ELTE PROG-MAT.
2000-2001

8.

Típuskonstrukciók

Az előző fejezetben megvizsgáltuk, hogy milyen lehetőségeink vannak meglevő programokból újak készítésére. A továbbiakban azt fogjuk megvizsgálni, hogyan használhatunk fel meglévő típusokat új típusok létrehozására. Ezeket a módszereket típuskonstrukciós módszereknek, az általuk megkapható típusokat típuskonstrukcióknak nevezzük.

8.1. A megengedett konstrukciók

Természetesen sokféle lehetőségünk van meglevő típusból újat csinálni, de mi a továbbiakban csak három speciális típuskonstrukcióval fogunk foglalkozni: a direktszorzattal, az unióval és az iteráltal. Ezeket fogjuk megengedett típuskonstrukcióknak nevezni.

Az első típuskonstrukciós módszer, amivel megismerkedünk a direktszorzat. Legyenek $\mathcal{T}_i = (\varrho_i, I_i, \mathbb{S}_i)$ $(i = 1, 2, \ldots, n)$ típusok, és jelöljük $T_1, T_2, \ldots, T_n$-nel a nekik megfelelő típusértékhalmazokat, és $E_1, E_2, \ldots, E_n$-nel pedig a hozzájuk tartozó elemi típusértékhalmazokat, és vezessük be az $E = E_1 \cup E_2 \cup \cdots \cup E_n$ és $B = T_1 \times T_2 \times \cdots \times T_n$ jelölést.

---

32. DEFINÍCIÓ: DIREKTSZORZAT

A $\mathcal{T} = (\varrho, I, \mathbb{S})$ típus direktszorzata a $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ típusoknak, ha

$$\varrho = \varphi_D \circ \psi_D$$

ahol $\varphi_D \subseteq B \times T$, $\psi_D \subseteq E^* \times B$ és

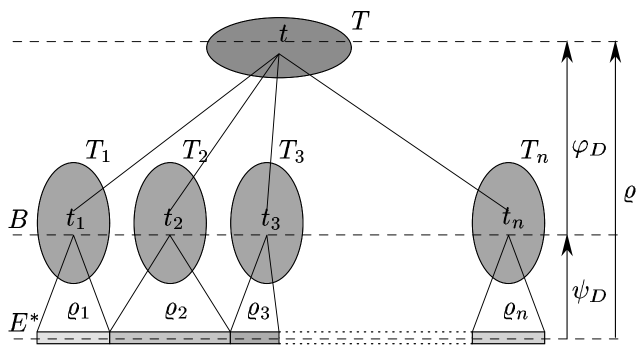
$$\psi_D = \{(\varepsilon, b) \in E^* \times B \mid \forall i \in [1..n] : \exists \varepsilon_i \in E_i^* : (\varepsilon_i, b_i) \in \varrho_i \ \wedge \\ \varepsilon = con(\varepsilon_1, \ldots, \varepsilon_n)\}.$$

A direktszorzat értékhalmazára bevezetjük a $T = (T_1, T_2, \ldots, T_n)$ jelölést.

Ha a φ_D leképezés kölcsönösen egyértelmű, akkor a direktszorzatot *rekord* típusnak nevezzük. A direktszorzat típusok általában rekordok, de nem mindig. Például tekintsük a racionális számok halmazának egy lehetséges reprezentációját:
$B = \mathbb{Z} \times \mathbb{Z}$, $\varphi_D \subseteq B \times \mathbb{Q}$:

$$((x, y), t) \in \varphi_D \Longleftrightarrow y \neq 0 \wedge t = x/y$$

Egyszerűen látható, hogy a fent definiált φ_D reláció a racionális számok halmazát reprezentálja, de nem kölcsönösen egyértelmű.



8.1. ábra. Rekord konstrukció

Nagyon fontos továbbá, hogy az új típusértékhalmazt (T) ne keverjük össze a közbülső direktszorzattal (B), hiszen egy adott B és T között nagyon sokféle φ_D leképezés megadható, és az új típus szempontjából egyáltalán nem mindegy, hogy ezek közül melyiket választjuk.

