7. előadás

Programozás-elmélet

Egy program/algoritmus adatokat alakit át:

$$\text{input adatok} \longrightarrow \text{közbülső adatok} \longrightarrow \text{output adatok.}$$

Alapfogalom

- elemi adattípus
- összetett adattípus, vagy adatszerkezet. (Az összetett adattípusok némi egyszerűsítéssel bizonyos típusú adatok együttesei valamilyen összefüggéssel. A két alapfogalom között nincs éles határ (van amit ide-is, oda is sorolnak).

Megjegyzések:

1. Számos axiomatikus felépítés létezik, de nincs általánosan elfogadott.
2. Leginkább a program-, vagy metanyelvekhez kötődő leírásokat használjuk.
3. Az adattípusok, adatszerkezetek géptől és programnyelvtől függnek (implementálásfüggők).
4. Az adattípusokhoz hozzá kell érteni a velük végezhető műveleteket is.
5. Vannak alaptípusok és belőlük létrehozható bonyolultabb, un. strukturált, vagy összetett típusok. (Pascal nyelvvel kezdve). Ezek már adatszerkezeteknek tekinthetők.

Legyen L egy programnyelv, D pedig a nyelvben definiálható adatok halmaza. Tetszőleges $d \in D$ adatnak pontosan egy meghatározott típusa van. Ezért a D halmaz felbontható

$$D = D_{t_1} \cup D_{t_2} \cup \ldots$$

alakban, ahol $t \neq r$ esetén $D_t \cap D_r = \emptyset$ és $T = \{t_1, t_2, \ldots\}$ az L nyelvben megengedett típusok azonosítóiból álló halmaz. Eszerint az L nyelv t típusú adatainak halmaza

$$D_t = \{d \mid d \in D,\ típus(d) = t\}.$$

Legyen F az L programnyelv műveleteinek halmaza. Ezek a műveletek lehetnek aritmetikai, logikai, konverziós és egyéb műveletek. Minden $f \in F$ művelet egy

$$f : D_{t_1} \times D_{t_2} \times \ldots \times D_{t_n} \rightarrow D_{r_1} \times D_{r_2} \times \ldots \times D_{r_m}$$

alakú leképezés. Így a programnyelvet a (D, F) pár adja meg.

Adattipusok, adatszerkezetek megadására az alábbi közelítést választjuk:

1. Alap adattípusok.
2. Strukturált, vagy összetett adattípusok.
3. Mutató típus.

A strukturált, vagy összetett típusokat az egyszerű (és/vagy összetett) típusokból képezzük valamilyen konstrukciós elvvel.

Definíció

A $t_1, \ldots, t_n$ típusokból képezett $t \in T$ típust a

$$C : D_{t_1} \times D_{t_2} \times \ldots \times D_{t_n} \rightarrow D_t$$

konstrukciós függvény (konstruktor) és az

$$S_i : D_t \rightarrow D_{t_i} \quad (i = 1, \ldots, n)$$

szelekciós függvény jellemzi.

A $t \in T$ típusú adatokat a C leképezés értékkészlete adja, azaz

$$D_t = \{C(d_1, \ldots, d_n) \mid d_1 \in D_{t_1}, \ldots, d_n \in D_{t_n}\}.$$

Ha $d \in D_t$, akkor $S_i(d)$ adja meg a t-típusú adat t_i-típusú komponensét (összetevőjét). A C és S_i függvényeknek ki kell elégíteniük a következő szabályokat:

$$S_i(C(d_1, \ldots, d_n)) = d_i \quad (d_1 \in D_{t_1}, \ldots, d_n \in D_{t_n})$$

és

$$C(S_1(d), \ldots, S_n(d)) = d \quad (d \in D_t).$$

Legfontosabb adattípusok:

Elemi adattípusok

- Egyszerű (v. skalár, v. megszámozható) adattípus
- Szabványos elemi adattípusok
  - logikai (Pascal nyelvben: Boolean)
  - karakter (Pascal nyelvben: char)
  - egész (Pascal nyelvben: integer)
  - valós (Pascal nyelvben: real)

Struktúrált adattípusok

- tömb
- rekord
- egyesítés
- halmaz
- sorozat
- rekurzív típus (nem foglalkozunk vele)

A tömb típus a lineáris algebrából ismert vektor és mátrix fogalom megfelelője. Ennek megfelelően a tömb típus definíciója (konstrukciója) igen egyszerű.

