vége.

Eljárás Sorból(Változó s:Sor, e:TElem):
  Változó újeleje: SorElem'Mutató
  Ha eleje≠Sehova akkor
        e:=SorElem(eleje).érték
        újeleje:=SorElem(eleje).köv
        Felszabadít(eleje); eleje:=újeleje
        Ha eleje=Sehova akkor vége:=Sehova
        hossz:-1
  különben
        hiba:=Igaz
  Elágazás vége
Eljárás vége.
```


2.2.1 Láncolt ábrázolás (folytatás)

```
Függvény SorHossz(Konstans s:Sor): Egész
  SorHossz:=hossz
Függvény vége.
Függvény Hibás?(Változó s:Sor): Logikai
  Hibás?:=hiba; hiba:=Hamis
Függvény vége.
Infix Operátor Azonos?(Konstans s1, s2:Sor):Logikai
  Másnéven s1=s2
  ???
Operátor vége.
Infix Operátor LegyenEgyenlő(Változó s1:Sor,
                              Konstans s2:Sor)
  Másnéven s1:=s2
  ???
Operátor vége.
```

2.2.1 Láncolt ábrázolás (folytatás)

Operátor Kiírás(**Konstans** s:Sor)
Másnéven Ki: s
…
Operátor vége.
Operátor Beolvasás(**Változó** s:Sor)
Másnéven Be: s
…
Operátor vége.
Inicializálás
fej:=Sehova; vége:=Sehova; hossz:=0; hiba:=Hamis
Modul vége.

2 A sor típuskonstrukció specifikációja

2.2. A sor meglósítási moduljai

2.2.2 Folytonos ábrázolás

Modul Sor(**Típus** TElem):
Reprezentáció
Konstans MaxHossz: Egész(???)
Típus SorElemek=**Tömb**(1..MaxHossz: TElem)
Változó se: SorElemek
eleje, vége, hossz: 0..MaxHossz
hiba: Logikai

2.2.2 Folytonos ábrázolás (folytatás)

```
Eljárás Üres(Változó s:Sor):
  eleje:=1; vége:=1; hossz:=0; hiba:=Hamis
Eljárás vége.
Függvény Üres?(Konstans s:Sor): Logikai
  Üres?:=hossz=0
Függvény vége.
Függvény Tele?(Konstans s:Sor): Logikai
  Tele?:=hossz=MaxHossz
Függvény vége.
Függvény Első(Változó s:Sor): TElem
  Ha hossz≠0 akkor  Első:=se(eleje)
                különben hiba:=Igaz
Függvény vége.
```

Programozásmódszertan 15 2005.03.15.

2.2.2 Folytonos ábrázolás (folytatás)

```
Eljárás Sorba(Változó s:Sor, Konstans e:TElem):
  Ha hossz<MaxHossz akkor
        se(vége):=e; hossz:+1; vége:⊕1
  különben
        hiba:=Igaz
  Elágazás vége
Eljárás vége.

Eljárás Sorból(Változó s:Sor, e:TElem):
  Ha hossz>0 akkor
        e:=se(eleje); hossz:-1; eleje:=eleje⊕1
  különben
        hiba:=Igaz
  Elágazás vége
Eljárás vége.
```

Programozásmódszertan 16 2005.03.15.

2.2.2 Folytonos ábrázolás (folytatás)

```
Függvény SorHossz(Konstans s:Sor): Egész
  SorHossz:=hossz
Függvény vége.
Függvény Hibás?(Változó s:Sor): Logikai
  Hibás?:=hiba; hiba:=Hamis
Függvény vége.
Infix Operátor Azonos?(Konstans s1, s2:Sor):Logikai
  Másnéven s1=s2
  ...
Operátor vége.
Infix Operátor LegyenEgyenlő(Változó s1:Sor,
                              Konstans s2:Sor):
  Másnéven s1:=s2
  ...
Operátor vége.
```


2.2.2 Folytonos ábrázolás (folytatás)

```
Operátor Kiírás(Konstans s:Sor):
  Másnéven Ki: s
  …
Operátor vége.
Operátor Beolvasás(Változó s:Sor):
  Másnéven Be: s
  …
Operátor vége.
Inicializálás
   eleje:=1; vége:=1; hossz:=0; hiba:=Hamis
Modul vége.
```

3 Alkalmazási példák

3.1Input/Output puffer

3.2 Mindennapok sorai

3.3 Szimuláció

3.1 Input/Output puffer

Típus InputPuffer(Elem):

Asszociált műveletek:

Üres?(InputPuffer): Logikai
Teletölt(InputPuffer,Valami): InputPuffer ∪ {NemDef}
Pufferből(InputPuffer): (InputPuffer × Elem) ∪ {NemDef}

Típus OutputPuffer(Elem):

Asszociált műveletek:

Tele?(OutputPuffer): Logikai
Kiürít(OutputPuffer,Valami): OutputPuffer ∪ {NemDef}
Pufferbe(OutputPuffer,elem): (OutputPuffer) ∪ {NemDef}

3.1 Input/Output puffer