Tekintsük például a komplex egészek ($a + bi, a, b \in \mathbb{Z}$ alakú számok) halmazát. Legyen továbbá $B = \mathbb{Z} \times \mathbb{Z}$, $x, y \in \mathbb{Z}$, és

$$\varphi_{D_1}((x, y)) = x + yi$$
$$\varphi_{D_2}((x, y)) = y + xi.$$

A két φ_D közötti különbség elsősorban akkor válik szignifikánssá, ha például a komplex egészek közötti szorzásműveletet kell implementálnunk a számpárok szintjén, hiszen ekkor az első és a második komponens értékét az

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$$

formula alapján különböző módon kell kiszámítani.

A következő módszer, amivel régi típusokból újakat hozhatunk létre, az unió. Legyenek $\mathcal{T}_i = (\varrho_i, I_i, \mathbb{S}_i)$ $(i = 1, 2, \ldots, n)$ típusok, és jelölje $T_1, T_2, \ldots, T_n$ a hozzájuk tartozó típusértékhalmazokat, illetve $E_1, E_2, \ldots, E_n$ a nekik megfelelő elemi típusértékhalmazokat. Vezessük be továbbá az $E = E_1 \cup E_2 \cup \cdots \cup E_n$ és $B = T_1 \cup T_2 \cup \cdots \cup T_n$ jelöléseket.

33. DEFINÍCIÓ: UNIÓ

Azt mondjuk, hogy a $\mathcal{T} = (\varrho, I, \mathbb{S})$ típus uniója a $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ típusoknak, ha

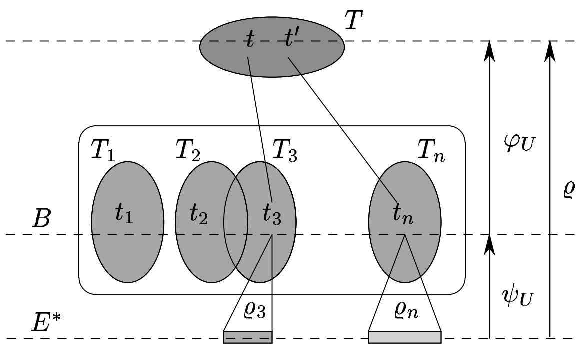
$$\varrho = \varphi_U \circ \psi_U$$

ahol $\varphi_U \subseteq B \times T$, $\psi_U \subseteq E^* \times B$ és

$$\psi_U = \{(\varepsilon, b) \in E^* \times B \mid \exists i \in [1..n] : (\varepsilon, b) \in \varrho_i\}$$

Az unió értékhalmazának jelölése: $T = (T_1; T_2; \ldots; T_n)$.

Itt is külön elnevezést adtunk annak az esetnek, amikor a φ_U leképezés kölcsönösen egyértelmű, ekkor az uniót *egyesítésnek* nevezzük.



8.2. ábra. Unió konstrukció

Ebben az esetben is nagyon fontos, hogy mindig megkülönböztessük a konstrukció közbülső szintjén levő uniót (B) az új típusértékhalmaztól (T).

A harmadik megengedett típuskonstrukciós művelet az iterált, amellyel egy meglevő típusból alkothatunk új típust. Legyen $\mathcal{T}_0 = (\varrho_0, I_0, \mathbb{S}_0)$ típus, T_0 a neki megfelelő típusértékhalmaz, és E a $\mathcal{T}_0$ típus elemi típusértékhalmaza.

