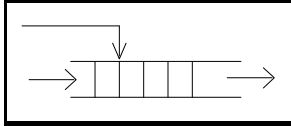


Prioritási sor típuskonstrukció



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

Copyright, 1999 © Szlávi Péter

Tartalomjegyzék

- 1 A prioritási sor algebrai specifikációja
 - 1.1 Prioritási sor-műveletek
 - 1.2 Prioritási sor-axiómák
- 2 A prioritási sor típuskonstrukció specifikációja
 - 2.1 A prioritási sor exportmodulja
 - 2.2 A prioritási sor megvalósítási moduljai
- 3 Alkalmazási példák

Programozásmódszertan 2 2005.03.15.

1 A prioritási sor algebrai specifikációja

1.1 Prioritási sor-műveletek

Típus PrSor(Elem,Prioritás):

Asszociált műveletek:

Üres: PrSor

Üres?(PrSor): Logikai

Tele?(PrSor): Logikai

Első(PrSor): (Elem × Prioritás) U {NemDef}

Sorba(PrSor,Elem,Prioritás): PrSor U {NemDef}

Sorból(PrSor): (PrSor × Elem × Prioritás) U {NemDef}

SorHossz(PrSor): Egész

1.2 Prioritási sor-axiómák

Axiómák:

1° Az Üres prioritási sor üres.

$$s \in \text{Üres} \Rightarrow \text{Üres?}(s) \wedge \text{SorHossz}(s)=0$$

2° Az a prioritási sor, amelyben legalább egy elem van, az nem üres; a sorba tétel során a sor hossza eggyel nő.

$$\neg \text{Tele?}(s) \Rightarrow \neg \text{Üres?}(\text{Sorba}(s,e,p)) \wedge \\ \text{SorHossz}(\text{Sorba}(s,e)) = \text{SorHossz}(s) + 1$$

3a° Az üres prioritási sornak nincs első eleme.

$$\text{Első?}(\text{Üres}) = \text{NemDef}$$

3b° Az Üres prioritási sorból nem lehet kivenni elemet.

$$\text{Sorból}(\text{Üres}) = \text{NemDef}$$

1.2 Prioritási sor-axiómák (folytatás)

4° A prioritási sor első eleme a legnagyobb prioritásúak közül a legrégebben betett elem.

$$\text{Üres?}(s) \wedge \neg \text{Tele?}(s) \Rightarrow \text{Első}(\text{Sorba}(s,e,p))=(e,p)$$

$$\neg \text{Üres?}(s) \wedge \neg \text{Tele?}(s) \Rightarrow$$

$$s'=\text{Sorba}(s,e,p) \wedge$$

$$(\text{Első}(s).\text{Pr} \geq p \Rightarrow \text{Első}(s')=\text{Első}(s)) \wedge$$

$$(\text{Első}(s).\text{Pr} < p \Rightarrow \text{Első}(\text{Sorba}(s,e,p))=(e,p)) \wedge$$

$$\text{SorHossz}(s')=\text{SorHossz}(s)+1$$

1.2 Prioritási sor-axiómák (folytatás)

5° A prioritási sorból az első elemet lehet kivenni (a többi nem változik).

$$\text{Üres?}(s) \wedge \neg \text{Tele?}(s) \Rightarrow \text{Sorból}(\text{Sorba}(s,e,p))=(s,e,p)$$

$$\neg \text{Üres?}(s) \wedge \neg \text{Tele?}(s) \Rightarrow$$

$$\text{Sorból}(s)=(s',e',p') \wedge (e',p')=\text{Első}(s) \wedge$$

$$(\text{Első}(s).\text{Pr} < p \Rightarrow e'=\text{Sorból}(\text{Sorba}(s,e,p).\text{Sor})) \wedge$$

$$(\text{Első}(s).\text{Pr} \geq p \Rightarrow \text{Sorba}(s',e,p)=\text{Sorból}(\text{Sorba}(s,e,p)).\text{Sor})$$

Állítás: a Sorból művelet után eggyel csökkent a prioritási sor hossza.

$$\neg \text{Üres?}(s) \Rightarrow \text{SorHossz}(\text{Sorból}(s).\text{Sor})=\text{SorHossz}(s)-1$$

Biz.:

2° & 5° -ből következik.

