**CS 323 Assignment 3 Documentation**

by James Vu

1. **Problem Statement**

Implement a Bottom-Up approach for the Syntax Analyzer.

1. **How to use program**

1. Open up command prompt

2. Navigate to folder containing source code

3. Compile using “g++ -o bottomup bottomup.cpp”

4. Run the program using “bottomup.exe”

5. Input file name and press enter

1. **Design of program**

This program first uses a lexer to break down the code into tokens and lexemes. The tokens and lexemes are stored in a queue of pairs of string and string. It prints out the list of token and lexemes in a separate file from the syntax. The syntax analyzer uses a stack to parse the input and the queue to get the input. In the syntax function, it pushes 0 onto the stack to initialize it for the starting state and pushes $ onto the input queue to mark the end. The syntax function uses a while loop which keeps on looping until it reaches the accepted state or an error. Inside the while loop, there is a switch statement which reference which state the stack is at. Each case is handled according to the input. Each case prints out the production or action performed. Shown below are the rules, closures, and transition table used.

Rules:

R0: S' -> S

R1: S -> A

R2: A -> id = E;

R3: E -> E + T

R4: E -> E - T

R5: E -> T

R6: T -> T \* F

R7: T -> T / F

R8: T -> F

R9: F -> ( E )

R10: F -> id

R11: F -> num

First and Follow Sets

First (S) = { id }

First (A) = { id }

First (E) = { (, id, num }

First (T) = { (, id, num }

First (F) = { (, id, num }

Follow (S) = { $ }

Follow (A) = { $ }

Follow (E) = { +, -, ;, ) }

Follow (T) = { \*, /, +, -, ;, ) }

Follow (F) = { \*, /, +, -, ;, ) }

State 0

i0 = ( [S' -> .S] ) = { [S' -> .S], [S -> .A], [A -> .id = E ; ] }

Transition from State 0

i1 = N(i0, S) = { [S' -> S.] }

i2 = N(i0, A) = { [S -> A.] }

i3 = N(i0, id) = { [A -> id .= E ; ] }

Transition from State 1

{ [S' -> S.] } = { }

Transition from State 2

{ [S -> A.] } = { }

Transition from State 3

i4 = N(i3, =) = { [A -> id = .E ; ], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F],

[F -> .( E )], [F -> .id], [F -> .num] }

Transition from State 4

i5 = N(i4, E) = { [A -> id = E .; ], [E -> E .+ T], [E -> E .- T] }

i6 = N(i4, T) = { [E -> T.], [T -> T .\* F], [T -> T ./ F] }

i7 = N(i4, F) = { [T -> F.] }

i8 = N(i4, () = { [F -> ( .E )], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )],

[F -> .id], [F -> .num] }

i9 = N(i4, id) = { [F -> id.] }

i10 = N(i4, num) = { [F -> num.] }

Transition from State 5

i11 = N(i5, ;) = { [A -> id = E ;.] }

i12 = N(i5, +) = { [E -> E + .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )], [F -> .id], [F -> .num] }

i13 = N(i5, -) = { [E -> E - .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )], [F -> .id], [F -> .num] }

Transition from State 6

{ [E -> T.] } = { }

i14 = N(i6, \*) = { [T -> T \* .F], [F -> .( E )], [F -> .id], [F -> .num] }

i15 = N(i6, /) = { [T -> T / .F], [F -> .( E )], [F -> .id], [F -> .num] }

Transition from State 7

{ [T -> F.] } = { }

Transition from State 8

i16 = N(i8, E) = { [F -> ( E .)], [E -> E .+ T], [E -> E .- T] }

i6 = N(i8, T) = { [E -> T.], [T -> T .\* F], [T -> T ./ F] }

i7 = N(i8, F) = { [T -> F.] }

i8 = N(i8, () = { [F -> ( .E )], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )],

[F -> .id], [F -> .num] }

i9 = N(i8, id) = { [F -> id.] }

i10 = N(i8, num) = { [F -> num.] }

Transition from State 9

{ [F -> id.] } = { }

Transition from State 10

{ [F -> num.] } = { }

Transition from State 11

{ [A -> id = E ;.] } = { }

Transition from State 12

i17 = N(i12, T) = { [E -> E + T.], [T -> T .\* F], [T -> T ./ F] }

i7 = N(i12, F) = { [T -> F.]}

i8 = N(i12, () = { [F -> ( .E )], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )],

[F -> .id], [F -> .num] }

i9 = N(i12, id) = { [F -> id.] }

i10 = N(i12, num) = { [F -> num.] }

Transition from State 13

i18 = N(i13, T) = { [E -> E - T.], [T -> T .\* F], [T -> T ./ F] }

i7 = N(i13, F) = { [T -> F.]}

i8 = N(i13, () = { [F -> ( .E )], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )],