34. DEFINÍCIÓ: ITERÁLT

Azt mondjuk, hogy a $\mathcal{T} = (\varrho, I, \mathbb{S})$ típus iteráltja a $\mathcal{T}_0$ típusnak, ha

$$\varrho = \varphi_I \circ \psi_I$$

ahol $\varphi_I \subseteq B \times T$, $\psi_I \subseteq E^* \times T_0^*$ és

$$\psi_I = \{(\varepsilon, b) \in E^* \times T_0^* \mid \exists \varepsilon_1, \ldots, \varepsilon_{|b|} \in E^* : (\varepsilon_i, b_i) \in \varrho_0 \ \wedge \ \varepsilon = kon(\varepsilon_1, \ldots, \varepsilon_{|b|})\}$$

Az iterált típusértékhalmazának jelölése: $T = it(T_0)$.

Az iterált típuskonstrukciónak három speciális esetét különböztetjük meg aszerint, hogy a φ_I leképezésre milyen feltételek teljesülnek.

- Ha a φ_I leképezés kölcsönösen egyértelmű, akkor *sorozat* típuskonstrukcióról beszélünk, és típusértékhalmazát $T = seq(T_0)$-lal jelöljük.
- Ha

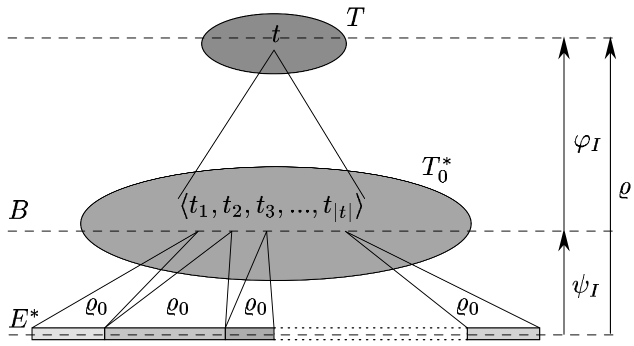
  $$(\alpha, t), (\beta, t) \in \varphi_I \Leftrightarrow \alpha \in perm(\beta),$$

  akkor az iterált konstrukcót *kombináció* típusnak nevezzük. A kombináció értékhalmazának jelölése: $T = com(T_0)$.
- Ha

  $$(\alpha, t), (\beta, t) \in \varphi_I \Leftrightarrow \bigcup_{i=1}^{|\alpha|} \{\alpha_i\} = \bigcup_{i=1}^{|\beta|} \{\beta_i\},$$

  akkor *halmaz* típuskonstrukcióról beszélünk. A halmaz típus értékhalmazának jelölése: $T = set(T_0)$.

Természetesen az imént felsorolt három eset csak speciális formája az iteráltképzésnek; létezik olyan iterált is, amely egyik fenti kritériumot sem teljesíti.



8.3. ábra. Iterált konstrukció

8.2. Szelektorfüggvények

Az előzőkben definiált típuskonstrukciókra most bevezetünk néhány olyan függvényt és jelölést, amelyek leegyszerűsítik a rájuk vonatkozó állítások, programok megfogalmazását.

Legyen $T = (T_1, T_2, \ldots, T_n)$. A $\varphi_D^{(-1)}$ függvény komponenseit a T rekord szelektorfüggvényeinek, vagy röviden szelektorainak nevezzük.

A fenti rekordnak tehát pontosan n darab szelektora van, és ha s_i-vel jelöljük az i-edik szelektort, akkor $s_i : T \to T_i$, és

$$\forall t \in T : \varphi_D(s_1(t), s_2(t), \ldots, s_n(t)) = t.$$

Tehát a szelektorfüggvényeket arra használhatjuk, hogy lekérdezzük a rekord egyes mezőinek (komponenseinek) az értékét. A szelektorokat bele szoktuk írni a típusértékhalmaz jelölésébe; a fenti esetben a szelektorokkal felírt jelölés:

$$T = (s_1 : T_1, s_2 : T_2, \ldots, s_n : T_n).$$

A rekordtípushoz hasonlóan az egyesítéshez is bevezetünk szelektorfüggvényeket. Mivel az unió esetében a közbülső szinten a típusértékhalmazok uniója szerepel, így nincs értelme komponensről beszélni. Hogyan definiáljuk ez esetben a szelektorokat? Azt fogják visszaadni, hogy egy adott T-beli elemet melyik eredeti típusértékhalmaz egy eleméhez rendelte hozzá a φ_U függvény.

