

8. előadás

Programozás-elmélet

Jelölje $[\]$ az üres halmazt, $[a..b]$ részintervallum típusú halmazt, $[a, b, c, d, \dots]$ felsorolással megadott halmazt. A halmaz típus egy megadott T_0 alaptípus részalmazainak halmazát jelenti. Formális definíciója:

$$\text{type } T = \text{set of } T_0.$$

A T_0 típus részalmazai a

$$D_T = 2^{T_0} = \{S \mid S \subset D_{T_0}\}$$

hatványhalmazt alkotják.

Példa:

$$\text{type } T = \text{set of } [1..3]$$

esetén

$$D_T = \{[\], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]\}.$$



A halmaztípussal megengedett műveletek:

- 1 $A + B$ ($A + B = A \cup B$)
- 2 $A * B$ ($A * B = A \cap B$)
- 3 $A - B$ ($A - B = A \setminus B$)

A logikai relációk:

- 1 $A = B$
- 2 $A \neq B$
- 3 $A \leq B \iff A \subset B$
- 4 $A \geq B \iff B \leq A$
- 5 $a \text{ in } B$

$$a \text{ in } B = \begin{cases} \text{true,} & \text{ha } a \in B \\ \text{false,} & \text{ha } a \notin B \end{cases}$$

A sorozat típust felfoghatjuk úgy, mint egy T_0 típusú adatokból álló változó hosszúságú tömbök (vektorok) halmazát:

type $T = \text{sequence of } T_0.$

Ennek megfelelően

$$D_T = \bigcup_{n=0}^{\infty} D_{T_0^n},$$

ahol $T_0^n = \underbrace{T_0 \times \dots \times T_0}_n.$

Példák

type *page* = sequence of *character*

type *book* = sequence of *page*

= sequence of sequence of *character*

Az e_1, e_2, \dots, e_n elemekből álló T típusú sorozatot

$$T \langle e_1, e_2, \dots, e_n \rangle$$

formában jelöljük. Ha $n = 0$, akkor ezt $T \langle \rangle$ jelöli, és üres sorozatnak nevezzük. Az elemek kiválasztására a következő függvényeket vezetjük be. Legyen x sorozat.

- $x[i]$ - az x sorozat i -edik eleme
- $first(x)$ - az x sorozat első eleme
- $last(x)$ - az x sorozat utolsó eleme

Ha $x = T \langle e_1, e_2, \dots, e_n \rangle$, akkor

$$x[i] = e_i \quad (1 \leq i \leq n), \quad \text{first}(x) = e_1, \quad \text{last}(x) = e_n.$$

Tehát $[i], \text{first}, \text{last} : D_T \rightarrow D_{T_0}$.

Műveletek sorozatok elemeivel

- a) $\text{tail}(x)$ - az x sorozat első elemének elhagyásával nyert sorozat
- b) $\text{initial}(x)$ - az x sorozat utolsó elemének elhagyásával nyert sorozat
- c) $\text{appendl}(x, e)$ - az e elem x sorozat elé írásával nyert sorozat
- d) $\text{appendr}(x, e)$ - az e elem x sorozat után írásával nyert sorozat.

Ha $x = T \langle e_1, e_2, \dots, e_n \rangle$, akkor

$$\mathit{tail}(x) = T \langle e_2, \dots, e_n \rangle,$$

$$\mathit{initial}(x) = T \langle e_1, \dots, e_{n-1} \rangle,$$

$$\mathit{appendl}(x, e) = T \langle e, e_1, e_2, \dots, e_n \rangle,$$

$$\mathit{appendr}(x, e) = T \langle e_1, e_2, \dots, e_n, e \rangle.$$

értelemszerűen:

$$\mathit{tail}, \mathit{initial} : D_T \rightarrow D_T, \quad \mathit{appendl}, \mathit{appendr} : D_T \times D_{T_0} \rightarrow D_T.$$

Ekkor teljesülnek a következő összefüggések:

- $first(appendl(x, e)) = e,$
- $tail(appendl(x, e)) = x,$
- $appendl(tail(x), first(x)) = x,$ ha $x \neq T \langle \rangle,$
- $last(appendr(x, e)) = e,$
- $initial(appendr(x, e)) = x,$
- $appendr(initial(x), last(x)) = x,$ ha $x \neq T \langle \rangle.$

