Programovanie a algoritmy - tretí týždeň

Gabriela Andrejková

e-mail: andrejk@kosice.upjs.sk

October 6, 2002

Obsah

0.1 Syntaktické diagramy na popis jazyka 2
0.2 Základné prvky jazyka 5
0.2.1 Abeceda a lexikálne symboly 5
0.2.2 Jednoduché typy údajov 6

0.1 Syntaktické diagramy na popis jazyka

Syntaktický diagram (nazývaný tiež Conwayov diagram) vyjadruje väzbu medzi jednotlivými symbolmi syntaxe jazyka, určenú vytvárajúcim pravidlom. Ak je daná syntax jazyka, tak syntaktický diagram je možné vytvárať nasledujúcim spôsobom:

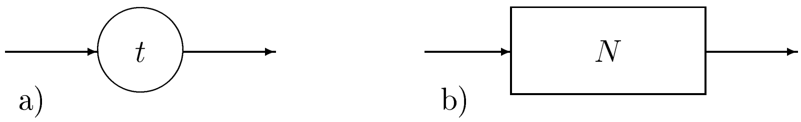
1. Každý neterminálny symbol (tj. symbol, ktorý je v jazyku ešte upravovaný) A, ktorý je definovaný vytvárajúcim pravidlom gramatiky

   $$A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$$

   je zobrazený v tvare syntaktického diagramu $G(A)$ pomocou nasledujúcich pravidiel 2.-5. Štruktúra syntaktického diagramu je určená pravou stranou vytvárajúceho pravidla.

2. Každému výskytu terminálneho alebo neterminálneho symbolu v α_i zodpovedá v syntaktickom diagrame vrchol ohodnotený daným symbolom. Sú zobrazované nasledovne:

   - terminálny symbol t tak, ako je uvedené na Obr. 1,
   - neterminálny symbol N tak, ako je uvedené na Obr. 1, pričom $\to$ vyjadruje väzbu medzi symbolmi.

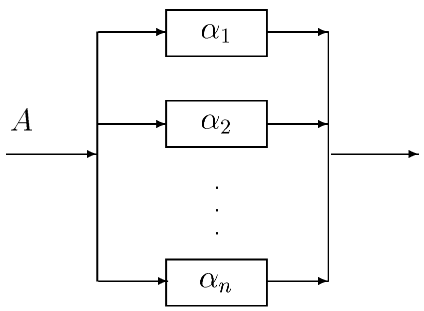


Obrázok 1: Syntaktický diagram a) terminálneho symbolu t a b) neterminálneho symbolu N.

3. Vytvárajúce pravidlo tvaru

$$A \to \alpha_1|\alpha_2|\dots|\alpha_n$$

sa zobrazí v tvare syntaktického diagramu tak, ako je znázornené na Obr. 2, pričom každé $\alpha_i, i = 1, 2, \dots, n$ je možné získať pomocou pravidiel 2.-5. To znamená, že terminálny symbol je definovaný svojím výskytom a pre zobrazenie jeho výskytu v diagrame použijeme zobrazenie podľa Obr. 1 a). Ak je to neterminálny symbol, zodpovedá mu vytvárajúce pravidlo, ktorým je daný symbol definovaný a podľa jeho pravej strany vytvorí preň syntaktický diagram. Syntaktický diagram k tomuto vytvárajúcemu pravidlu sa nachádza na Obr. 2.

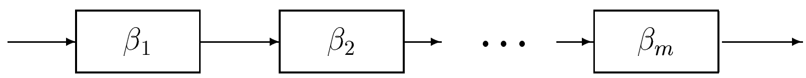


Obrázok 2: Syntaktický diagram pre pravidlo $A \to \alpha_1|\alpha_2|\dots|\alpha_n$.

4. Ak α_i má tvar

$$\alpha_i = \beta_1\beta_2\dots\beta_m$$

je zobrazené v tvare syntaktického diagramu na Obr. 3, pričom každé $\beta_i, i = 1, 2, \dots, m$ je získané pomocou pravidiel 2.-5.