Legyen $T = (T_1; T_2; \ldots; T_n)$ egyesítés típus, $s_i : T \to \mathbb{L}$ $(i = 1, \ldots, n)$. Azt mondjuk, hogy az s_i logikai függvények a T egyesítés szelektorai, ha $\forall i \in [1..n]$: $\forall t \in T$:

$$s_i(t) = \left(\varphi_U^{(-1)}(t) \in T_i\right).$$

A rekordtípushoz hasonló módon az egyesítés szelektorait is bele szoktuk írni az új típusértékhalmaz jelölésébe. A szelektorokkal felírt típusértékhalmaz jelölése:

$$T = (s_1 : T_1; s_2 : T_2; \ldots; s_n : T_n).$$

Az iterált típuskonstrukciók közül a sorozathoz definiálunk szelektorfüggvényt. A sorozattípusban a közbülső szinten T_0-beli sorozat szerepel, a szelektor ennek a sorozatnak a tagjait adja vissza.

Formálisan: Legyen $T = seq(T_0)$. Az $s : T \times \mathbb{N} \to T_0$ parciális függvény a T szelektorfüggvénye, ha $\forall t \in T : \forall i \in [1..|\varphi_I^{(-1)}(t)|]$:

$$s(t, i) = \varphi_I^{(-1)}(t)_i.$$

A sorozat szelektorát nem szoktuk külön elnevezni, helyette indexelést alkalmazunk, azaz az $t_i = s(t, i)$ jelölést használjuk.

8.3. Az iterált specifikációs függvényei

Ha az iterált típus az előzőekben bevezetett három speciális osztály valamelyikébe tartozik, akkor további függvényeket definiálunk hozzá.

Legyen $T = it(T_0)$, $(\alpha, t) \in \varphi_i$, és tegyük fel, hogy az iterált sorozat, kombináció, vagy halmaz. Ekkor $dom : T \to \mathbb{N}_0$,

$$dom(t) = \begin{cases} |\alpha|, & \text{ha } T = seq(T_0) \text{ vagy } T = com(T_0) \\ |\bigcup\limits_{i=1}^{|\alpha|} \{\alpha_i\}|, & \text{ha } T = set(T_0) \end{cases}$$

A dom függvény tehát a t elemeinek számát adja meg. A függvény jóldefiniált, ugyanis felhasználva a sorozat, kombináció és halmaz típus definícióját, könnyen látható, hogy a függvényérték független az α választásától.

A továbbiakban a sorozattípussal fogunk foglalkozni. Ahol külön nem jelöljük, ott $T = seq(T_0)$, $(\alpha, t) \in \varphi_I$, $\alpha = \langle \alpha_1, \alpha 2, \dots \alpha_{|\alpha|} \rangle$.

- Nem üres sorozat első és utolsó eleme: $lov : T \to T_0$, $hiv : T \to T_0$,

$$\begin{aligned} lov(t) &= \alpha_1 \\ hiv(t) &= \alpha_{|\alpha|} \end{aligned}$$

- Sorozat kiterjesztése a sorozat elején, vagy végén (legyen $e \in T_0$): $loext : T \times T_0 \to T$, $hiext : T \times T_0 \to T$,

$$\begin{aligned} loext(t, e) &= \varphi_I(con(\langle e \rangle, \alpha)) \\ hiext(t, e) &= \varphi_I(con(\alpha, \langle e \rangle)) \end{aligned}$$

- Nem üres sorozat első, vagy utolsó elemének elhagyásával kapott sorozat: $lorem : T \to T$, $hirem : T \to T$,

$$\begin{aligned} lorem(t) &= \varphi_I(\langle \alpha_2, \dots, \alpha_{|\alpha|} \rangle) \\ hirem(t) &= \varphi_I(\langle \alpha_1, \dots, \alpha_{|\alpha|-1} \rangle) \end{aligned}$$

8.4. A függvénytípus

A gyakorlatban nagyon fontos szerepet játszik egy speciális rekordtípus. Legyen H egy tetszőleges (megszámlálható) halmaz, amelyen van egy rákövetkezési reláció. Jelöljük ezt a rákövetkezési relációt $succ$-cal, és inverzére vezessük be a $pred$ jelölést.

