Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №4

по дисциплине «Трансляторы и компиляторы»

## на тему «Разработка синтаксического анализатора методом рекурсивного спуска»

Выполнили студенты группы 22ВВП1

Демин М.С.

Беляев Д. И.

Приняли:

Дубинин В.Н.

Карамышева Н.С.

Пенза 2024

**Название**

Разработка синтаксического анализатора методом рекурсивного спуска

**Цель работы**

Исследование распознавателей КС-языков без возвратов.

**Задание**

Написать программу на языке C для заданной грамматики методом рекурсивного спуска. Варианты заданий.

1. G({a,b,c,d},{A,B,C,S},P,S)

P:

S→ aB | dC | b

A→ aC | bB | dC

B→ aC | dC | a

C→ b | aA | d

**Листинг**

**main.c**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_RULES 100

int Rules[MAX\_RULES];

char\* SymbolS(char\* szInput);

char\* SymbolA(char\* szInput);

char\* SymbolB(char\* szInput);

char\* SymbolC(char\* szInput);

void AddRuleToOutput(int nRule) {

static int i = 0;

if (i == MAX\_RULES) {

printf("Input chain is too long\n");

exit(0);

}

Rules[i++] = nRule;

}

int main() {

char szInput[256];

char\* szOutput;

int i = 0;

memset(Rules, 0, MAX\_RULES \* sizeof(int));

printf("Enter chain> ");

scanf("%s", szInput);

szOutput = SymbolS(szInput);

if (!szOutput) {

printf("Parse error\n");

return 1;

}

if (\*szOutput) {

printf("Unexpected symbol %c\n", \*szOutput);

return 2;

}

printf("Output tree for chain: ");

while (Rules[i]) {

printf("%d ", Rules[i++]);

}

printf("\n");

return 0;

}

char\* SymbolS(char\* szInput) {

// S -> aB | dC | b

switch (\*szInput) {

case '\0':

return szInput;

case 'a':

AddRuleToOutput(1); // S -> aB

return SymbolB(szInput + 1);

case 'd':

AddRuleToOutput(2); // S -> dC

return SymbolC(szInput + 1);

case 'b':

AddRuleToOutput(3); // S -> b

return szInput + 1;

default:

return NULL;

}

}

char\* SymbolA(char\* szInput) {

// A -> aC | bB | dC

switch (\*szInput) {

case '\0':

return szInput;

case 'a':

AddRuleToOutput(4); // A -> aC

return SymbolC(szInput + 1);

case 'b':

AddRuleToOutput(5); // A -> bB

return SymbolB(szInput + 1);

case 'd':

AddRuleToOutput(6); // A -> dC

return SymbolC(szInput + 1);

default:

return NULL;

}

}

char\* SymbolB(char\* szInput) {

// B -> aC | dC | a

switch (\*szInput) {

case '\0':

return szInput;

case 'a':

if (\*(szInput + 1) == 'a' || \*(szInput + 1) == 'b' || \*(szInput + 1) == 'd') {

AddRuleToOutput(7); // B -> aC

return SymbolC(szInput + 1);

}

else {

AddRuleToOutput(9); // B -> a (final)

return szInput + 1;

}

case 'd':

AddRuleToOutput(8); // B -> dC

return SymbolC(szInput + 1);

default:

return NULL;

}

}

char\* SymbolC(char\* szInput) {

// C -> b | aA | d

switch (\*szInput) {

case '\0':

return szInput;

case 'b':

AddRuleToOutput(10); // C -> b

return szInput + 1;

case 'a':

AddRuleToOutput(11); // C -> aA

return SymbolA(szInput + 1);

case 'd':

AddRuleToOutput(12); // C -> d

return szInput + 1;

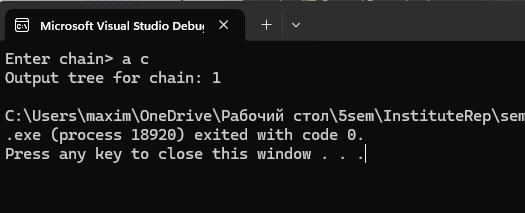
default:

return NULL;

}

}

**Результат работы**



**Вывод**

Исследовали распознаватели КС-языков без возвратов.