LAB – 7(Project)

Group Details:

IT004 Parth Bandel

IT005 Dev Bhagat

IT006 Kaushal Bhalaiya

Grammar Name: Simple and Float number operations with binary and unary assignment operators.

Grammar Rules:

1. X++: used to indicate two increments to “X”
2. X--: used to indicate two decrements to “X” integer variable.
3. X**: used to indicate X variable should be doubled.
4. X=+Y12: Does the addition of X and Y12 and stores value back in “X”.
5. X=-Y: Does the addition of X and Y and stores value back in “X”.
6. X=<integer or float value>: To assign some value to the variable
7. <datatype> <variable name list separated by semicolon>: To define the variables with a particular datatype.

- Variables can be of type: “simple” or “float”
- Variable names should start with capital alphabet and can be followed by maximum 2 digits.
- Sentence end is indicated by “:”

Regular Expression:

letter → [A-Za-z]
cletter → [A-Z]
digit → [0-9]
op → ++ | -- | **
assignop → =+ | =- | =
simple → simple
float → float
id → cletter(digit?)(digit?)(letter)*
num → digit + (.digit +)?(E(+|-)?digit +)?
ws → (blank | tab | newline) +

Token Table:

LEXEMES	TOKEN NAME	ATTRIBUTE VALUE
Any *ws*	—	—
simple	**simp**	—
float	**flt**	—
Any id	**id**	Pointer to table entry
Any number	**num**	Pointer to table entry
++	**op**	INC
--	**op**	DEC
**	**op**	SQR
=+	**assignop**	ASADD
=-	**assignop**	ASSUB
=	**assignop**	AS

Grammer:

S -> TYPE XID EOL | ASSSTMT EOL | EXPR EOL

EXPR -> ID UOP | ID BOP ID

XID -> YID | YID SEMI XID

YID -> ID | ASSSTMT

ASSSTMT -> ID AOP VALUE

AOP -> =

UOP -> ++| --| **

BOP -> =+| =-

ID -> id

EOL -> :

SEMI -> ;

TYPE -> simple | float

Tokens that I've used:

LEXEMES	**TOKENS**	**DISCRIPTION**
simple	SIMP	Simple Datatype
float	FLT	Float Datatype
++	UOP	Unary increment
--	UOP	Unary decrement
**	UOP	Unary double
=	AOP	Assignment operator
=+	BOP	Binary addition operator
=-	BOP	Binary subtraction operator
;	SEMI	To Separate variable list
:	EOL	Sentence end
[A-Z][[0-9]?[0-9]?]?[a-zA-Z]*	ID	Variable name (identifier)
[+-]?[0-9]+	NUM_SIMP	Simple numbers

[+-]?([0-9]+([.][0-9]*)?|[.][0-9]+)	NUM_FLT	Float numbers
Whitespace (‘\0’)	WS	space

First:

First Set	
Non-Terminal Symbol	**First Set**
AOP	=
UOP	++, --, **
BOP	=+, =-
ID	id
EOL	:
SEMI	;
TYPE	simple, float
S	simple, float, id
EXPR	id
YID	id
ASSSTMT	id
XID	id

FOLLOW:

Follow Set

Non-Terminal Symbol	Follow Set
S	$
EXPR	:
XID	:
YID	;, :
ASSSTMT	:, ;
AOP	VALUE
UOP	:
BOP	id
ID	=, ++, --, **, =+, =-, ;, :
EOL	$
SEMI	id
TYPE	id