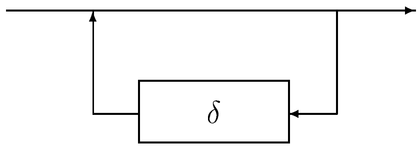


Obrázok 3: Syntaktický diagram pre pravidlo $\alpha_i = \beta_1\beta_2\dots\beta_m$.

5. Ak α_i má tvar

$$\alpha_i = \{\delta\}$$

je zobrazené v tvare syntaktického diagramu na Obr. 3, pričom δ je získané pomocou pravidiel 2.-5.



Obrázok 4: Syntaktický diagram pre pravidlo tvaru $\alpha_i = \{\delta\}$.

0.2 Základné prvky jazyka

0.2.1 Abeceda a lexikálne symboly

Abeceda je množina znakov, ktoré sa môžu vyskytovať v zdrojovom texte programu. Na písanie programov v jazyku Turbo Pascal sa používa štandardná množina znakov kódu ASCII (American Standard Code for Information Interchange).

Lexikálne symboly sú najmenšie významové jednotky v texte programu a delíme ich na:

1. Identifikátory
2. Číselné konštanty
3. Znakové konštanty
4. Špeciálne symboly a vyhradené slová
5. Návestia

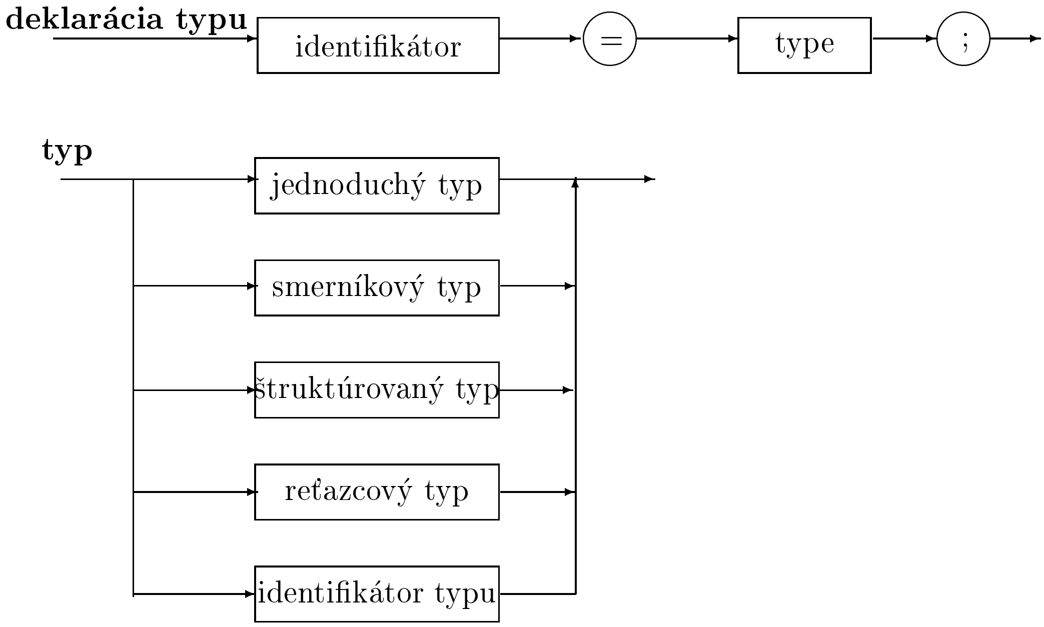
Program pozostáva z lexikálnych symbolov, ktoré sú oddelené oddeľovačmi: **medzerou** a **komentárom**.

Identifikátory slúžia na symbolické pomenovania objektov ako sme to už uviedli v jazyku nášho procesora v 1. kapitole.

Vyhradené slová: and, asm, array, begin, case, class, const, constructor, destructor, div, do, downto, else, end, external, file, for, forward, function, goto, if, implementation, in, inline, interface, interrupt, label, mod, not, nil, of, or, procedure, program, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, while, with, xor, atď.

