Compiler Construction (CS 402)

Spring 2020 – Assignment 1

March 31, 2020

Given a simple expression grammar, the attributes of each symbol and the translation scheme for each production rule, develop a syntax-directed translation to convert an expression into Assembly Code.

Production Rules	Translation Schemes (|| is concatenation)
Expr → Term Expr'	Expr.val = Term.val || Expr'.val
Expr' → + Term Expr'	Expr'.val = Term.val || Expr'.val || ASM Code for +
Expr' → – Term Expr'	Expr'.val = Term.val || Expr'.val || ASM Code for –
Expr' → ϵ	
Term → Factor Term'	Term.val = Factor.val || Term'.val
Term' → * Factor Term'	Term'.val = Factor.val || Term'.val || ASM Code for *
Term' → / Factor Term'	Term'.val = Factor.val || Term'.val || ASM Code for /
Term' → % Factor Term'	Term'.val = Factor.val || Term'.val || ASM Code for %
Term' → ϵ	
Factor → (Expr)	Factor.val = Expr.val
Factor → Number	Factor.val = ASM Code for Number

Assembly Code:

+	`POP AX`
	`POP BX`
	`ADD AX,BX`
	`PUSH AX`

–	`POP AX`
	`POP BX`
	`SUB AX,BX`
	`PUSH AX`

*	`POP AX`
	`POP BX`
	`MUL BX`
	`PUSH AX`

/	`POP AX`
	`POP BX`
	`MOV DX,0`
	`DIV BX`
	`PUSH AX`

%	`POP AX`
	`POP BX`
	`MOV DX,0`
	`DIV BX`
	`PUSH DX`

Number	`MOV AX,Number` `PUSH AX`

There are only 2 types of tokens:

1. Operator: + – * / % ( )
2. Number: 0 – 9999
3. Any number of spaces which should be ignored

Output:

On console, the program requires to input an expression and shows token list and an assembly code. A separate ASM file is also needed to be generated, which should be provided to accompanied assembler to display the result of the expression.

Example:

```
C:\> Eval /?
Eval <<expression>>

C:\> Eval 22     * (50 + (36 / 12)     - 16)
Token[0]  = 22 (Number)
Token[1]  = * (Operator)
Token[2]  = ( (Operator)
Token[3]  = 50 (Number)
Token[4]  = + (Operator)
Token[5]  = ( (Operator)
Token[6]  = 36 (Number)
Token[7]  = / (Operator)
Token[8]  = 12 (Number)
Token[9]  = ) (Operator)
Token[10] = - (Operator)
Token[11] = 16 (Operator)
Token[12] = ) (Operator)

<< Assembly Code for the above Expression>>

C:\> masm output.asm

C:\> output

814
```