Introduction to PL/0 Language

xStone

July 8, 2012

1 Lexical Rule

Keyword

begin	call	const	do
end	if	procedure	read
then	var	while	write
odd			

Operator

+	−	*	/
<	<=	>	>=
=	#	:=	

Delimiter

;	,	(	)
.			

Identifier

'ident' $\longrightarrow$ [a-zA-Z]$^+$

Number

'number' $\longrightarrow$ [0-9]$^+$

2 Grammar Rule

1. program $\longrightarrow$ block '.'
2. block $\longrightarrow$ [const_declaraction] [var_declaraction] { procedure_declaraction } statement
3. const_declaraction $\longrightarrow$ 'const' ident '=' number { ',' ident '=' number } ';'
4. var_declaraction $\longrightarrow$ 'var' ident { ',' ident } ';'

5. procedure_declaraction $\longrightarrow$ 'procedure' ident ';' block ';'
6. statement $\longrightarrow$ assign_statement | call_statement | begin_end_statement | if_then_statement | while_do_statement | read_statement | write_statement | ϵ
7. assign_statement $\longrightarrow$ ident ':=' expression
8. call_statement $\longrightarrow$ 'call' ident
9. begin_end_statement $\longrightarrow$ 'begin' statement { ';' statement } 'end'
10. if_then_statement $\longrightarrow$ 'if' condition 'then' statement
11. while_do_statement $\longrightarrow$ 'while' condition 'do' statement
12. read_statement $\longrightarrow$ 'read' '(' ident { ',' ident } ')'
13. write_statement $\longrightarrow$ 'write' '(' expression { ',' expression } ')'
14. condition $\longrightarrow$ 'odd' expression | expression ('=' | '#' | '<' | '<=' | '>' | '>=') expression
15. expression $\longrightarrow$ ['+' | '−'] term { ('+' | '−') term }
16. term $\longrightarrow$ factor { ('$*$' | '/') factor }
17. factor $\longrightarrow$ ident | number | '(' expression ')'