0.2.2 Jednoduché typy údajov

Keď deklarujeme premennú, určujeme aké údaje bude obsahovať, t. j. určujeme jej typ. Len vo výnimočných prípadoch sa môže vyskytnúť premenná, ktorá nemá definovaný typ, tzv. netypová premenná.



Obrázok 5: Syntaktický diagram a) deklarácie typu a b) typu.

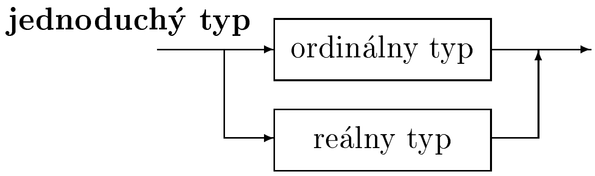
Typ premennej popisuje

1. **množinu hodnôt, ktoré by mohla táto premenná nadobudnúť** (vždy jednu hodnotu z tejto množiny);

2. **operácie, ktoré je možné vykonávať s hodnotami uloženými v tejto premennej.**

Deklarácia typu určuje identifikátor, ktorý sa stáva menom tohto typu (označuje tento typ).

Jednoduché typy sú definované usporiadanými množinami hodnôt, t. j. sú to *ordinálne typy* a typy reálne.



Obrázok 6: Syntaktický diagram pre jednoduchý typ

Napríklad, reálny typ je predstavovaný jedným z nasledujúcich štandartných identifikátorov: **Real, Single, Double, Extended, alebo Comp**.

Ordinálne typy

Ordinálne typy sú podmnožiny jednoduchých typov okrem reálneho typu. Ordinálne typy splňajú nasledujúce charakteristiky:

1. Všetky hodnoty daného ordinálneho typu tvoria usporiadanú množinu a každá možná hodnota je asociovaná s poradím (ordinalitou) v tejto množine.

   Poradie je vlastne celé číslo. Okrem poradia u typu **Integer** prvá hodnota každého ordinálneho typu má poradie **0**, ďalšia má poradie **1**, atď.

   Poradie hodnôt u typu Integer je určené hodnotou samotnou. V ľubovoľnom ordinálnom type, každý prvok rôzny od prvého má svojho predchodcu a každý prvok rôzny od posledného má svojho nasledovníka.

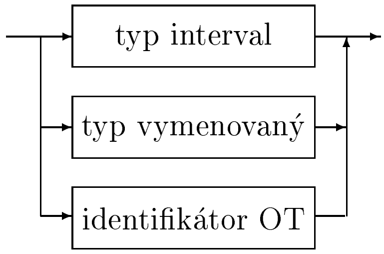
2. Štandardná funkcia **Ord** môže byť použitá na ľubovoľnú hodnotu ordinálneho typu a jej výsledkom je poradové číslo. Štandardná funkcia **Pred** môže byť použitá na

   ľubovoľnú hodnotu ordinálneho typu a jej výsledkom je predchádzajúca hodnota. V prípade použitia tejto funkcie na prvý prvok, vznikne chyba. Štandardná funkcia

   Succ môže byť použitá na ľubovoľnú hodnotu ordinálneho typu a jej výsledkom je nasledujúca hodnota. V prípade použitia tejto funkcie na posledný prvok, vznikne chyba.

Syntax ordinálneho typu je nasledovná:

ordinálny typ



Obrázok 7: Syntaktický diagram pre ordinálny typ

Pascal má päť preddefinovaných celočíselných typov: **ShortInt, Integer, LongInt, Byte a Word**. Každý z nich predstavuje určitú podmnožinu množiny celých čísel podľa nasledujúcej tabuľky:

Typ	Rozsah	**Format**
ShortInt	-128..127	8 bitov (znamienko)
Integer	-32768..32767	16 bitov (znamienko)
LongInt	-2147483648..2147483647	32 bitov (znamienko)
Byte	0..255	8 bitov (bez znamienka)
Word	0..65535	16 bitov (bez znamienka)

Tabuľka 1: Preddefinované celočíselné typy.

