

```

S_NS: if symbol.typ = S_ENDOFFILE then reduce(0)
      else error('očekáván konec souboru');
S_NV: case symbol.typ of
      S_PLUS,S_MINUS,S_RZAV: push;
      S_ENDOFFILE: reduce(1);
      else error('symbol ',VypisHodn(symbol),' není očekávaného typu ',
                VypisTyp(S_PLUS),' ',VypisTyp(S_MINUS),' ',VypisTyp(S_RZAV),
                ' nebo konec souboru');
      end;
S_NA: if symbol.typ in [S_NUM,S_ID,S_LZAV] then reduce(9)
      else error('symbol ',VypisHodn(symbol),' není očekávaného typu ',
                VypisTyp(S_NUM),' ',VypisTyp(S_ID),' nebo ',VypisTyp(S_LZAV));
... // atd. pro všechny neterminály
S_NUM: if symbol.typ in [S_PLUS,S_MINUS,S_MUL,S_DIV,S_RZAV,S_ENDOFFILE]
      then reduce(10)
      else error(...); // ošetření chyby podobně jako v předchozím kódu,
S_ID: case symbol.typ of // to platí i dále
      S_PLUS,S_MINUS,S_MUL,S_DIV,S_RZAV,S_ENDOFFILE: reduce(11);
      S_ROVNASE: push;
      else error(...);
      end;
S_ROVNASE: if symbol.typ in [S_NUM,S_ID,S_LZAV] then reduce(5)
           else error(...);
... // atd. pro všechny terminály
           else error(...);
           end;
end;
end;

procedure Analyza;
begin
  Init;
  while (not Konec) do Akce;
  Done;
end;

```

---

## Úkoly ke kapitole 5

1. Překládovou gramatiku z příkladu 5.3 na straně 122 obohatte o operátory odčítání (priorita stejná jako sčítání), dělení a celočíselného dělení div (priorita operátoru stejná jako u operátoru násobení).
2. Sestrojte konečný překládový automat (přímo, bez konstrukce gramatiky), který dokáže dělit binární čísla třemi. Předpokládejte, že na vstupu je vždy číslo dělitelné třemi.

*Nápověda:* nejdřív vyzkoušejte dělení binárních čísel „ručně“. Zbytek po dělení je po každém kroku některé z čísel 0, 1 nebo 2, použijte stavy N, J a D pro uložení této informace. Automat reaguje na momentální stav (tj. N, J nebo D) a vstupní symbol (0 nebo 1), akcí je přechod do některého ze stavů (podle momentálního zbytku po dělení) a výpisu číslice na výstup.

3. Podle následující regulární překladové gramatiky (překládající některé prvky syntaxe PASCALU do C) sestrojte konečný překladový automat reprezentovaný  $\delta$ -funkcí i přechodovou tabulkou.

$RPG = (\{S\}, \{\langle begin \rangle, \langle end \rangle, x\}, \{\textcircled{1}, \textcircled{0}, \textcircled{x}\}, R, S)$ , kde  $x$  je kterýkoliv ze symbolů ve vstupní abecedě přímo neuvedených a v množině  $R$  jsou tato pravidla:

$$S \rightarrow \langle begin \rangle \textcircled{1} S \mid \langle end \rangle \textcircled{1} S \mid x \textcircled{x} S \mid \langle end \rangle \textcircled{1}$$

4. Zjistěte, zda následující překladová gramatika je  $LL(1)$  – její vstupní gramatiku najdete v úkolu 4 na straně 94. Pokud je to  $LL(1)$  překladová gramatika, sestrojte k ní překladový automat reprezentovaný rozkladovou tabulkou a podle této tabulky zpracujte (přeložte) jakékoliv slovo z jazyka vstupní gramatiky delší než 4.

$PG = (\{S, A, B, C\}, \{a, b, c, d\}, \{\textcircled{a}, \textcircled{b}, \textcircled{c}, \textcircled{d}\}, R, S)$ , pravidla jsou

$$S \rightarrow \textcircled{a} abA\textcircled{b} \mid cbBC\textcircled{c}\textcircled{b}$$

$$A \rightarrow cA\textcircled{c} \mid dB\textcircled{b}\textcircled{b} \mid a\textcircled{a}$$

$$B \rightarrow bB\textcircled{b}S \mid a\textcircled{a}\textcircled{a}$$

$$C \rightarrow cC\textcircled{c} \mid \textcircled{c}cBd\textcircled{d} \mid \varepsilon$$

5. Je dána překladová gramatika  $PG = (\{A, B\}, \{a, b\}, \{x, y\}, R, A)$  s pravidly v  $R$

$$A \rightarrow aAy \mid bxB$$

$$B \rightarrow bxB \mid bx$$

Zjistěte, jaký překlad generuje tato gramatika a sestrojte překladový automat reprezentovaný  $\delta$ -funkcí.

6. Následující atributová gramatika generuje deklarace v syntaxi jazyka C a deklarované proměnné přidává do tabulky symbolů.

$$D \rightarrow T \{S.d\_typ = T.typ\} S;$$

$$T \rightarrow \langle int \rangle \{T.typ = integer\}$$

$$T \rightarrow \langle float \rangle \{T.typ = float\}$$

$$T \rightarrow \langle char \rangle \{T.typ = char\}$$

$$S \rightarrow i \{PridejDoTab(i.nazev, S.d\_typ), M.d\_typ = S.d\_typ\} M$$

$$M \rightarrow , \{S.d\_typ = M.d\_typ\} S$$

$$M \rightarrow \varepsilon$$

Vytvořte derivační strom věty  $\langle int \rangle x, y, prom;$  s vyznačením toku hodnot atributů. Zjistěte, zda je gramatika typu  $LL(1)$  a promyslete si způsob implementace rekurzivním sestupem – sémantické funkce, které je třeba naprogramovat, dále parametry funkcí podle neterminálů volané hodnotou nebo odkazem a kód těchto funkcí.