2 A prioritási sor típuskonstrukció specifikációja

2.1 A prioritási sor exportmodulja

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

ExportModul PrSor(**Típus** TElem: **Típus** TPrior):

Eljárás Üres(**Változó** s:PrSor)

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

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

Függvény ElsőElem(**Változó** s:PrSor): TElem

Függvény ElsőPrioritás(**Változó** s:PrSor): TPrior

Eljárás Sorba(**Változó** s:PrSor,
Konstans e:TElem, p:TPrior)

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

Programozásmódszertan

7

2005.03.15.

2.1 A prioritási sor exportmodulja (folytatás)

Eljárás Sorból(**Változó** s:PrSor, e:TElem, p:TPrior)

Infix Operátor Azonos?(**Konstans** s1,s2:PrSor):Logikai

Másnéven s1=s2

Infix Operátor LegyenEgyenlő(**Változó** s1:PrSor,
Konstans s2:PrSor)

Másnéven s1:=s2

Operátor Kiírás(**Konstans** s:PrSor)

Másnéven Ki: s

Operátor Beolvasás(**Változó** s:PrSor)

Másnéven Be: s

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

Modul vége.

Programozásmódszertan

8

2005.03.15.

2 A prioritási sor típuskonstrukció specifikációja

2.2. A prSor megvalósítási moduljai

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás

Modul PrSor(**Típus** TElem: **Típus** TPrior):

Reprezentáció

Konstans MaxHossz: Egész(???)

Típus TPrElem=**Rekord**(elem: TElem, pr: TPrior)

PrElemek=**Tömb**(1..MaxHossz: TPrElem)

[**TípusInvariáns**: növekvően...]

Változó se: PrElemek

eleje, vége, hossz : 0..MaxHossz

hiba: Logikai

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás (folytatás)

Implementáció

Eljárás Üres(**Változó** s:PrSor):

eleje:=1; vége:=1; hossz:=0; hiba:=Hamis

Eljárás vége.

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

Üres?:=hossz=0

Függvény vége.

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

Tele?:= hossz=MaxHossz

Függvény vége.

Függvény ElsőElem(**Változó** s:PrSor): TElem

Ha hossz≠0 **akkor** ElsőElem:=se(eleje).elem

különben hiba:=Igaz

Függvény vége.

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás (folytatás)

Függvény ElsőPrioritás(**Változó** s:PrSor): TPrior
Ha hossz \neq 0 **akkor** ElsőPrioritás:=se(eleje).pr
különben hiba:=Igaz

Függvény vége.

Eljárás Sorba(**Változó** s:PrSor,
Konstans e:TElem, p:TPrior):

Ha hossz $<$ MaxHossz **akkor**
Ha hossz $=$ 0 **akkor** hely:=eleje
különben Helykeresés(eleje,vége,hely,p)
Eltolás(hely,vége)

Elágazás vége
se(hely).elem:=e; se(hely).pr:=p
vége: \oplus 1; hossz:+1

...

Programozásmódszertan 11 2005.03.15.

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás (folytatás)

...

különben
hiba:=Igaz

Elágazás vége

Eljárás vége.

Eljárás Helykeresés(**Konstans** eleje,vége:0..MaxHossz,
Változó hely: 0..MaxHossz,
Konstans p:TPrior):

hely:=eleje
Ciklus amíg hely \neq vége \oplus 1 és se(hely).pr \geq p
hely: \oplus 1
Ciklus vége
[hely \neq vége \oplus 1 és se(hely).pr $<$ p **vagy**
hely=vége \oplus 1 \Rightarrow TipusInvariáns]

Eljárás vége.

