CO308:COMPILER DESIGN
SESSIONAL TEST II
Lab Submission Report
SPRING 2020

-Submitted by :
Deepankar Acharyya
(CSB17017)

The CFG for the Loop-Construct:

The Context-Free Grammar (CFG) for the above loop construct is as follows:

- A → repeatwhile (C) { S }
- C → id R id | id R n | True | False | !C
- R → == | > | <
- S → id := M;S | ϵ
- M → M+M | M*M | M-M | M / M | (M) | id | n

The Example Loop Considered for the demonstration purpose:

```
1. repeatwhile( abc < 100 ) {
2.     abc = abc + bd ;
3.     bd = bd - 1 ;
4. }
5. $
```

- Arithmetic and assignment statements are present in lines 2 and 3.
- $ present in line 5 marks the end of the program. Without it, the parser works, but the parser will continue to wait for the next lexeme from the lexical analyzer.

Execution Steps:

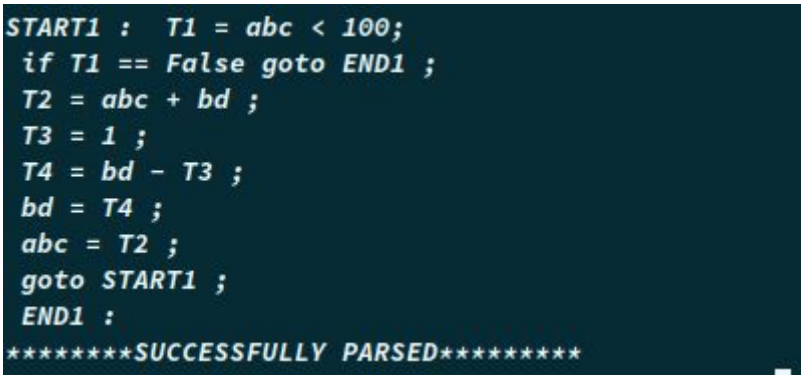
- bison bison.y -d
- flex lex.l
- gcc lex.yy.c bison.tab.c -ll -o output
- ./output

Submitted Files:

- bison.y : the bison file
- lex.l : the lex file
- input.c : text file containing the input sample loop

Expected Output:

- On Valid Input : The 3 AC code will be displayed along with a successfully parsed message



.

- On Invalid Input : An error message stating “Unable to parse” and the error encountered.