**Parser Design Document**

**Abdullah Farooq - 2020022**

**Overview**

The analyzer is tailored to manage a straightforward language of expressions, accommodating fundamental arithmetic functions, variables, and parentheses. It comprises two primary components: Lexical analysis, implemented using Lex code, and Syntax analysis, implemented using Yacc code.

**Lexical Analysis (Lex)**

**2.1 Token Specification**

The Lex code outlines the regulations for tokenizing the source code input, encompassing numbers, variables, and operators.

[0-9]+ { yylval.num = atoi(yytext); return NUMERIC; }

[a-zA-Z]+ { yylval.str = strdup(yytext); return VAR; }

[ \t\n] ; // Disregard spaces

[-+\*()] { return yytext[0]; }

. { printf("Error: Unrecognized character %s\n", yytext); }

**Syntax Analysis (Yacc)**

**3.1 Grammar Definition**

The Yacc code establishes the grammar structures for the expression language, dictating the organization of expressions, terms, and factors.

expr: term expr\_ext

;

expr\_ext: '+' term expr\_ext { printf("+ "); }

| '-' term expr\_ext { printf("- "); }

| /\* empty / { printf("\n"); }

;

term: factor term\_ext

;

term\_ext: '' factor term\_ext { printf("\* "); }

| /\* empty \*/ { }

;

factor: NUMERIC { printf("Num= %d ", $1); }

| VAR { printf("Var = %s ", $1); free($1); }

| '(' expr ')' { }

;

**3.2 Concept of Abstract Syntax Tree (AST)**

While not explicitly articulated in the provided code, the Yacc actions can be expanded to construct an Abstract Syntax Tree (AST). This AST may find utility in subsequent processes or code generation.

Integration

The primary function amalgamates the Lex and Yacc components, instigating the parsing procedure.

int main() {

yyparse(); // Commence parsing

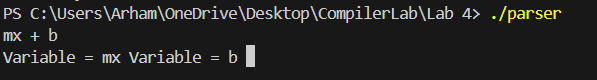
return 0;

}

**Summary**

The analyzer adeptly breaks down and interprets expressions, producing output that mirrors the input's structure. Future improvements could involve extending the grammar for additional functionalities, refining error handling, or integrating the generated AST for advanced applications. Keep this document up-to-date as the analyzer progresses and additional features are introduced.

**Test Scenarios:**

****

**A black screen with white text

Description automatically generated**