Programozásmódszertan 12 2005.03.15.

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás (folytatás)

Eljárás Eltolás(Konstans hely,vége:0..MaxHossz):
Változó i:Egész
 i:=vége
Ciklus amíg i≠hely⊕1
 se(i⊕1):=se(i); i:⊕1
Ciklus vége
Eljárás vége.

Eljárás Sorból(Változó s:PrSor, e:TElem, p:TPrior):
Ha hossz>0 **akkor**
 p:=se(eleje).pr; e:=se(eleje).elem
 eleje:⊕1; hossz:-1
különben
 hiba:=Igaz
Elágazás vége
Eljárás vége.

Programozásmódszertan 13 2005.03.15.

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás (folytatás)

Függvény SorHossz(Konstans s:Sor): Egész
 SorHossz:=hossz
Függvény vége.

Függvény Hibás?(Változó s:PrSor): Logikai
 Hibás?:=hiba; hiba:=Hamis
Függvény vége.

Infix Operátor Azonos?(Konstans s1, s2:PrSor):Logikai
 Másnéven s1=s2
 ???
Operátor vége.

Infix Operátor LegyenEgyenlő(Változó s1:PrSor, Konstans s2:PrSor)
 Másnéven s1:=s2
 ???
Operátor vége.

Programozásmódszertan 14 2005.03.15.

2.2.1 Prioritás-szerint rendezett (folytonos) ábrázolás (folytatás)

Operátor Kiírás(Konstans s:PrSor)
 Másnéven Ki: s
 ...
Operátor vége.

Operátor Beolvasás(Változó s:PrSor)
 Másnéven Be: s
 ...
Operátor vége.

Inicializálás
 eleje:=1; vége:=1; hossz:=0; hiba:=Hamis

Modul vége.

Programozásmódszertan 15 2005.03.15.

2 A prioritási sor típuskonstrukció specifikációja

2.2. A prSor meglósítási moduljai

2.2.2 Időrendben rendezett (folytonos) ábrázolás

Modul PrSor(Típus TElem: Típus TPrior):
Reprezentáció
 ...
 ugyanaz, mint 2.2.1-ben
 ...

Programozásmódszertan 16 2005.03.15.

*2.2.2 Időrendben rendezett (folytonos)
ábrázolás (folytatás)*

Eljárás Üres(**Változó** s:PrSor):
...ugyanaz, mint 2.2.1-ben
Eljárás vége.

Függvény Üres?(**Konstans** s:PrSor): Logikai
...ugyanaz, mint 2.2.1-ben
Függvény vége.

Függvény Tele?(**Konstans** s:PrSor): Logikai
...ugyanaz, mint 2.2.1-ben
Függvény vége.

Függvény ElsőElem(**Változó** s:PrSor): TElem
Ha hossz \neq 0 **akkor** ElsőElem:=se(Max(s)).eleje
különben hiba:=Igaz
Függvény vége.

Programozásmódszertan 17 2005.03.15.

*2.2.2 Időrendben rendezett (folytonos)
ábrázolás (folytatás)*

Függvény Max(**Változó** s:PrSor): 1..MaxHossz
 $i:=e$ leje; $m:=i$
Ciklus $j=1$ -től hossz-1-ig
 $i:\oplus 1$; **Ha** se(i).pr > se(m).pr **akkor** $m:=i$
Ciklus vége
 $Max:=m$
Függvény vége.

Függvény ElsőPrioritás(**Változó** s:PrSor): TPrior
Ha hossz \neq 0 **akkor** ElsőPrioritás:=se(Max(s)).pr
különben hiba:=Igaz
Függvény vége.

Eljárás Sorba(**Változó** s:PrSor,
Konstans e:TElem, p:TPrior):
Ha hossz < MaxHossz **akkor**
se(vége).pr:=p; se(vége):=e; hossz:=+1; vége: $\oplus 1$

Programozásmódszertan 18 2005.03.15.

