COMBINED PARSING

Jiří ZUZAŇÁK, Bachelor Degree Programme (3)
Dept. of Information Systems, FIT, BUT
E-mail: xzuzan02@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This article deals with ideas, which could improve efficiency and eliminate difficulties of implementation of syntactic analysis by dividing entire syntactic analyzer in two parts communicating with each other. First part is parsing input code by syntactic "top-down" method and second is parsing input code by syntactic "bottom–up" method. The first method is supposed to parse input code for program flow description and the second one is used to parse mathematical, logical and other similar expressions.

1 ÚVOD

Syntaktická analýza je důležitou součástí interpretů a překladačů. Smyslem projektu je navrhnout metodu syntaktické analýzy založenou na využití několika metod syntaktické analýzy shora dolů a zdola nahoru. Každá část navrženého analyzátoru bude zpracovávat část kódu, pro kterou bude nejvhodnější.

2 ROZBOR

Navržený syntaktický analyzátor se skládá ze dvou částí. První realizuje rozklad shora dolů (top–down) a druhá realizuje rozklad zdola nahoru (bottom–up). Je nutné definovat typ analýzy, která bude řídit celý proces rozkladu. Řídící analýza bude zpracovávat vstupní kód jí určený a pokud narazí na úsek kódu, který by měl být zpracován podřízenou analýzou, spustí algoritmus její syntaktické analýzy, dále jen analyzátor.

Jako řídící jsem navrhl analýzu shora dolů. Ta bude zpracovávat kód reprezentující řízení toku programu. Podřízenou analýzou, která bude zpracovávat výrazy matematické, logické, atd. bude analýza zdola nahoru. Vyvstává problém, jak řídící analyzátor pozná, že následující kód je určen pro rozklad zdola nahoru. Dalším úkolem je vyřešit spojení rozkladů kódů z jednotlivých analýz.

K realizaci jednotlivých analyzátorů budou použity standardní algoritmy. Pro analýzu shora dolů je založený na rozkladové tabulce. Která se skládá ze dvou akcí, přepisu pravidlem a vyzvednutí symbolu ze zásobníku. Pro rozklad zdola nahoru bude využit algoritmus založený na dvou tabulkách, tabulce akcí a tabulce přechodů. Akce se budou skládat z shift a reduce. Přechody budou popisovat funkci použitou při redukci podle pravidel.

2.1 JAZYK

Pro příklady a jednodušší pochopení zavedu jazyk popsaný níže uvedenými pravidly. Pravidla 1 až 3 jsou určená pro rozklad výrazů, zatímco 4 až 7 jsou používána pro rozklad příkazů řídících tok programu. Následuje i přepis terminálů a nonterminálů na znakovou reprezentaci.

```
1. <expr> → <expr> + <expr>            <start>    - S
2. <expr> → <expr> - <expr>            <cmd-list> - L
3. <expr> → id                         <cmd>      - C
4. <start> → begin <cmd-list>          <expr>     - E
5. <cmd-list> → end                    begin      - b
6. <cmd-list> → <cmd> ; <cmd-list>     end        - e
7. <cmd> → write <expr>                read       - r
8. <cmd> → read id                     write      - w
                                       id         - p
```

2.2 SPUŠTĚNÍ A UKONČENÍ ANALÝZY ZDOLA NAHORU

K účelu spuštění analýzy zdola nahoru jsem definoval množinu *startLR*. Tato množina obsahuje část symbolů ze vstupní abecedy, které se mohou vyskytovat na začátku výrazu rozkládaného analýzou zdola nahoru. V případě navrženého jazyka je to pouze symbol *id* (tj. *p*). Když analyzátor shora dolů bude na vstupu očekávat výraz (tj. na vrcholu zásobníku bude symbol *E*) a na vstupu bude symbol obsažený v množině *startLR*, dojde ke spuštění analýzy zdola nahoru. Po ukončení analýzy zdola nahoru se řízení vrátí analyzátoru shora dolů. Ten odstraní ze zásobníku symbol *E* a pokračuje v rozkladu shora dolů. Tyto akce popisuji výrazem *LR*,*popE*.

