

AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

I-CREDIT HOURS ENGINEERING PROGRAMS



CSE 247 – DESIGN OF COMPILERS

MILESTONE 2

SUBMITTED BY:

Adham Walid	23P0024
Carol Kamal	23P0328
Eslam Mohamed	23P0052
Nouran Mohamed	23P0006
Reetaj Ahmed	23P0114
Hassan Ismail	23P0152

1.Terminals:

Token	Description
int	Keyword for integer type
real	Keyword for real (float) type
string	Keyword for string type
main	Entry point of the program
write	Output function
read	Input function
return	Return statement
if	If conditional
then	Then part of if statement
elseif	Else-if condition
else	Else condition
end	End of block
repeat	Start of repeat loop
until	End condition for repeat loop
endl	Line break or end of line
(Left parenthesis
)	Right parenthesis
{	Left curly brace
}	Right curly brace
,	Comma

Token	Description
;	Semicolon
:=	Assignment operator
<	Less than
>	Greater than
=	Equal to
<>	Not equal to
&&	Logical AND
	Logical OR
+	Addition
-	Subtraction
*	Multiplication
/	Division
IDENTIFIER	Variable or function name
NUMBER	Numeric literal
STRING	String literal

2.Production Rules:

Non-Terminal	Production Rule
<program>	<function_list> <main_function>
<function_list>	<function_statement> <function_list> ϵ
<function_statement>	<function_declaration> <function_body>
<function_declaration>	<datatype> IDENTIFIER "(" <parameter_list> ")"
<parameter_list>	<parameter> <parameter_list_tail> ϵ
<parameter_list_tail>	"," <parameter> <parameter_list_tail> ϵ
<parameter>	<datatype> IDENTIFIER
<main_function>	<datatype> "main" "(" ")" <function_body>
<function_body>	"{" <statement_list> <return_statement> "}"
<statement_list>	<statement> <statement_list> ϵ
<statement>	<declaration_statement> <assignment_statement> <write_statement> <read_statement> <return_statement> <if_statement> <repeat_statement> <function_call>
<declaration_statement>	<datatype> <init_decl> <init_decl_tail> ";"
<init_decl>	IDENTIFIER <optional_assign>
<init_decl_tail>	"," <init_decl> <init_decl_tail> ϵ
<optional_assign>	":=" <expression> ϵ
<assignment_statement>	IDENTIFIER ":=" <expression>

Non-Terminal	Production Rule
<write_statement>	"write" <write_target> ";;"
<write_target>	<expression> "endl"
<read_statement>	"read" IDENTIFIER ";;"
<return_statement>	"return" <expression> ";;"
<if_statement>	"if" <condition_statement> "then" <statement_list> <elseif_block> <else_block> "end"
<elseif_block>	"elseif" <condition_statement> "then" <statement_list> <elseif_block> ϵ
<else_block>	"else" <statement_list> ϵ
<repeat_statement>	"repeat" <statement_list> "until" <condition_statement>
<condition_statement>	<condition> <condition_statement_tail>
<condition_statement_tail>	<boolean_operator> <condition> <condition_statement_tail> ϵ
<condition>	IDENTIFIER <condition_operator> <term>
<expression>	STRING <term> <equation>
<equation>	<primary> <arithmetic_operator> <primary> <equation_tail>
<primary>	<term> "(" <equation> ")"
<equation_tail>	<arithmetic_operator> <primary> <equation_tail> ϵ
<term>	NUMBER IDENTIFIER <function_call>
<function_call>	IDENTIFIER "(" <arguments> ")"

Non-Terminal	Production Rule
<arguments>	<expression> <arguments_tail> ϵ
<arguments_tail>	"," <expression> <arguments_tail> ϵ
<datatype>	"int" "Float" "string"
<boolean_operator>	"&&" " "
<condition_operator>	"=" "<" ">" "<" ">"
<arithmetic_operator>	"+" "-" "*" "/"