Bevezetjük még a következő függvényt:

$$empty(x) = \begin{cases} true, & \text{ha } x = T \langle \rangle \\ false, & \text{ha } x \neq T \langle \rangle \end{cases}$$

A szekvenciális fájl megadása:

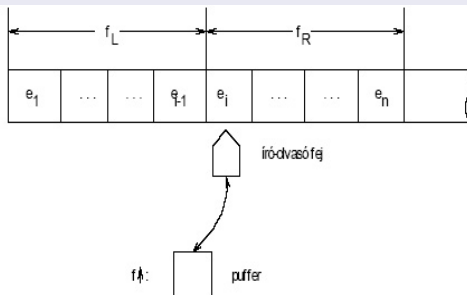
```
type  $T = \text{file of } T_0$ 
```

Példa

```
type szoveg = file of char
```

Az f szekvenciális fájl szerkezetet legegyszerűbben egy mágnesszalagon egymásután elhelyezett adatokkal jeleníthetjük meg:

Szekvenciális fájl



A fájl elemeinek olvasása és írása egy író-olvasó fejen keresztül történik. Egyszerre csak egy adatot lehet olvasni, amelyet az "író-olvasó fej" pozíciója határoz meg. Ha a pozíció az i -edik elemnél van, akkor a fájl tartalmát két halmazra bontja:

$$f_L = \langle e_1, \dots, e_{i-1} \rangle, \quad f_R = \langle e_i, \dots, e_n \rangle.$$

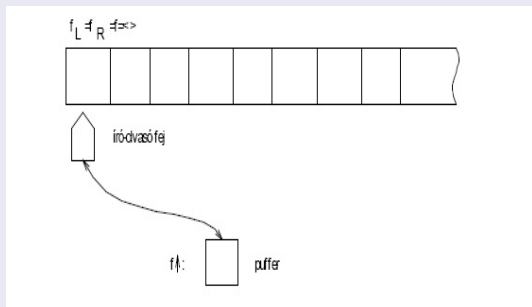
Az egész fájl tartalma ekkor konkatenációval

$$f = f_L f_R = \langle e_1, \dots, e_{i-1}, e_i, \dots, e_n \rangle.$$

Az írás-olvasás egy pufferen keresztül történik, amelynek tartalmát az $f \uparrow$ változó tartalmazza. Az $f \uparrow$ típusa azonos a fájl elemek típusával.

A szekvenciális fájlknál 4 művelet (*rewrite*, *reset*, *write*, *read*) és egy logikai függvényt (*eof*) engedünk meg.

A szekvenciális fájlban nem lehet az egyes komponenseket felülírni. Az egyetlen lehetséges út a fájl törlése és újraírása. Ezt a *rewrite* (újraírás) utasítás végzi el.



A *rewrite* (f) művelet az

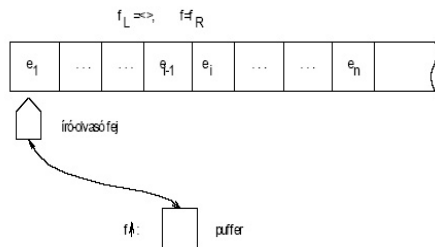
$$f := \langle \rangle$$

értékadást jelenti. Ekkor írhatjuk azt is, hogy

$$\text{rewrite}(f) \iff f_L = \langle \rangle, f_R = \langle \rangle.$$

Vegyük észre, hogy a *rewrite* utasítással hozhatunk létre üres fájlt.

A *reset* utasítással az író-olvasó fej a fájl első pozíciójára helyezhető:



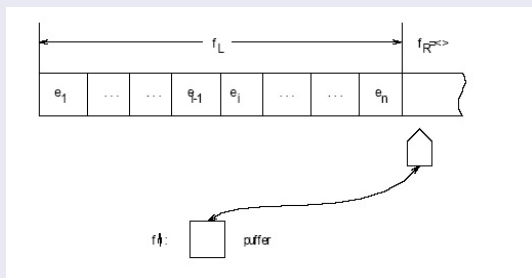
Az $eof(f)$ (end of file) logikai függvény a fájl végét jelzi.