[F -> .id], [F -> .num] }

i9 = N(i13, id) = { [F -> id.] }

i10 = N(i13, num) = { [F -> num.] }

Transition from State 14

i19 = N(i14, F) = { T -> T \* F. }

i8 = N(i14, () = { [F -> ( .E )], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )],

[F -> .id], [F -> .num] }

i9 = N(i14, id) = { [F -> id.] }

i10 = N(i14, num) = { [F -> num.] }

Transition from State 15

i20 = N(i15, F) = { T -> T / F. }

i8 = N(i15, () = { [F -> ( .E )], [E -> .E + T], [E -> .E - T], [E -> .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )],

[F -> .id], [F -> .num] }

i9 = N(i15, id) = { [F -> id.] }

i10 = N(i15, num) = { [F -> num.] }

Transition from State 16

i21 = N(i16, )) = { [F -> ( E ).] }

i12 = N(i16, +) = { [E -> E + .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )], [F -> .id], [F -> .num] }

i13 = N(i16, -) = { [E -> E - .T], [T -> .T \* F], [T -> .T / F], [T -> .F], [F -> .( E )], [F -> .id], [F -> .num] }

Transition from State 17

{ [E -> E + T.] } = { }

i14 = N(i17, \*) = { [T -> T \* .F], [F -> .( E )], [F -> .id], [F -> .num] }

i15 = N(i17, /) = { [T -> T / .F], [F -> .( E )], [F -> .id], [F -> .num] }

Transition from State 18

{ [E -> E - T.] } = { }

i14 = N(i18, \*) = { [T -> T \* .F], [F -> .( E )], [F -> .id], [F -> .num] }

i15 = N(i18, /) = { [T -> T / .F], [F -> .( E )], [F -> .id], [F -> .num] }

Transition from State 19

{ T -> T \* F. } = { }

Transition from State 20

{ T -> T / F. } = { }

Transition from State 21

{ [F -> ( E ).] } = { }

Transition Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State | id 22 | num 23 | = 24 | + 25 | - 26 | \* 27 | / 28 | ( 29 | ) 30 | ; 31 | $32 | S 33 | A 34 | E 35 | T 36 | F 37 |
| 0 | S3 |  |  |  |  |  |  |  |  |  |  | 1 | 2 |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  | ACCT |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  | R1 |  |  |  |  |  |
| 3 |  |  | S4 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | S9 | S10 |  |  |  |  |  | S8 |  |  |  |  |  | 5 | 6 | 7 |
| 5 |  |  |  | S12 | S13 |  |  |  |  | S11 |  |  |  |  |  |  |
| 6 |  |  |  | R5 | R5 | S14 | S15 |  | R5 | R5 |  |  |  |  |  |  |
| 7 |  |  |  | R8 | R8 | R8 | R8 |  | R8 | R8 |  |  |  |  |  |  |
| 8 | S9 | S10 |  |  |  |  |  | S8 |  |  |  |  |  | 16 | 6 | 7 |
| 9 |  |  |  | R10 | R10 | R10 | R10 |  | R10 | R10 |  |  |  |  |  |  |
| 10 |  |  |  | R11 | R11 | R11 | R11 |  | R11 | R11 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  | R2 |  |  |  |  |  |
| 12 | S9 | S10 |  |  |  |  |  | S8 |  |  |  |  |  |  | 17 | 7 |
| 13 | S9 | S10 |  |  |  |  |  | S8 |  |  |  |  |  |  | 18 | 7 |
| 14 | S9 | S10 |  |  |  |  |  | S8 |  |  |  |  |  |  |  | 19 |
| 15 | S9 | S10 |  |  |  |  |  | S8 |  |  |  |  |  |  |  | 20 |
| 16 |  |  |  | S12 | S13 |  |  |  | S21 |  |  |  |  |  |  |  |
| 17 |  |  |  | R3 | R3 | S14 | S15 |  | R3 | R3 |  |  |  |  |  |  |
| 18 |  |  |  | R4 | R4 | S14 | S15 |  | R4 | R4 |  |  |  |  |  |  |
| 19 |  |  |  | R6 | R6 | R6 | R6 |  | R6 | R6 |  |  |  |  |  |  |
| 20 |  |  |  | R7 | R7 | R7 | R7 |  | R7 | R7 |  |  |  |  |  |  |
| 21 |  |  |  | R9 | R9 | R9 | R9 |  | R9 | R9 |  |  |  |  |  |  |

1. **Limitations**

Source code size is limited to string.maxsize(). Program will not work on empty files.

1. **Shortcomings**

Program does not print out stack or input.