35. DEFINÍCIÓ: FÜGGVÉNY TÍPUS

Legyen E egy tetszőleges típus értékhalmaza. Ekkor az $F = (H, seq(E))$ rekordot függvénytípusnak nevezzük, és $F = fun(H, E)$-vel jelöljük.

A függvénytípusra is bevezetünk néhány fontos specifikációs függvényt. A továbbiakban legyen $F = fun(H, E)$, $((h, t), f) \in \varphi_D$. Ekkor

- $dom : F \to \mathbb{N}_0$,
$$dom(f) = dom(t)$$
- $lob : F \to H$,
$$lob(f) = h$$
- $hib : F \to H$,
$$hib(f) = succ^{dom(f)-1}(h)$$
- $lov : F \to E$,
$$lov(f) = lov(t)$$
- $hiv : F \to E$,
$$hiv(f) = hiv(t)$$
- $loext, hiext : F \times E \to F$,
$$\begin{aligned} loext(f, e) &= \varphi_D(pred(h), loext(t, e)) \\ hiext(f, e) &= \varphi_D(h, hiext(t, e)) \end{aligned}$$
- $lorem, hirem : F \to F$,
$$\begin{aligned} lorem(f) &= \varphi_D(succ(h), lorem(t)) \\ hirem(f) &= \varphi_D(h, hirem(t)) \end{aligned}$$

A sorozathoz hasonlóan a függvénytípusra is bevezetünk egy szelekciós parciális függvényt. TekintsÜk a fentiekben használt f-et. Ekkor $s_f : H \to E$, $\mathcal{D}_{s_f} = \{succ^i(lob(f)) \mid 0 \leq i < dom(f)\}$, és ha $g \in \mathcal{D}_{s_f}$, $g = succ^k(lob(f))$, akkor:
$$s_f(g) = t_{k+1}$$
A függvénytípus szelektorfüggvényét nem szoktuk külön elnevezni, helyette a matematikában – a függvény helyettesítési értékének jelölésére – használt zárójeles hivatkozást használjuk, azaz
$$f(g) := s_f(g).$$
A függvénytípus elnevezés azt a szemléletes képet tükrözi, hogy egy függvénytípusú érték felfogható egy $H \to E$ típusú parciális függvénynek, amelynek értelmezési tartománya a "lob-tól a hib-ig tart", és értékeit pedig a sorozat komponens tartalmazza.

Az előbbiekben bevezetett dom, lov, hiv, lob, hib függvényeket kiterjeszthetjük az egész állapottérre is: komponáljuk a megfelelő változóval. Tehát ha például x egy sorozattípusú változó, akkor $dom \circ x$ egy az egész állapottéren értelmezett függvény. Az ilyenfajta függvénykompozíciókra bevezetünk egy újabb jelölést: ha t a fenti függvények valamelyike, és x a neki megfelelő típusú változó, akkor az $t \circ x$ helyett $x.t$-t írunk.

8.5. A típuskonstrukciók típusműveletei

A típuskonstrukciók eddigi tárgyalásából még hiányzik valami: nem beszéltünk még a konstruált típusok műveleteiről. Az előbb felsorolt speciális esetekhez – az imént definiált függvények segítségével – bevezetünk néhány típusműveletet.

A továbbiakban megengedett feltételnek fogjuk nevezni azokat az $A \to \mathbb{L}$ állításokat, amelyek lehetnek elágazás, vagy ciklus feltételei.