Legyen T_0 egy már definiált típus. A

$$T_0^n = \underbrace{T_0 \times T_0 \times \ldots \times T_0}_{n}$$

direkt szorzat elemeit (rendezett elem n-eseket) tekintjük n (hosszúságú) T_0 tipusú tömböknek (vektoroknak). A T_0 típusú tömbök (adatok) száma: $|T_0|^n$.
A konstrukciós függény:

$$x_1, x_2, \ldots, x_n \in T_0 \longrightarrow (x_1, \ldots, x_n) \in T_0^n.$$

A szelektor:

$$(x_1, \ldots, x_n) \in T_0^n \xrightarrow{x[i]} x_i \in T_0.$$

A T_0 alaptípus maga is lehet tömb. Pl. kétdimenziós T_0 tipusú tömböt ($m \times n$ mátrixot) a T_0^n tömb m-szer önmagával vett direkt szorzatával képezhetünk:

$$T_0^{m \times n} = \underbrace{T_0^n \times T_0^n \times \ldots \times T_0^n}_{m}.$$

Ennél a definíciónál némileg bonyolultabb a következő - a Pascal nyelvhez kötődő - definíció.

Legyen I a valós, vagy egész típust kivéve egy megszámozható, vagy részintervallum típus, T_0 pedig tetszőleges alaptípus. Az I indexhalmazú T_0 típusú tömb Pascal definíciója:

$$\textbf{type } T = \textbf{array } I \textbf{ of } T_0$$

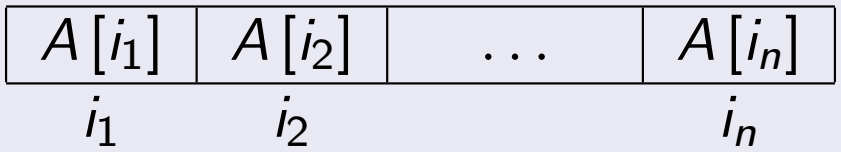
Itt a T típusú A tömböt az

$$A : D_I \to D_{T_0}$$

leképezéssel, pontosabban ennek értékkészletével azonosítjuk. Tehát a T típus az $D_I \to D_{T_0}$ leképezések halmazaként is felfogható. Legyen $I = \{i_1, i_2, \ldots, i_n\}$. Ekkor az A leképezés értékkészlete

$$\{A[i_1], A[i_2], \ldots, A[i_n]\},$$

ahol $A[i]$ az A értéket jelöli az $i \in I$elemen. Az A tömböt az



alakban is ábrázolhatjuk.

Tekintve, hogy I rendezett és megszámlálható, a T tömb típust azonosíthatjuk az $T_0^{|I|}$ halmazzal is, ami éppen a tömb első meghatározásának felel meg.

Példák

type $A =$**array** $[1..20]$ **of** *integer*
type $alkalmazott = (Tanaka, Nakamura, Kato, Yamamoto)$;
$jovedelem =$ **array** $[alkalmazott]$ **of** *integer*

Legyen $a_k = A[i_k]$ $(k = 1, \ldots, n)$. Ekkor a T típusú A tömböt az

$$A = T(a_1, \ldots, a_n) = T(A[i_1], \ldots, A[i_n]) = (A[i_1], \ldots, A[i_n])$$

formában írhatjuk fel. A $T : D_{T_0^n} \to D_T$ függvényt *tömbkonstruktornak* nevezzük.

A konstruktor inverz művelete a *szelektor*, amely a tömb egyes komponenseit választja ki:

$$A[i]$$

az $i \in I$ komponensnek megfelelő érték, $[i] : D_T \to D_{T_0}$ indexfüggvény,

$$T(a_1, \ldots, a_n)[i] = a_i.$$

Az i index kifejezés kiértékelésével is megkapható. Adott i index esetén az i-edik komponens értékét megváltoztathatjuk:

$$A[i] := a$$

ahol a egy T_0 típusú adat.

Példa

A *jovedelem* típus esetén a *Kato* indexű elem/komponens: $A[Kato]$.

