МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | М.Д.  Поляк |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

«Восходящая трансляция на основе вычисления синтезируемых атрибутов»

ПО ДИСЦИПЛИНЕ «МЕТОДЫ ТРАНСЛЯЦИИ»

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4831 |  |  |  | К.А. Корнющенков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

Цель работы:

Дополните программу, разработанную в лабораторной работе N1, вычислением синтези- руемых атрибутов для программной реализации синтаксически управляемого решения задачи, выбранной по номеру варианта.

Задание:

4 вариант

<S>::=<E>

<E>::=<E>+<T>

<S>::=<T>

<T>::=<T>\*<F>

<T>::=<F>

<F>::=x

<F>::=6

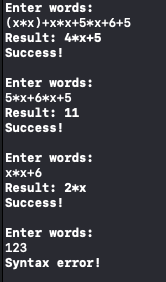
<F>::=5

<F>::=(E)

S – аксиома

Синтаксически управляемое дифференцирование алгебраического выражения

Результаты выполнения работы:



Вывод:

В ходе выполнения лабораторной работы дополнили программу вычислением синтезируемых атрибутов с помощью утилиты Bison.

Приложение А:

flex.txt

%option noyywrap

%{

#include <stdio.h>

#include "lab8.h"

%}

Identifier [\*x65()]

BadSymbol [^\*x65()\n]

%%

{Identifier} { return \*yytext; }

{BadSymbol} { return badsym; }

\n { return ident; }

%%

Bison.txt

%{

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

void yyerror(char const\* msg);

int yylex();

int yyparse();

%}

%union{char\* c;}

%type <c> S E T F

%token ident

%token badsym

%%

word: | word line;

line: ident |

S ident { printf("Success!\n\n"); YYACCEPT; } |

badsym { printf("Syntax error!\n"); YYERROR;} |

error ident { YYABORT;};

S : E { printf("Result: %s", $1); };

E : E'+'T {

if ($$ == "5" || $$ == "6"){

if ($1 == "5" || $1 =="6"){

$$ = "0";

}else{

$$ = $1;

}

}else{

if (strlen($$) == strlen($1){

if (strlen($$)>2){

char\* data = $1;

int a = $$ - '0';

int b = data - '0';

int finish = a\*b;

char buffer[10];

sprintf(buffer, "%d", finish);

$$ = buffer;

strcat(yyval.c,"\*x");

}else{

int check = 0;

if ($$ == "x"){

$$ = "1";

check = 1;

}

if($1 == "x"){

int a = $$ - '0' + 1;

char buffer[10];

sprintf(buffer, "%d", a);

$$ = buffer;

}else{

if (check != 1){

int a = $$ - '0' + $$ - '0';

char buffer[10];

sprintf(buffer, "%d", a);

$$ = buffer;

}

}

}

}else{

char\* data = $$;

$$ = malloc(strlen(data)+strlen("+")+strlen($1));

strcat($$, data);

if (!($1 == "5" || $1 == "6")){

strcat($$,"+");

strcat($$,$1);

}

}

}

};

E : T { strcpy($$, $1); };

T : T'\*'F {

char\* time1 = $$;

$$ = malloc(strlen($$)+strlen("\*") + strlen($1));

strcat($$, time1);

strcat($$, "\*");

strcat($$, $1);

int cntr = 0;

for (int i = 0; i < strlen($$); ++i) {

if ($$[i] == 'x'){

++cntr;

}

}

if (cntr>0){

char\* time = $$;

char count = cntr + '0';

$$ = malloc(strlen($$)+strlen("\*") + strlen(&count));

char buffer[10];

sprintf(buffer, "%d", cntr);

strcat($$, buffer);

strcat($$,"\*");

strcat($$,time);

int pos = -1;

for (int i = 0; i < strlen($$) - 1; ++i) {

if ($$[i] == 'x' && $$[i+1] == '\*') {

pos = i;

}

}

if (pos != -1) {

int i;

for (i = pos; i < strlen($$) - 3; ++i) {

$$[i] = $$[i+2];

}

$$[i+1] = '\0';

$$[i+2] = '\0';

} else {

for (int i = 0; i < strlen($$) - 1; ++i) {

if ($$[i] == '\*' && $$[i+1] == 'x') {

pos = i;

}

}

if (pos != -1) {

int i;

for (i = pos; i < strlen($$) - 3; ++i) {

$$[i] = $$[i+2];

}

$$[i+1] = '\0';

$$[i+2] = '\0';

}

}

if (pos == -1) {

$$ = "";

}

};

T : F { strcpy($$, $1); };

F : 'x' { strcpy($$, "x"); };

F : '5' { strcpy($$, "5"); };

F : '6' { strcpy($$, "6"); };

F : '('E')' { strcat($$, $2); };

%%

void yyerror(char const\* msg) { printf("Syntax error!\n"); }

int main() {

while(1) {

printf("Enter words:\n");

yyparse();

}

return 0;

}