

Adeeb Sheikh

CSC 173

Write-Up

Grammar:

The following grammar was originally based on that by Aho and Ullman in their *Foundations of Computer Science*. The grammar was modified to eliminate left recursion, add left factoring, and then add extra operators.

1. $\langle \text{Expression} \rangle \rightarrow \langle \text{Term} \rangle \langle \text{ExpTail} \rangle$
2. $\langle \text{ExpTail} \rangle \rightarrow + \langle \text{Term} \rangle \langle \text{ExpTail} \rangle$
3. $\langle \text{ExpTail} \rangle \rightarrow - \langle \text{Term} \rangle \langle \text{ExpTail} \rangle$
4. $\langle \text{ExpTail} \rangle \rightarrow \langle \text{PostIncrement} \rangle \langle \text{ExpTail} \rangle$
5. $\langle \text{ExpTail} \rangle \rightarrow \epsilon$
6. $\langle \text{PostIncrement} \rangle \rightarrow \langle \text{Increment} \rangle \langle \text{IncrementTail} \rangle$
7. $\langle \text{Term} \rangle \rightarrow \langle \text{Factor} \rangle \langle \text{TermTail} \rangle$
8. $\langle \text{TermTail} \rangle \rightarrow * \langle \text{Factor} \rangle \langle \text{TermTail} \rangle$
9. $\langle \text{TermTail} \rangle \rightarrow / \langle \text{Factor} \rangle \langle \text{TermTail} \rangle$
10. $\langle \text{TermTail} \rangle \rightarrow \% \langle \text{Factor} \rangle \langle \text{TermTail} \rangle$
11. $\langle \text{TermTail} \rangle \rightarrow \epsilon$
12. $\langle \text{Factor} \rangle \rightarrow \langle \text{FactorHead} \rangle \langle \text{FactorTail} \rangle$
13. $\langle \text{FactorHead} \rangle \rightarrow \langle \text{Increment} \rangle \langle \text{Sign} \rangle \langle \text{FactorHead} \rangle$
14. $\langle \text{FactorHead} \rangle \rightarrow \langle \text{Sign} \rangle \langle \text{Increment} \rangle \langle \text{FactorHead} \rangle$
15. $\langle \text{FactorHead} \rangle \rightarrow (\langle \text{Expression} \rangle)$
16. $\langle \text{FactorHead} \rangle \rightarrow \langle \text{Number} \rangle$
17. $\langle \text{Sign} \rangle \rightarrow +$
18. $\langle \text{Sign} \rangle \rightarrow (+)$
19. $\langle \text{Sign} \rangle \rightarrow -$
20. $\langle \text{Sign} \rangle \rightarrow (-)$
21. $\langle \text{Sign} \rangle \rightarrow \epsilon$
22. $\langle \text{Increment} \rangle \rightarrow (++)$
23. $\langle \text{Increment} \rangle \rightarrow (--)$
24. $\langle \text{Increment} \rangle \rightarrow \epsilon$
25. $\langle \text{FactorTail} \rangle \rightarrow !$
26. $\langle \text{FactorTail} \rangle \rightarrow ^ \langle \text{Factor} \rangle$
27. $\langle \text{FactorTail} \rangle \rightarrow \epsilon$
28. $\langle \text{IncrementTail} \rangle \rightarrow \langle \text{TermTail} \rangle$
29. $\langle \text{IncrementTail} \rangle \rightarrow \langle \text{FactorTail} \rangle$
30. $\langle \text{IncrementTail} \rangle \rightarrow \epsilon$

Parse Table:

	+	-	(+)	(-)	(++)	(--)	*	%	/	()	!	^	<i>N</i>	;
E	1	1	1	1	1	1				1				1	
ET	2	3			4	4					5				5
PI					6	6									
T	7	7	7	7	7	7				7				7	
TT	11	11			11	11	8	9	10		11				11
F	12	12	12	12	12	12				12				12	
FH	14	14	14	14	13	13				15				16	
S	17	19	18	20	21	21				21				21	
I	24	24	24	24	22	23				24	24			24	24
FT	27	27	27	27	27	27	27	27	27		27	25	26		27
IT	30	30					28	28	28		30	29	29		30

(N is any number.)

Deterministic Finite Automata:

This program uses several deterministic finite automata, one for each type of token.
(**BOLD** is accepting state.)

T_EOF

STATE	END	OTHER TERMINALS
start	done	reject
done		
reject	reject	reject

T_SPACE

STATE	SPACE	OTHER TERMINALS
start	got_space	reject
got_space	got_space	done
done		
reject	reject	reject

T_NL_SPACE

STATE	SPACE	EOLN	OTHER TERMINALS
start	got_space	got_nl_space	reject
got_space	got_space	got_nl_space	reject
got_nl_space	got_nl_space	got_nl_space	done
done			
reject	reject	reject	reject

T_UNARY

STATE	(+	-)	OTHER TERMINALS

start	got_lparen	reject	reject	reject	reject
got_lparen	reject	got_incr	got_decr	reject	reject
got_incr	reject	reject	reject	done	reject
got_decr	reject	reject	reject	done	reject
done					
reject	reject	reject	reject	reject	reject

T_INCREMENT

T_OPERATOR

T_LITERAL

T_LPAREN

T_RPAREN

T_SEMIC

T_DOT