Legyen $T = (s_1 : T_1, \ldots, s_n : T_n)$ rekord, $t : T$, $t_i : T_i$ ($i \in [1..n]$). Mivel t az állapottér változója, t komponálható a szelektorfüggvényekkel, és így az állapotéren értelmezett függvényeket kapunk. A $s_i \circ t$ függvénykompozíciót a továbbiakban $t.s_i$-vel fogjuk jelölni. Egy rekord típusnál a szelektorfüggvény használatát megengedettnek tekintjük.

Ezen kívül bevezetjük a $t.s_i := t_i$ jelölést is. Ezen azt a $t := t'$ értékadást értjük, amelyben $t'.s_i = t_i$ és t' minden más komponense megegyezik t megfelelő komponensével.

A fenti típusműveletek arra adnak lehetőséget, hogy egy rekord “mezőinek" az értékét lekérdezhessük, illetve megváltoztathassuk. A fent definiált műveletben zavaró lehet, hogy egy függvény helyettesítési értékének ($t.s_i$) “adunk értéket". Ezért fontos megjegyezni, hogy ez csupán egy jelölése az értékadásnak.

Legyen $T = (s_1 : T_1; \ldots; s_n : T_n)$ egyesítés, $t : T$, $t_i : T_i$ ($i \in [1..n]$). Ekkor a rekord típusnál bevezetett jelölést az egyesítés esetén is bevezetjük, $t.s_i$-n, az $s_i \circ t$ kompozíciót értjük, és megengedett függvénynek tekintjük.

Ezen kívül megengedett művelet a $t := \varphi_U(t_i)$ értékadás. Ennek az értékadásnak a jelölését leegyszerűsítjük, a továbbiakban $t := t_i$ alakban fogjuk használni.

A fenti értékadást bizonyos ésszerű korlátozások bevezetésével “megfordíthatjuk". Így kapjuk az alábbi parciális értékadást: $t_i := t$. Ez az értékadás csak akkor végezhető el, ha $t.s_i$ igaz.

A sorozat típuskonstrukció nagyon gyakori, és sokféle művelet definiálható vele kapcsolatban. Attól függően, hogy melyeket tekintjük megvalósítottnak, különböző konstrukciókról beszélünk. Most előbb megadunk néhány lehetséges műveletet, majd ezután a sorozat típusokat osztályba soroljuk a megengedett műveleteik alapján.

Legyen a továbbiakban $T = seq(E)$, $t : T$, $e : E$. Ekkor az iménti szelektorokhoz hasonlóan bevezetjük az alábbi jelöléseket:

$$\begin{array}{rcl} dom \circ t & \to & t.dom \\ lov \circ t & \to & t.lov \\ hiv \circ t & \to & t.hiv \end{array}$$

Természetesen $t.lov$ és $t.hiv$ csak parciális függvények. Ezen kívül az alábbi (esetleg parciális) értékadásokra a következő jelöléseket fogjuk használni:

$$\begin{aligned}
t := lorem(t) &\quad\rightarrow\quad t : lorem \\
t := hirem(t) &\quad\rightarrow\quad t : hirem \\
t := loext(t, e) &\quad\rightarrow\quad t : loext(e) \\
t := hiext(t, e) &\quad\rightarrow\quad t : hiext(e) \\
e, t := lov(t), lorem(t) &\quad\rightarrow\quad e, t : lopop \\
e, t := hiv(t), hirem(t) &\quad\rightarrow\quad e, t : hipop
\end{aligned}$$

A bevezetett jelölések első látásra zavarba ejtőnek tűnhetnek, hiszen ugyanazt a kulcsszót a baloldalon függvényként, a jobboldalon pedig a művelet neveként használjuk. Lényeges ezért megjegyezni, hogy a jobb oldalon található műveletek csa a baloldali értékadás egyszerűsítő *jelölései*.

Attól függően, hogy a fent definiált műveletek közül melyeket tekintjük megengedettnek, különböző konstrukciókról beszélünk.