Tömbtípusok összetevői maguk is lehetnek strukturáltak. Egy tömbváltozót, amelynek komponensei szintén tömbök, *mátrixnak* nevezünk. Legyen $I_1, I_2, \ldots, I_n$ a valós, vagy egész típust kivéve egy megszámozható, vagy részintervallum típus, T_0 pedig tetszőleges alaptípus. Az n dimenziós tömb típust a

$$\textbf{type } T = \textbf{array } [I_1] \textbf{ of array } [I_2] \textbf{ of } \ldots \textbf{ of array } [I_n] \textbf{ of } T_0$$

vagy az ekvivalens

$$\textbf{type } T = \textbf{array } [I_1, I_2, \ldots, I_n] \textbf{ of } T_0$$

előírás definiálja.
Ha A egy T típusú tömb, akkor az

$$A : D_{I_1} \times D_{I_2} \times \ldots \times D_{I_n} \rightarrow D_{T_0}$$

leképezés értékkészletével azonosítjuk.

Az A tömb egy elemét $A[i_1, i_2, \ldots, i_n]$ formában adjuk meg, ahol $i_1 \in I_1, i_2 \in I_2, \ldots, i_n \in I_n$. Ez ugyanaz mint a szelektorok egymásután fűzésével kapott alak: $A[i_1][i_2]\ldots[i_n]$.

Példák

$T =$**array** $[1..10]$ **of array** $[1..5]$ **of** *real*
$T =$**array** $[1..10, 1..5]$ **of** *real*

Az elnevezés eredete: $A = [a_{ij}]_{i,j=1}^{m,n}$ mátrix szokásos alakja:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix},$$

ahol a_{ij} az (i, j) indexű elem. Az $A =$**array** $[1..m, 1..n]$ **of** T_0 tömb ezzel analóg.

A rekord típus definíciója (konstrukciója) is egyszerű. Rekordoknak a

$$T = \underbrace{T_1 \times T_2 \times \ldots \times T_n}_{n}$$

direkt szorzat elemeit (rendezett elem n-eseket) tekintjük. A T_i típushalmazok között lehetnek azonosak is, de az egyes komponenseket (mezőket) különbözőnek tekintjük.

A konstrukciós függény:

$$x_1 \in T_1, x_2 \in T_2, \ldots, x_n \in T_n \longrightarrow (x_1, \ldots, x_n) \in T.$$

A szelektor:

$$(x_1, \ldots, x_n) \in T \xrightarrow{x[i]} x_i \in T_i.$$

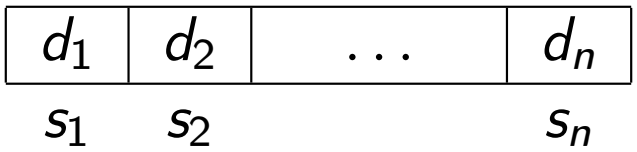
Legyen $T_1, T_2, \ldots, T_n$ típusazonosító, $s_1, s_2, \ldots, s_n$ pedig névazonosítók. A $T_1, \ldots, T_n$ típusokból álló rekord típus definíciója:

$$\begin{array}{lll} \textbf{type } T = \textbf{record} & s_1 : T_1; \\ & s_2 : T_2; \\ & \ldots \\ & s_n : T_n \\ \multicolumn{3}{c}{\textbf{end}} \end{array}$$

Az $s_1, \ldots, s_n$ azonosítók a rekord komponensek (mezők)nevei. A T típusú rekordok halmaza

$$D_T = D_{T_1} \times D_{T_2} \times \ldots \times D_{T_n}.$$

Minden $d \in D_T$ felírható $d = (d_1, \ldots, d_n)$ alakban, ahol $d_i \in D_{T_i}$. Ezt a tömbhöz hasonlóan ábrázolhatjuk:



Példák

type *ido* =**record** *ora* : 1..24
perc : 1..60
sec : 1..60
end

Megjegyzés: *T_i lehet összetett típusú is.*

Legyen $d_i \in D_{T_i}$ $(i = 1, \ldots, n)$. Ekkor a megfelelő T típusú d rekordot a

$$d = T(d_1, d_2, \ldots, d_n) = (d_1, d_2, \ldots, d_n)$$

konstruktor függvény adja meg. A rekord egyes komponenseit (mezőit) az

$$.s_i : D_T \to D_{T_i} \quad (i = 1, \ldots, n)$$

szelektorfüggvények adják meg:

$$d.s_i = d_i$$

Teljesülnek a következő feltételek:

$$T(d_1, d_2, \ldots, d_n).s_i = d_i, \quad T(d.s_1, d.s_2, \ldots, d.s_n) = d.$$

Az s_i mező értékadása:

$$d.s_i := d_i$$

Példák

$x.ora = 11;$

$x.perc := 20;$

$x.\,\mathrm{sec} := 35$

Megjegyzés

A tömb és rekord közötti azonosságok és különbségek:

1. Tömb esetén $T_1 = T_2 = \cdots = T_n = T_0$
2. A tömb indexei rendezett tipusúak és így adott sorrendben feldolgozhatók. A rekord komponensei rendezetlenek és nem formálnak egy adott típust.
3. A tömb és a rekord asszociatív adatszerkezetek.

