Project Report

on

PhyGPT

Developed by

IT132 – JAY RATIYA – 20ITUBS157

IT133 – JITTAM SAKHIA – 20ITUON152

IT141 – DEVANSHU SHAH – 20ITUOS099

IT143 – HETAV SHAH – 20ITUOS078

IT144 – KATHAN SHAH – 20ITUOS093

Guided By:

Prof. Nilamba Vala

Dept. of Information Technology

Department of Information Technology

Faculty of Technology

Dharmsinh Desai University

College Road, Nadiad – 387001

2022-2023

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**PhyGPT"** is a bonafide report of the work carried out by:

1) IT132 – JAY RATIYA – 20ITUBS157
2) IT133 – JITTAM SAKHIA – 20ITUON152
3) IT141 – DEVANSHU SHAH – 20ITUOS099
4) IT143 – HETAV SHAH – 20ITUOS078
5) IT144 – KATHAN SHAH – 20ITUOS093

of Department of Information Technology, semester VI, under the guidance and supervision of **Prof. Nilamba Vala** for the award of the degree of Bachelor of Technology at Dharmsinh Desai University, Nadiad (Gujarat). They were involved in Project in subject of "**Language Translator**" during academic year 2022-2023.

Prof. Nilamba Vala
(Lab In-charge)
Department of Information Technology, Faculty of Technology, Dharmsinh Desai University, Nadiad

Date:

Prof. (Dr.) V K Dabhi,
Head, Department of Information Technology, Faculty of Technology, Dharmsinh Desai University, Nadiad

Date:

Table of Contents

1. INTRODUCTION

1.1 PROJECT DETAILS

Language Name: PhyGPT

Language description:

Write an appropriate language for evaluation of problems related to Kinematic Equations which can do string operations to calculate Final Velocity or Distance based upon factors like Time, Acceleration, Initial Velocity.

Examples of valid questions in this language are:

- A CAR IS MOVING WITH INITIAL SPEED OF 5 MS IT HAS ACCELARATION OF 12 MS2 CALCULATE FINAL VELOCITY AFTER TIME OF 4 S
- A CAR IS STARTING FROM REST IT HAS ACCELARATION OF 12 MS2 CALCULATE FINAL VELOCITY AFTER TIME OF 4 S
- A CAR IS MOVING WITH SPEED OF 5 MS IT HAS ACCELARATION OF 10 MS2 CALCULATE DISTANCE TRAVELLED BY CAR AFTER 3 S USE CONSTANT
- A CAR IS STARTING FROM REST IT HAS ACCELARATION OF 10 MS2 CALCULATE DISTANCE TRAVELLED BY CAR AFTER 3 S USE CONSTANT

1.2 PROJECT PLANNING

List of Students with their Responsibilities:

1) IT132 JAY RATIYA: Regular Expression, DFA Design, Final Report
2) IT133 JITTAM SAKHIA: YACC & Scanner Phase Implementation, Final Report
3) IT141 DEVANSHU SHAH: YACC & Scanner Phase Implementation, Final Report
4) IT143 HETAV SHAH: YACC & Scanner Phase Implementation, Final Report
5) IT144 KATHAN SHAH: Grammar Rules, DFA Design, Final Report

2. Lexical Phase Design

2.1 Regular Expression:

Keywords:

ACCELARATION
SPEED
VELOCITY
TIME
DISTANCE

Identifiers:

CONSTANT
REST

Operations:

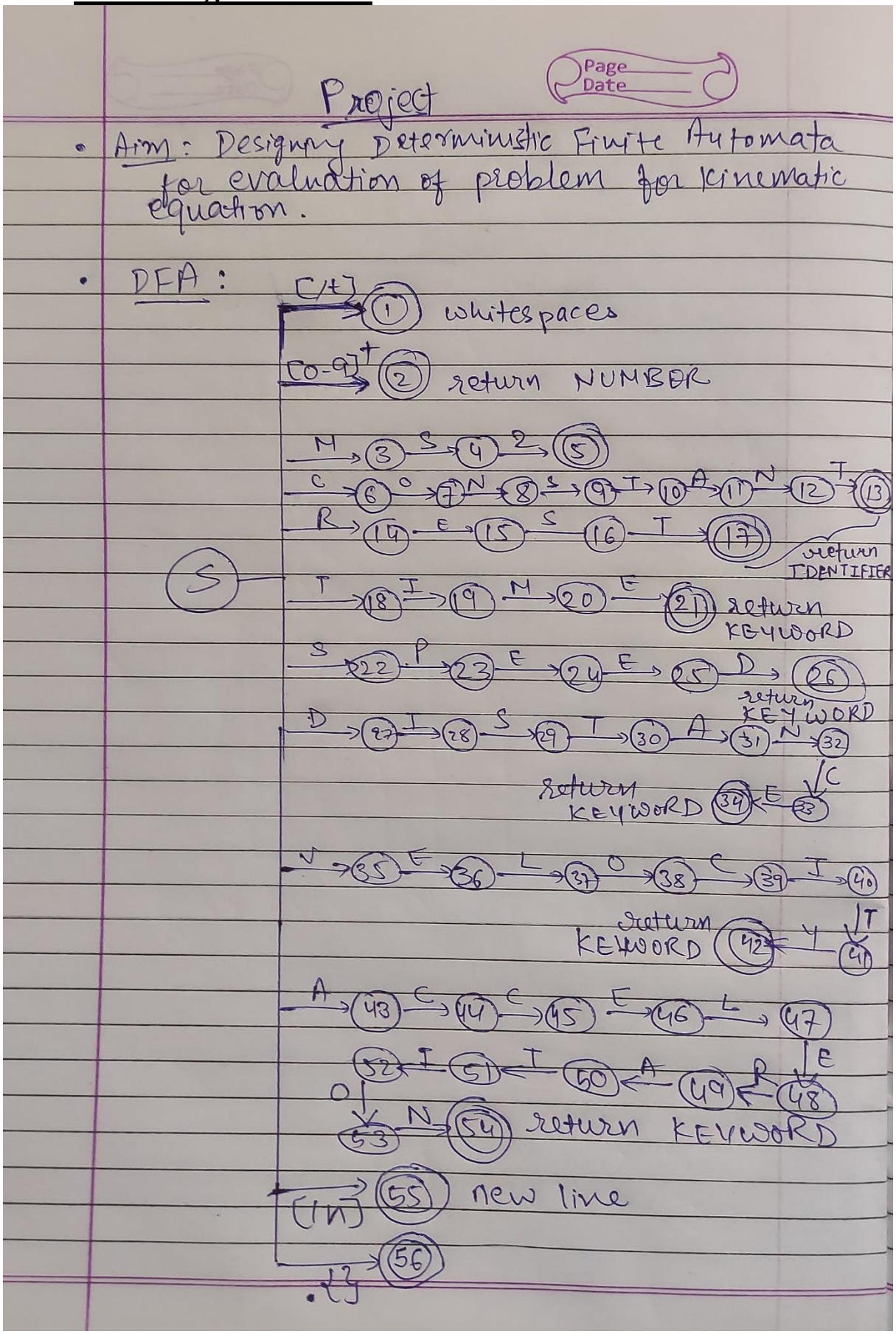
Values type: int and float RE

Token
[0-9]+ int

Delimiters: {\n \t} RE

Token
[\n] new line
[\t] whitespace

2.2 DFA Design for Lexer:



2.3 Implementation of Lexer:

Flex Program: (lex.l)

```
%
{
/* Definition section */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "y.tab.h"
    extern int yylval;
    char *token;
%
}

/* Rule Section */
%%
        [0 - 9] +
{
    yylval = atoi(yytext);
    return NUMBER;
}
MS2
{
}
CONSTANT
{
    return IDENTIFIER;
}
REST
{
    return IDENTIFIER;
}
ACCELARATION | SPEED | VELOCITY | TIME | DISTANCE { printf("%s is
KEYWORD\n", yytext); }

[\t];
" ";
[\n] return 0;

.{}
%%
    int yywrap()
{
    return 1;
}
```

2.4 Output Screenshots of Lexer:

1) If Initial Speed, Acceleration and Time is given

```
 PhyGPT
A CAR IS MOVING WITH INITIAL SPEED OF 5 MS IT HAS ACCELARATION OF 12 MS2 CALCULATE FINAL VELOCITY AFTER TIME OF 4 S
SPEED is KEYWORD
ACCELARATION is KEYWORD
VELOCITY is KEYWORD
TIME is KEYWORD

 ANS IS = 53

Yay Your Question is Correct.
```