---

36. DEFINÍCIÓ: SPECIÁLIS SOROZATTÍPUSOK
Legyen $T = seq(E)$. Ekkor a T

- *szekvenciális input file*, ha csak a $lopop$ a megengedett művelet,
- *szekvenciális output file*, ha csak a $hiext$ a megengedett művelet,
- *verem*, ha a megengedett műveletek a $loext$ és a $lopop$, vagy a $hiext$ és a $hipop$,
- *sor*, ha a megengedett műveletek a $hiext$ és a $lopop$, vagy a $loext$ és a $hipop$.

Ahhoz, hogy a szekvenciális input file a $lopop$ művelettel használható legyen, tudnunk kell, hogy mikor olvastuk ki az utolsó elemet a file-ból. Ezt a problémát úgy szoktuk megoldani, hogy bevezetünk egy extremális elemet, és kikötjük, hogy a file-ban ez az utolsó elem (tehát még az üres file is tartalmazza). Ez a technika valósul meg azokban az operációs rendszerekben, ahol a szövegfile-ok végét file-vége (EOF) karakter jelzi.

Mivel a $lopop$ művelet bizonyos esetekben kényelmetlen lehet – gondoljunk arra, amikor nehézkes extremális elemet találni –, bevezetünk egy másik olvasóműveletet is. Használjuk az olvasás sikerességének jelzésére a $\{norm, abnorm\}$ halmaz elemeit. Ekkor az $sx, dx, x : read$ műveleten az alábbi értékadásokat értjük:

$$sx, dx, x : read \rightarrow \begin{cases} sx, dx, x := norm, lov(x), lorem(x), & \text{ha } dom(x) \neq 0 \\ sx := abnorm, & \text{ha } dom(x) = 0 \end{cases}$$

Ha egy szekvenciális file-ra a $read$ művelet van megengedve, akkor nincs szükség extremális elemre, helyette az sx változó értéke alapján lehet eldönteni, hogy végére értünk-e a file-nak.

Legyen $F = fun(H, E)$, $f : F$, $e : E$, $i : H$. Ekkor a sorozat típushoz hasonlóan bevezetjük az alábbi jelöléseket:

$$\begin{array}{rcl} dom \circ f & \to & f.dom \\ lov \circ f & \to & f.lov \\ hiv \circ f & \to & f.hiv \\ lob \circ f & \to & f.lob \\ hib \circ f & \to & f.hib \end{array}$$

A fenti függvényeken kívül a függvény típus szelektorfüggvényét $f(i)$-t is megengedettnek tekintjük. A rekord típusnál bevezetett szelektorra (mezőre) vonatkozó értékadásnak jelen esetben is van megfelelője: az $f(i) := e$ parciális értékadás. Az értékadás azért parciális, mert csak akkor végezhető el, ha $f.lob \le i \le f.hib$. Ekkor a fenti jelölésen azt az $f := f'$ értékadást értjük, amelyre:

$$\begin{array}{l} f'.lob = f.lob \wedge f'.hib = f.hib \wedge f'(i) = e \wedge \\ \forall j \in [f.lob..f.hib] : j \neq i \to f'(j) = f(j). \end{array}$$

A sorozatokra bevezetett kiterjesztő és elhagyó műveleteket függvény típusra is definiáljuk:

$$\begin{array}{rcl} f := lorem(f) & \to & f : lorem \\ f := hirem(f) & \to & f : hirem \\ f := loext(f, e) & \to & f : loext(e) \\ f := hiext(f, e) & \to & f : hiext(e) \end{array}$$

Ha ezen utolsó csoportban felsorolt műveleteket egy függvénytípusra nem engedjük meg, akkor egy speciális függvénytípushoz, a vektorhoz jutunk. Az általános függvénytípustól megkülönböztetendő a vektortípusra külön jelölést vezetünk be: $V = vekt(H, E)$. Ha azt is jelölni kívánjuk, hogy mik a vektor indexhatárai, akkor a típust $V = vekt([ah..fh] : E)$-vel jelöljük. További jelölésbeli eltérés az általános függvénytípustól: a szelektorfüggvény jelölésére nem a kerek, hanem a szögletes zárójelet használjuk.