Legyen A és B két nemüres halmaz. Az A és B halmazok diszjunkt egyesítése

$$A \oplus B = \{(s, 1) \mid s \in A\} \cup \{(s, 2) \mid s \in B\}.$$

Legyen T_1 és T_2 két adattípus, t_1 és t_2 a típusok azonosítói. A két típus T egyesítése a $t_1 : d_1$ és $t_2 : d_2$ ($d_1 \in D_{T_1}$, $d_2 \in D_{T_2}$) párokból áll. Formális deklarációja:

$$\begin{aligned} \textbf{type } T \quad = \textbf{union} \quad & t_1 : T_1; \\ & t_2 : T_2 \\ \textbf{end} & \end{aligned}$$

Tehát

$$D_T = \{t_1 : d_1 \mid d_1 \in D_{T_1}\} \cup \{t_2 : d_2 \mid d_2 \in D_{T_2}\}.$$

A T egyesített típus egy elemét a következőképpen ábrázolhatjuk:

t_i	d_i

Példa:

$$
\begin{aligned}
\textbf{type } \mathit{coordinate1} &= \textbf{record} \\
&\quad x, t : \mathit{real} \\
&\textbf{end} \\
\textbf{type } \mathit{coordinate2} &= \textbf{record} \\
&\quad r : \mathit{real}; \\
&\quad \theta : \mathit{angle} \\
&\textbf{end} \\
\textbf{type } \mathit{coordinate} &= \textbf{union} \\
&\quad \mathit{Cartesian} : \mathit{coordinate1}; \\
&\quad \mathit{Polar} : \mathit{coordinate2} \\
&\textbf{end}
\end{aligned}
$$

Az egyesítés egy lehetséges Pascal megvalósítása:

$$
\begin{array}{lll}
\textbf{type} & \mathit{ckind} & = (\mathit{Cartesian}, \mathit{Polar}) \\
 & \mathit{coordinate} & = \textbf{record}\ \mathit{ckind}\textbf{of} \\
 & & \qquad \mathit{Cartesian} : (x, y : \mathit{real})\,; \\
 & & \qquad \mathit{Polar} : (r : \mathit{real}; \theta : \mathit{angle}) \\
 & & \textbf{end}
\end{array}
$$

Megjegyzés:

Az egyesítés kettőnél több komponensre is kiterjeszthető. Az union szerkezet nem Pascal utasítás. A Pascalban a szerkezetet váltakozó rekordnak hívják (case utasítással valósítják meg).

Legyen $d \in D_T$. Két eset lehetséges: d vagy $t_1 : d_1$ ($d_1 \in D_{T_1}$), vagy $t_2 : d_2$ ($d_2 \in D_{T_2}$). Ennek megfelelően a T konstrukciós függvényt két részletben adhatjuk meg: $T = t_1$, vagy $T = t_2$, ahol $t_i : D_{T_i} \to D_T$ ($i = 1, 2$). A $d_i \in D_{T_i}$ értékhez a $t_i : d_i \in D_T$ pár kerül hozzárendelésre. A

$$.t_i : D_T \to D_{T_i} \quad (i = 1, 2)$$

szelektor függvényeket a következőképpen adhatjuk meg:

$$d.t_1 = \begin{cases} d_1, & \text{ha } d = t_1 : d_1 \\ \text{definiálatlan egyébként} \end{cases} \qquad d.t_2 = \begin{cases} \text{definiálatlan egyébként} \\ d_2, & \text{ha } d = t_2 : d_2 \end{cases}$$

A fenti definíciókkal teljesülnek a

$$d.t_i = (t_i : d_i).t_i = d_i \quad (d_i \in D_{T_i})$$

$$t_i : (d.t_i) = t_i : d_i = d \quad (d \in D_T, d.t_i \text{ definiált})$$

feltételek.

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

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

A T_0 típus részhalmazai a

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

hatványhalmazt alkotják.

Példa:

type T =**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 **in** B

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