Definíciója:

$$eof(f) \equiv f_R \equiv \langle \rangle .$$

Tehát $eof(f) = true$, ha $f_R = \langle \rangle$, és $eof(f) = false$, ha $f_R \neq \langle \rangle$.

A fájlba új elemet írni csak az utolsó elem után lehetséges a *write* (Pascalban *put(f)*) utasítással.



Az írási pozícióban $f_R = \langle \rangle$, $eof(f) = true$ és

$$write(f, e) \iff f_L = appendr(f_L, e), f_R = \langle \rangle.$$

Figyeljük meg, hogy az írás után, újra írási pozícióba kerülünk. A fájlból olvasni csak az $eof(f) = hamis$ esetben lehetséges. A $read$ (Pascalban $get(f)$) olvasási utasítás végrehajtása után az olvasó fej egy pozícióval jobbra mozdul:

$$read(f, e) \iff f_L = appendr(f_L, first(f_R)), f_R = tail(f_R), e = first(f_R).$$

Itt e azt az elemet jelöli, amelynél az író-olvasó fej aktuálisan áll.

A verem olyan sorozattípus, ahol az írás és a törlés a sorozat ugyanazon végén történik (Last in First Out = LIFO):

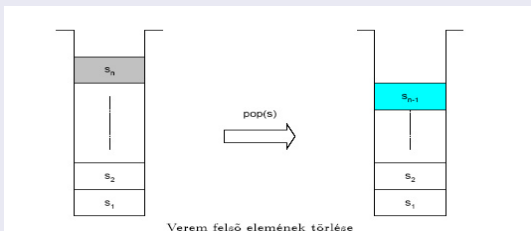
```
var:  s : stack of  $T_0$ ;  
      e :  $T_0$ 
```

Három művelete van:

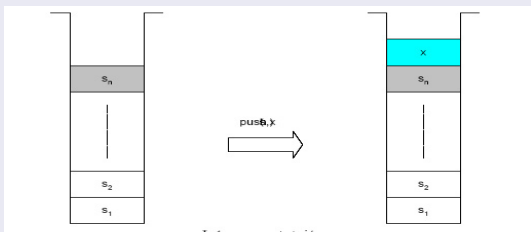
- ① *top* - megadja a sorozat utolsó elemét (verem tetejét):

$$top(s) = last(s).$$

- ② *pop* - törli a sorozat utolsó elemét
③ *push* - új elemet ír a sorozat végére



Képlettel kifejezve: $pop(s) \iff s = initial(s)$.



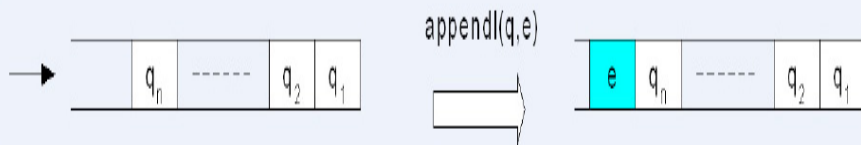
Képlettel kifejezve: $push(s, x) \iff appendr(s, x)$.

A sor olyan sorozattípus, ahol az írás a sorozat elején (*tkp. végén*), a törlés a sorozat végén (*tkp. elején*) történik (First in First Out = FIFO):

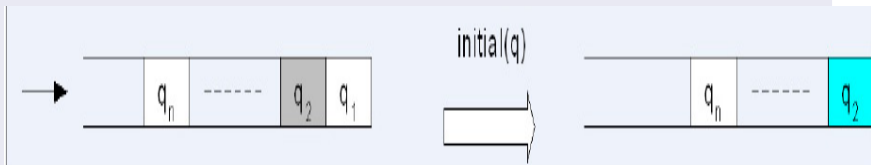
```
var:  q : queue of  $T_0$ ;  
      e :  $T_0$ 
```

Két művelete van:

- 1 *enter* - új elem írása a sorozat végére
- 2 *leave* - a sorozat első elemének törlése



Képlettel kifejezve: $\text{enter}(q, e) \iff q = \text{appendl}(q, e)$.



Képlettel kifejezve: $\text{leave}(q) \iff q = \text{initial}(q)$.