7. Následující bezkontextová gramatika popisuje deklarace proměnných v PASCALovské syntaxi.

$$\begin{aligned} D &\rightarrow S : T; \\ T &\rightarrow \langle int \rangle \mid \langle real \rangle \mid \langle char \rangle \\ S &\rightarrow iM \\ M &\rightarrow , S \mid \varepsilon \end{aligned}$$

Přidejte sémantická pravidla tak, aby gramatika plnila podobnou funkci jako gramatika se syntaxí jazyka C v předchozím úkolu (tj. přidává deklarované proměnné do tabulky symbolů).

*Nápověda:* protože se vstup vyhodnocuje zleva doprava, na místě uvedení názvu proměnné (terminál  $i$ ) na vstupu neznáme její datový typ a proto nemůžeme do tabulky symbolů přidat oba údaje najednou; tento problém můžeme vyřešit například tak, že na místě uvedení názvu proměnné přidáme do tabulky symbolů záznam s datovým typem, který dokážeme snadno vyhledat (například  $\langle nedef \rangle$ ), a pak v pravidle, kdy konečně zjistíme datový typ proměnné, projdeme celou tabulku symbolů a ve všech záznamech s datovým typem  $\langle nedef \rangle$  tento typ zaměníme za zjištěný. Jinou možností je dočasné uschování názvů proměnných ve spojovém seznamu.

8. Promyslete si implementaci atributové gramatiky, kterou jste vytvořili v úkolu 7, pro interpretační překladač. Pro datové typy vytvořte typy symbolů stejně jako například pro identifikátory (názvy proměnných), dvojtečku, čárku a středník, pro prvky v tabulce symbolů pak použijte záznamy obsahující tyto tři údaje:

- název proměnné (typu řetězec),
- datový typ proměnné (typu `TTypSymbolu`),
- hodnotu proměnné (použijte variantní záznam nebo union).

Tabulku symbolů implementujte jako spojový seznam výše popsaných záznamů. Vytvořte přístupové funkce k tabulce (vyhledání proměnné, zjištění a změnu hodnoty proměnné, přidání nové proměnné a její odstranění, vhodným způsobem zajistěte ošetření chyb.

9. Je dána gramatika popisující syntaxi seznamu čísel:

$$\begin{aligned} S &\rightarrow L \\ L &\rightarrow nA \\ A &\rightarrow , L \mid ; \end{aligned}$$

Přidejte sémantická pravidla tak, aby

- (a) v atributu  $S.soucet$  byl součet všech čísel v seznamu.
- (b) v atributu  $S.max$  byla hodnota nejvyššího prvku seznamu.

- (c) v atributu *n.index* každého prvku seznamu bylo pořadové číslo tohoto prvku (začínáme zleva indexem 1).

10. Je dána následující překladová gramatika:

$$\begin{aligned} Z &\rightarrow Sv \\ S &\rightarrow ( A \mid ca \\ A &\rightarrow a ) \mid [ cSb ] \mid \langle B \mid \varepsilon \\ B &\rightarrow b \rangle \mid S \mid \varepsilon \end{aligned}$$

Přidejte sémantická pravidla tak, aby v atributu výstupního terminálu

- (a) *v.leve* byl počet všech levých závorek ve vygenerovaném výrazu (tj. závorek (, [ a ⟨).
- (b) *v.kulate* byl počet všech kulatých závorek (levých i pravých) ve vygenerovaném výrazu.
- (c) *v.n* byl počet všech (vstupních terminálních) symbolů ve vygenerovaném výrazu, které nejsou (jakýmkoliv) závorkami, tedy symbolů *a*, *b* a *c*.
- (d) *v.eps* byl počet použití  $\varepsilon$ -pravidla během derivace (nezapomeňte na inicializaci syntetizovaného atributu ve všech terminálních pravidlech).
11. K dané překladové gramatice přidejte sémantická pravidla tak, aby při interpretaci docházelo k vyhodnocování logických výrazů v uvedené syntaxi.

$$\begin{aligned} V &\rightarrow Sv \\ S &\rightarrow A \langle or \rangle S \mid A \\ A &\rightarrow B \langle and \rangle A \mid B \\ B &\rightarrow (S) \mid C < C \\ C &\rightarrow i \mid n \end{aligned}$$

Použijte atributy  $V[ ], v[vysl], S[log], A[log], B[log], C[val], i[name], n[lex]$ . Určete, které jsou dědičné a které syntetizované. Dále použijte vhodnou sémantickou funkci pro zjištění hodnoty proměnné z tabulky symbolů a pro rozhodování příkaz typu if.

12. Zjistěte, zda je daná překladová gramatika typu  $LL(1)$ . Pokud ano, přidejte sémantická pravidla tak, aby se funkčně shodovala s atributovou překladovou gramatikou, kterou jste vytvořili v úkolu 11 a naprogramujte metodou rekurzivního sestupu.

$$\begin{aligned} V &\rightarrow Sv \\ S &\rightarrow AD \\ D &\rightarrow \langle or \rangle S \mid \varepsilon \\ A &\rightarrow BE \\ E &\rightarrow \langle and \rangle A \mid \varepsilon \\ B &\rightarrow (S) \mid C < C \\ C &\rightarrow i \mid n \end{aligned}$$