	b	**e**	**r**	**w**	**;**	**p**	**#**
S	bL,4						
L		e,5	C;L,6	C;L,6			
C			rp,8	wE,7			
E						LR,popE	
b	pop						
e		pop					
r			pop				
w				pop			
;					pop		
p						pop	
#							**acc**

Tab. 1: *Rozkladová tabulka syntaktické analýzy shora dolů*

Obdobně jako v případě spuštění analýzy zdola nahoru jsem si k jejímu ukončení definoval pomocnou množinu - tentokrát *stopLR*. Ta obsahuje symboly, které mohou následovat za výrazem rozkládaným analýzou zdola nahoru. V případě našeho jazyka, je to jen symbol *;*. Tato množina nahrazuje v syntaktickém analyzátoru zdola nahoru znak definující konec vstupního řetězce.

	p	**+**	**-**	**;**	**E**
0	s2				1
1		s3	s4	**acc**	
2		r3	r3	r3	
3	s2				5
4	s2				6
5		r1	r1	r1	
6		r2	r2	r2	

Tab. 2: *Rozkladová tabulka syntaktické analýzy zdola nahoru*

2.3 VSTUPNÍ A VÝSTUPNÍ INFORMACE

Vstupní řetězec je společný pro oba analyzátory. Analyzátor shora dolů zapisuje identifikátory pravidel popisující rozklad vstupu obvyklou metodou na výstup. Analyzátor zdola nahoru zapisuje identifikátory svých pravidel do odděleného řetězce a po ukončení analýzy shora dolů je tento řetězec zapsán v obráceném pořadí na výstup analýzy shora dolů. Jako výsledný rozklad vstupního řetězce je považován výstupní řetězec analýzy shora dolů po jejím ukončení.

2.4 PŘÍKLAD ROZKLADU PROGRAMU

Uvažujme následující kód. Popíši jeho rozklad pomocí výše popsaných pravidel a rozkladových tabulek.

Program: *begin write x+x; end* Přepis: *bwp+p;e#*

Zásobník	**Vstup**	**Operace**	**Výstup**
#S	bwp+p;e#	4:S → bL	
#Lb	bwp+p;e#	Pop	4
#L	wp+p;e#	6:L → C;L	4
#L;C	wp+p;e#	7:C → wE	4,6
#L;Ew	wp+p;e#	Pop	4,6,7
#L;E	p+p;e#	LR,popE	4,6,7
#L;	;e#	Pop	4,6,7,**1,3,3**
#L	e#	5:L → e	4,6,7,**1,3,3**
#e	e#	pop	4,6,7,**1,3,3**,5
#	#	**acc**	4,6,7,**1,3,3**,5

Tab. 3: *Rozklad shora dolů*

Zásobník	**Vstup**	**Operace**	**Výstup**
0	p+p;e#	s2	
0p2	+p;e#	r3:E → id	
0E1	+p;e#	s3	3
0E1+3	p;e#	s2	3
0E1+3p2	;e#	r3:E → id	3
0E1+3E5	;e#	r1:E → E + E	3,3
0E1	;e#	**acc**	3,3,1

Tab. 4: *Rozklad zdola nahoru*

3 SHRNUTÍ

Navrhovaná metoda rozkladu vstupního řetězce je výhodnější než samostatné použití metody shora dolů, nebo zdola nahoru. Při samostatném použití metody shora dolů by bylo obtížnější vytvořit pravidla pro rozklad výrazů, zatímco při samostatném použití metody zdola nahoru by bylo daleko komplikovanější vytvořit rozkladovou tabulku. V rámci projektu bude navrhnut programovací jazyk a sestrojen jeho syntaktický analyzátor, založený na výše popsané metodě.

LITERATURA

[1] Meduna, A.: Formal Languages and Compilers

[2] Češka, M., Rábová, Z.: Gramatiky a jazyky, VUT Brno 1992