Министерство образования Республики Беларусь

УО «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

ОТЧЕТ

по лабораторной работе № 1

РАСПОЗНАВАНИЕ И ПОИСК ЛЕСЕМ

Вариант 4

Выполнил : Марьин А.C.

Студент гр. 051005

Проверил : Болтак С.В.

Минск 2021

**Регулярное выражение**

Константы:

Letter = [A-Z]

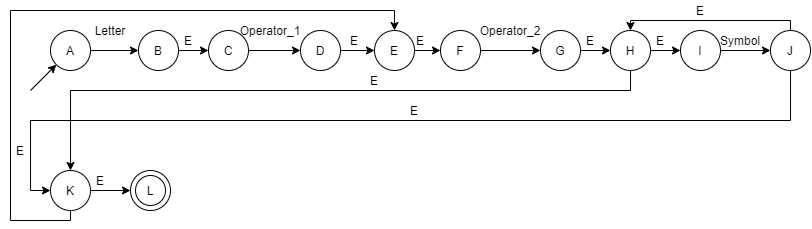
Operator\_1 = ‘:’

Operator\_2 = ‘\’

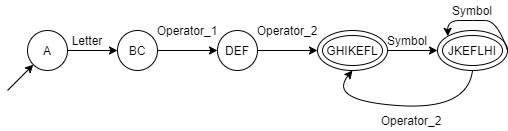
Symbol = ^ ’\*’| ^ ’|’| ^ ‘/’| ^ ‘:’| ^ ‘”’| ^’<’| ^’>’| ^’?’| ^’\’

**ValidString** = Letter + Operator\_1 + (Operator\_2 + Symbol +)\*

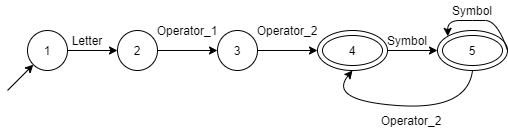
**НКА (для регулярного выражения)**

****

**ДКА (для регулярного выражения)**

****

**ДКА (готовый к программной реализации)**

****

**Таблица переходов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ctUnknown** | **ctLetter** | **ctOperator\_1** | **ctOperator\_2** | **Symbol** | **X** |
| 0 | 0 | 0 | 0 | 0 | Состояние  “Ошибка” |
| 0 | 2 | 0 | 0 | 0 | Состояние  “Ждем латинскую букву” |
| 0 | 0 | 3 | 0 | 0 | Состояние  “Читаем латинскую букву, ждем двоеточие” |
| 0 | 0 | 0 | 4 | 0 | Состояние  “Читаем двоеточие, ждем обратный слеш” |
| 0 | 0 | 0 | 0 | 5 | Состояние  “Читаем обратный слеш, ждем допустимый символ файла” |
| 0 | 0 | 0 | 4 | 5 | Состояние  “Читаем допустимый символ файла, ждем допустимый символ файла или обратный слеш” |

**Код программы, реализующий работы ДКА**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <conio.h>

#define State 5

#define n 100

/\*

char c1[] = "ctUnknown";

char c2[] = "ctLetter";

char c3[] = "ctOperator\_1";

char c4[] = "ctOperator\_2";

char c5[] = "ctSymbol";

\*/

int IsFinalState[State+1] = { 0, 0, 0, 0, 1, 1 };

int Transitions[State+1][State] = {

{0, 0, 0, 0, 0},

{0, 2, 0, 0, 0},

{0, 0, 3, 0, 0},

{0, 0, 0, 4, 0},

{0, 0, 0, 0, 5},

{0, 0, 0, 4, 5}

};

int GetCharType(char c)

{

if (c >= 65 && c <= 90)

{

return 1;

}

else if (c == ':')

{

return 2;

}

else if (c == '\\')

{

return 3;

}

else if (c != '\*' && c != '|' && c != '\\' && c != ':' && c != '"' && c != '<' && c != '>' && c != '?' && c != '/')

{

return 4;

}

else

{

return 0;

}

}

int GetCharType\_1(char c)

{

if (c == ':')

{

return 2;

}

else if (c == '\\')

{

return 3;

}

else if (c != '\*' && c != '|' && c != '\\' && c != ':' && c != '"' && c != '<' && c != '>' && c != '?' && c != '/')

{

return 4;

}

else

{

return 0;

}

}

int GetCharType\_2(char c)

{

if (c == ':')

{

return 2;

}

else if (c == '\\')

{

return 3;

}

else if (c != '\*' && c != '|' && c != '\\' && c != ':' && c != '"' && c != '<' && c != '>' && c != '?' && c != '/' && c != 32)

{

return 4;

}

else

{

return 0;

}

}

int checkString(char s[], int x)

{

int state = 1;

for (int i = 0; i < strlen(s); i++)

{

int k = GetCharType(s[i]);

if (i >= 2)

{

if (x == 1)

{

k = GetCharType\_1(s[i]);

}

if (x == 2)

{

k = GetCharType\_2(s[i]);

}

}

state = Transitions[state][k];

}

return IsFinalState[state];

}

int main()

{

int x = 0;

char str[n];

char words[n][n] = { '\0' };

printf("\n1 - task 1, 2 - task 2, 3 - exit\n");

scanf\_s("%d", &x);

getchar();

while (x != 3)

{

if (x == 1)

{

printf("\nEnter path:\n");

fgets(str, n, stdin); // без вылета gets

int k = checkString(str,1);

if (k == 1)

{

printf("\nCorrect!\n");

}

else

{

printf("\nError!\n");

}

for (int j = 0; j < strlen(str); j++)

{

str[j] = '\0';

}

}

if (x == 2)

{

printf("\nEnter pathes:\n");

fgets(str, n, stdin); // без вылета gets

int i = 0, j = 0;

while (i < (int)(strlen(str) - 1))

{

while (str[i] == ',')

{

i++;

}

int start = i;

while (str[i] != ',' && i < (int)(strlen(str) - 1))

{

i++;

}

int end = i;

int l = 0;

for (int k = start; k < end; k++)

{

words[j][l] = str[k];

l++;

}

j++;

}

int k = 0;

printf("\nYour pathes:\n");

for (int i = 0; i < j; i++)

{

if (words[i] != '\0')

{

k = checkString(words[i],2);

printf("%s - ", words[i]);

if (k == 1)

{

printf("Correct!\n");

}

else

{

printf("Error!\n");

}

}

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < strlen(words[i]); j++)

{

words[i][j] = '\0';

}

}

for (int j = 0; j < strlen(str); j++)

{

str[j] = '\0';

}

}

printf("\n1 - task 1, 2 - task 2, 3 - exit\n");

scanf\_s("%d", &x);

getchar();

}

return 0;

}

**Тестирование и отладка программы**

**Корректные примеры:**

1. C:\Windows\winmine.exe

2. D:\WebServer\home\site.by\www\.htaccess

3. Z:\autoexec.bat

4. N:\testfile.

5. X:\testfile2

**Состояние ошибки:**

1. AQ:\

2. A::\

3. A:\\

4. A :\

5. A:\ abc

**Пример для подстрок**

1. **A:\,B:,C,D:\qaz qaz, D:\qaz**
2. **A:\,B:,C,D:\\*qazqaz, D:\|qaz,E:\q:**
3. **A:\,B:,C,D:\>qazqaz, D:\<qaz,E:\q**
4. **A:\,B:,C,D:\\qazqaz, D:\qaz,E:\q**
5. **A:\,B:\ qaz,C:\qaz**