2.2.2 *Időrendben rendezett (folytonos)
ábrázolás (folytatás)*

különb
hiba:=Igaz
Elágazás vége
Eljárás vége.
Eljárás Sorból(Változó s:PrSor, e:TElem, p:Priritás):
Változó maxI:Egész
Ha hossz>0 **akkor**
maxI:=Max(s); e:=se(maxI).elem; p:=se(maxI).pr;
Rátolás(maxI,vége); hossz:-1; vége:⊙1

különb
hiba:=Igaz
Elágazás vége
Eljárás vége.
Függvény Hibás?(Változó s:PrSor): Logikai
...ugyanaz, mint 2.2.1-ben
Függvény vége.

Programozásmódszertan 19 2005.03.15.

2.2.2 *Időrendben rendezett (folytonos)
ábrázolás (folytatás)*

Infix Operátor Azonos?(Konstans s1, s2:PrSor):Logikai
... ugyanaz, mint 2.2.1-ben ...
Operátor vége.
Infix Operátor LegyenEgyenlő(Változó s1:PrSor,
Konstans s2:PrSor)
... ugyanaz, mint 2.2.1-ben ...
Operátor vége.
Operátor Kiírás(Konstans s:PrSor)
... ugyanaz, mint 2.2.1-ben ...
Operátor vége.
Operátor Beolvasás(Változó s:PrSor)
... ugyanaz, mint 2.2.1-ben ...
Operátor vége.
Inicializálás
... ugyanaz, mint 2.2.1-ben ...
Modul vége.

Programozásmódszertan 20 2005.03.15.

3 Alkalmazási példák

3.1 *Megszakítási rendszer*

3.2 *Mindennapok prioritási sorai*

Programozásmódszertan 21 2005.03.15.

3.1 *Megszakítási rendszer*

A számítógép az esetleg konkuráló (egyidejűleg érkező) megszakításokat egy *prioritási sorba* gyűjti, majd ennek sorrendjében hajtja végre az egyes megszakításokhoz rendelt rutinokat...

Programozásmódszertan 22 2005.03.15.

3.2 *Mindennapok prioritási sorai*

Az Élet „dezertőrei” mindenki által kellően átélt példákkal illusztrálják a prioritási sorok „létfogosultságát”, de legalábbis létezését.

Programozásmódszertan 23 2005.03.15.

Megjegyzés -- Változó

A „Változó”-ság oka:

- valamilyen hiba lehetősége fönáll (üres vagy tele a sor), s ennek visszajelzésére a „hiba” mező változhat.
- Vagy valamely operáció (l. *Hiba függvény*) kifejezetten állítja.

Programozásmódszertan 24 2005.03.15.

Megjegyzés -- Tele?

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

Programozásmódszertan 25 2005.03.15.

Megjegyzés -- ciklikus növelés

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

$\oplus 1: \{1..MaxHossz\} \rightarrow \{1..MaxHossz\}$
 $x \oplus 1 := (x \bmod MaxHossz) + 1,$
 $\ominus 1: \{1..MaxHossz\} \rightarrow \{1..MaxHossz\}$
 $x \ominus 1 := (x - 1 + MaxHossz) \bmod MaxHossz$

Programozásmódszertan 26 2005.03.15.

Megjegyzés -- Azonosság?

„ $s1=s2$ ” értelmezési lehetőségek:

- „tökéletesen” azonos állapot, azaz
 $s1.eleje=s2.eleje$ és $s1.vége=s2.vége$
(**értékmegosztás** esetén \Rightarrow 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 \Rightarrow azonos elemek)
- **értékmásolás**, azaz
 $s1$ minden elemének $s2$ -másolalelem létrehozása.

Megjegyzés -- Prioritás

Az elemek *rendezésére* szolgáló többletinfor-
máció.

Infix Operátor \leq (TPrior,TPrior): Logikai

Megvalósítása:

- az elem egy „rendezett” mezője,
- az elemen értelmezett valamilyen alkalmas *rendezett* értéket szolgáltató függvény.