2) If Acceleration and Time is given

```
 PhyGPT
A CAR IS STARTING FROM REST IT HAS ACCELARATION OF 12 MS2 CALCULATE FINAL VELOCITY AFTER TIME OF 4 S
ACCELARATION is KEYWORD
VELOCITY is KEYWORD
TIME is KEYWORD

 ANS IS = 48

Yay Your Question is Correct.
```

3) If Initial Speed, Acceleration, Time, and Constant is given

```
 PhyGPT
A CAR IS MOVING WITH SPEED OF 5 MS IT HAS ACCELARATION OF 10 MS2 CALCULATE DISTANCE TRAVELLED BY CAR AFTER 3 S USE CONSTANT
SPEED is KEYWORD
ACCELARATION is KEYWORD
DISTANCE is KEYWORD

 ANS IS = 60

Yay Your Question is Correct.
```

4) If Acceleration, Time, and Constant is given

```
 PhyGPT
A CAR IS STARTING FROM REST IT HAS ACCELARATION OF 10 MS2 CALCULATE DISTANCE TRAVELLED BY CAR AFTER 3 S USE CONSTANT
ACCELARATION is KEYWORD
DISTANCE is KEYWORD
10 3
 ANS IS = 45

Yay Your Question is Correct.
```

2.5 Execution Environment Setup:

Step by Step Guide to Install FLEX and Run FLEX Program using Command Prompt(cmd)

Step 1:

/*For downloading CODEBLOCKS */
- Open your Browser and type in "codeblocks"
- Goto to Code Blocks and go to downloads section
- Click on "Download the binary release"
- Download codeblocks-20.03mingw-setup.exe
- Install the software keep clicking on next

/*For downloading FLEX GnuWin32 */
- Open your Browser and type in "download flex gnuwin32"
- Goto to "Download GnuWin from SourceForge.net"
- Downloading will start automatically
- Install the software keep clicking on next

/*SAVE IT INSIDE C FOLDER*/

Step 2:

/*PATH SETUP FOR CODEBLOCKS*/
- After successful installation

Goto program files->CodeBlocks-->MinGW-->Bin
- Copy the address of bin :- it should somewhat look like this

C:\Program Files (x86)\CodeBlocks\MinGW\bin
- Open Control Panel-->Goto System-->Advanced System Settings-->Environment Variables
- Environment Variables--> Click on Path which is inside System variables - Click on edit
- Click on New and paste the copied path to it:- - C:\Program Files (x86)\CodeBlocks\MinGW\bin
- Press Ok!

Step 3:

/*PATH SETUP FOR GnuWin32*/
- After successful installation Goto C folder
- Goto GnuWin32-->Bin
- Copy the address of bin it should somewhat look like this

C:\GnuWin32\bin
- Open Control Panel-->Goto System-->Advanced System Settings-->Environment Variables

Language Translator (IT 608)

12
- Environment Variables--> Click on Path which is inside System

variables - Click on edit
- Click on New and paste the copied path to it:- - C:\GnuWin32\bin - Press Ok!
/*WARNING!!! PLEASE MAKE SURE THAT PATH OF CODEBLOCKS
IS BEFORE GNUWIN32---THE ORDER MATTERS*/

Step 4:
- Create a folder on Desktop flex_programs or whichever name you like - Open notepad type in a flex program - Save it inside the folder like filename.l
-Note :- also include “”” void yywrap(){} “””””””” in the .l file
/*Make sure while saving save it as all files rather than as a text document*/

Step 5:
/*To RUN FLEX PROGRAM*/
- Goto to Command Prompt(cmd)
- Goto the directory where you have saved the program - Type in command :- flex filename.l - Type in command :- gcc lex.yy.c
- Execute/Run for windows command prompt :- a.exe

Step 6:
- Finished

3. Syntax Analyzer Design

3.1 Grammar Rules:

E: IDENTIFIER NUMBER NUMBER IDENTIFIER {
$$=half*$2*$3*$3;} |

IDENTIFIER NUMBER NUMBER {
$$= $2*$3*1.0;} |

NUMBER NUMBER NUMBER IDENTIFIER {
$$=$1*$3*1.0 + half*$2*$3*$3*1.0;} |

NUMBER NUMBER NUMBER {
$$=$2*$3+$1;}

3.2 YACC based implementation of Syntax Analyzer: FLEX Program: (lex.l)

```
%
{
/* Definition section */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "y.tab.h"
    extern int yylval;
    char *token;
%
}

/* Rule Section */
%%
        [0 - 9] +
{
    yylval = atoi(yytext);
    return NUMBER;
}
MS2
{
}
CONSTANT
{
    return IDENTIFIER;
}
REST
{
    return IDENTIFIER;
}
ACCELARATION | SPEED | VELOCITY | TIME | DISTANCE { printf("%s is
KEYWORD\n", yytext); }

[\t];
" ";
[\n] return 0;

.{}
%%
    int yywrap()
{
    return 1;
}
```