```
Modul InputPuffer(Típus TElem):
 Reprezentáció
    Konstans MaxHossz: Egész(???)
    Típus    PufferElemek=Tömb(1..MaxHossz: TElem)
    Változó pe    : PufferElemek
            eleje : 0..MaxHossz
            hiba  : Logikai
 Implementáció
  Függvény Üres?(Konstans p:InputPuffer): Logikai
   Üres?:=eleje>MaxHossz
  Függvény vége.
  Eljárás Teletölt(Változó p:InputPuffer, ???):
   ... [a pe puffer-tömb teletöltése elemekkel]
   eleje:=1
  Eljárás vége.
```

Programozásmódszertan 21 2005.03.15.

3.1 Input/Output puffer (folytatás)

```
  Eljárás Pufferből(Változó p:InputPuffer, e:TElem):
   Ha eleje≤MaxHossz akkor
            e:=pe(eleje); eleje:+1
      különben
            hiba:=Igaz [vagy Teletölt(p.???)]
   Elágazás vége
  Eljárás vége.
  Függvény Hibás?(Változó p:InputPuffer): Logikai
   Hibás?:=hiba; hiba:=Hamis
  Függvény vége.
 Inicializálás
  eleje:=MaxHossz+1; hiba:=Hamis
Modul vége.
```

Programozásmódszertan 22 2005.03.15.

3.1 Input/Output puffer (folytatás)

```
Modul OutputPuffer(Típus TElem):
 Reprezentáció
    Konstans MaxHossz: Egész(???)
    Típus    PufferElemek=Tömb(1..MaxHossz: TElem)
    Változó pe    : PufferElemek
            vége : 0..MaxHossz
            hiba  : Logikai
 Implementáció
  Függvény Tele?(Konstans p:OutputPuffer): Logikai
       Tele?:=vége≥MaxHossz
  Függvény vége.
  Eljárás Kiürít(Változó p:OutputPuffer, ???):
      ... [a pe puffer-tömbből az elemek kimásolása]
      vége:=0
  Eljárás vége.
```


3.1 Input/Output puffer (folytatás)

```
  Eljárás Pufferbe(Változó p:OutputPuffer,
                   Konstans e:TElem):
   Ha vége<MaxHossz akkor
            vége:+1;  pe(vége):=e
     különben
            hiba:=Igaz [vagy Kiürít(p.???)]
   Elágazás vége
  Eljárás vége.
  Függvény Hibás?(Változó p:OutputPuffer): Logikai
   Hibás?:=hiba; hiba:=Hamis
  Függvény vége.
 Inicializálás
  vége:=0; hiba:=Hamis
Modul vége.
```

3.2 Mindennapok sorai

Csak körül kell nézni az utcán, a bevásárló központokban, a moziban, a metróban …

3.3 Szimuláció

Párhuzamos folyamatok egyprocesszoros számítógépen… A probléma, amit meg kell oldani: az időbeliség fenntartása a folyamatok imitálása során.…

Megjegyzés -- Változó

A „Változó”-ság oka:

- valamilyen hiba lehetősége fönnáll (üres vagy tele a sor), s ennek visszajelzésére a „*hiba*” *mező* változhat.
- Vagy „direkt” beállítja az adott operáció (l. *Hiba? függvény*t).

Megjegyzés -- Tele?

Ez a furcsaság azt fejezi ki, hogy a memória akkor is elfogyhat, amikor az adott sor első elemét igyekeznénk beletenni.

Megjegyzés -- ciklikus növelés

A rövidség kedvéért bevezetjük *ciklikus növelés*, illetve *csökkentés* műveleteket.

$\oplus\mathbf{1}$:{1..MaxHossz}→{1..MaxHossz}

x$\oplus\mathbf{1}$:= (x **Mod** MaxHossz)+**1**,

$\Theta\mathbf{1}$:{1..MaxHossz}→{1..MaxHossz}

x$\Theta\mathbf{1}$:= (x**–1**+MaxHossz) **Mod** MaxHossz

Megjegyzés -- Azonosság?

„s1=s2” értelmezési lehetőségek:

– „tökéletesen” azonos állapot, azaz
s1.eleje=s2.eleje és s1.vége=s2.vége
(**értékmegosztás** esetén ⇒ azonos elemek)

– „lényegi” azonosság, azaz
azonos számú és értékű elemek.

Megjegyzés -- Értékadás

„s1:=s2” értelmezési lehetőségek:

– „tökéletesen” azonos állapot létrehozása, azaz
s1.eleje:=s2.eleje; s1.vége:=s2.vége
(**értékmegosztás** esetén ⇒ azonos elemek**)**

– **értékmásolás**, azaz
s1 minden elemének *s2*-másolatelem létrehozása.

Megjegyzés -- „Valami” paraméter

A Valami lehet tömb, lehet szekvenciális file egy blokkja, lehet egyéb külső perifériáról behozott adatcsomag, ...