Aritmetické operácie s operandami celočíselného typu spolu s typmi výsledkov sú uvedené v nasledujúcej tabuľke:

Operátor	**Operácia**	**Typy operandov**	**Typ výsledku**
+	sčítanie	celočíselné	celočíselný
-	odčítanie	celočíselné	celočíslený
*	násobenie	celočíselné	celočíselný
/	delenie	celočíselné	reálny
div	celočíselné delenie	celočíselné	celočíselný
mod	zvyšok po delení	celočíselné	celočíselný

Tabuľka 2: Aritmetické operácie na celočíselných typoch.

Vykonávanie sa aritmetických operácií prebieha na základe nasledujúcich pravidiel:

- Pre binárne operácie sú oba operandy prevedené (konvertované) na **spoločný typ** ešte pred operáciou.

  Tento spoločný typ je určený ako celočíselný typ najmenšieho rozsahu, ktorý zahŕňa všetky možné hodnoty oboch typov.

  Napríklad, spoločný typ pre Integer a Byte je Integer, spoločný typ pre Integer a Word je LongInt.

  Operácia je vykonaná použitím presnosti spoločného typu a typ výsledku je spoločný typ.

- Výraz na pravej strane priraďovacieho príkazu je vyhodnocovaný nezávisle od veľkosti alebo typu premennej na ľavej strane.

- Ľubovoľný operand veľkosti jedného bytu je konvertovaný na operand veľkosti slova, ktorý je kompatibilný s oboma Word aj Integer, pred vykonaním ľubovoľnej aritmetickej operácie.

Typ Boolean

Hodnoty typu Boolean sú určené dvoma konštantami **True** a **False**. Pretože Boolean je vymenovaný typ, platia nasledujúce vlastnosti:

- $False < True$,
- $Ord(False) = 0, Ord(True) = 1$,
- $Succ(False) = True, Pred(True) = False$

Operácie na tomto type: S štyri logické operátory - **and, xor, or, not**. Pracujú s logickými hodnotami (true, false) a umožňujú kombinovať relačné výrazy (ich výsledkom je logická hodnota), booleovské premenné a booleovské výrazy.

Typ Char

Tento typ je tvorený množinou znakov usporiadaných podľa rozšírenej tabuľky ASCII kódu pre znaky.

Funkcia **Ord(Ch)** vráti výsledok, ktorý predstavuje poradové (ordinálne) číslo znaku **Ch**. Inverzná funkcia k tejto funkcii je **Chr**.

Reálne typy

Tieto typy predstavujú podmnožiny reálnych čísel, ktoré môžu byť reprezentované v pohyblivej rádovej čiarke pri pevnom počte číslic. Zobrazenie v pohyblivej rádovej čiarke

normálne ukladá tri hodnot – M, B a E – také, že platí $M \times B^E = N$, kde $B = 2$ a M a E sú celočíselné hodnoty obsiahnuté v reálnom type.

Hodnoty M a E ďalej určujú rozsah a presnosť reálneho typu.

Je päť druhov preddefinovaný reálnych typov: **Real, Single, Double, Extended, Comp**. Ich rozsahy sa nachádzajú v tabuľke 0.2.2.

Typ	**Rozsah**	**Signifikantné číslice**	Veľkosť v bytoch
Real	$2.9 \times 10^-39..1.7 \times 10^{38}$	11-12	6
Single	$1.5 \times 10^{-45}..3.4 \times 10^{38}$	7-8	4
Double	$5.0 \times 10^{-324}..1.7 \times 10^{308}$	15-16	8
Extended	$3.4 \times 10^{-4932}..1.1 \times 10^{4932}$	19-20	10
Comp	$-2^{63} + 1..2^{63} - 1$	19-20	8

Tabuľka 3: Preddefinované reálne typy.