YACC Program: (yacc.y)

```
%{
/* Definition section */
#include<stdio.h>
int flag=0;
float half=0.5f;
%}

%token NUMBER
%token IDENTIFIER

/* Rule Section */
%%

ArithmeticExpression: E {
        printf("\n ANS IS = %d \n",$$ );
        return 0;
};

E:  IDENTIFIER NUMBER NUMBER IDENTIFIER {
    $$=half*$2*$3*$3;} |
    IDENTIFIER NUMBER NUMBER {
    $$= $2*$3*1.0;} |
    NUMBER NUMBER NUMBER IDENTIFIER {
    $$=$1*$3*1.0 + half*$2*$3*$3*1.0;} |
    NUMBER NUMBER NUMBER {$$=$2*$3+$1;};

%%

void main()
{
   printf("\n PhyGPT\n");
   yyparse();
   if(flag==0)
   printf("\nYay Your Question is Correct.\n\n");
}

void yyerror()
{
   printf("\n Oops Your Question is wrong\n\n");
   flag=1;
}
```

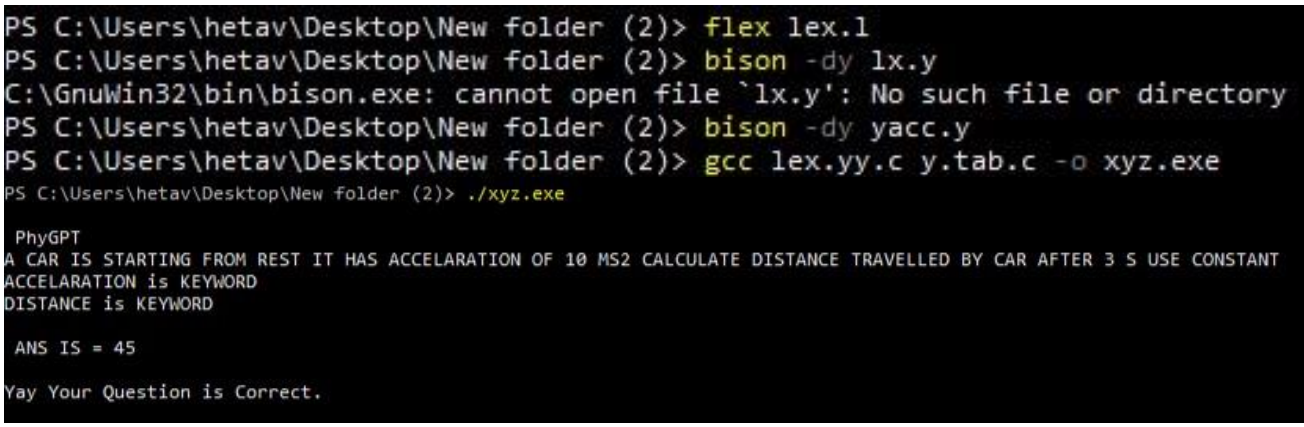
3.3 Execution Environment Setup:

Download flex and bison from the given links.
http://gnuwin32.sourceforge.net/packages/flex.htm
http://gnuwin32.sourceforge.net/packages/bison.htm
when installing on windows you store this in c:/gnuwin32 folder and not in c:/program files(X86)/gnuwin32

Download IDE
https://sourceforge.net/projects/orwelldevcpp/ set environment variable for flex and bison.

To run the program:
Open a prompt, cd to the directory where your ".l" and ".y" are, and compile them with: flex lex.l bison -dy yacc.y gcc lex.yy.c y.tab.c -o xyz.exe

3.4 Output Screenshot of YACC based Implementation:



4. CONCLUSION

This project has been implemented from what we have learned in our college curriculum and many rich resources from the web. After doing this project we conclude that we have soaked up more knowledge about how different compilers work in the practical world and how various types of errors are handled.

Link:
GitHub Repository Link