МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем и цифровых технологий

**ОТЧЕТ**

по лабораторной работе №2

на тему: «Структурное тестирование программного обеспечения»

по дисциплине: «Качество и тестирование программного обеспечения»

Вариант 10

Выполнил: Музалевский Н.С.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 92ПГ

Проверил: Олькина Е.В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

Орел, 2022 г.

Вариант 9. Требования для шкафа-купе:

1. Шкаф имеет три вертикальные секции и три двери

2. Двери шкафа должны крепиться на горизонтальных направляющих

3. Высота шкафа должна составлять 2 метра

4. Двери шкафа должны быть созданы с таким расчетом, чтобы ими ничего нельзя было прищемить

5. Глубина шкафа должна быть выбрана таким образом, чтобы в него помещался велосипед

Комментарии:

1. Необходимо указать три вертикальные секции на протяжение всего шкафа или только в 1 части.

2. Двери шкафа должны крепиться на вертикальных направляющих

3. Необходимо также указать ширину шкафа и длину.

4. К требованию номер 5 надо добавить среднестатистический или габариты самого

**Задание:** Парсер, использующий простые регулярные выражения, вводимые с клавиатуры, содержащие управляющие конструкции . – любой символ, \* - 0 и более символов, + - 1 и более символов (вводится регулярное выражение и строка, результатом является позиция, с которой это выражение встречается в тексте)

**Программа на C#:**

int idxStart = 0;//1

int idxReg = 0;

bool skip = false;

for (int i = 0; i < str.Length; i++)//2

{

if (normalRegExp[idxReg] == '\*')//3

{

skip = true;//4

idxReg++;

if (idxReg == normalRegExp.Length)//5

{

PrintAnswer(regularExp, str, idxStart);//6

return;

}

}

if (normalRegExp[idxReg] == '+')//7

{

ResetIdx(idxReg, i, ref idxStart);//8

idxReg++;//9

skip = true;

}

else if (normalRegExp[idxReg] == '.')//10

{

ResetIdx(idxReg, i, ref idxStart);//11

idxReg++;

}

else

{

if (str[i] == normalRegExp[idxReg])//12

{

ResetIdx(idxReg, i, ref idxStart);//13

idxReg++;//14

skip = false;

}

else if (!skip)//15

{

if (idxReg == normalRegExp.Length - 1)//16

{

PrintAnswer(regularExp, str, idxStart);//6

return;

}

idxStart++;//17

i = idxStart + 1;

idxReg = 0;

}

}

if (idxReg == normalRegExp.Length)//18

{

PrintAnswer(regularExp, str, idxStart);//6

return;

}}

Console.WriteLine("нет вхождений");//19

На основе текста программы метода dfs\_paths сформирован потоковый граф, представленный на рисунке 1.

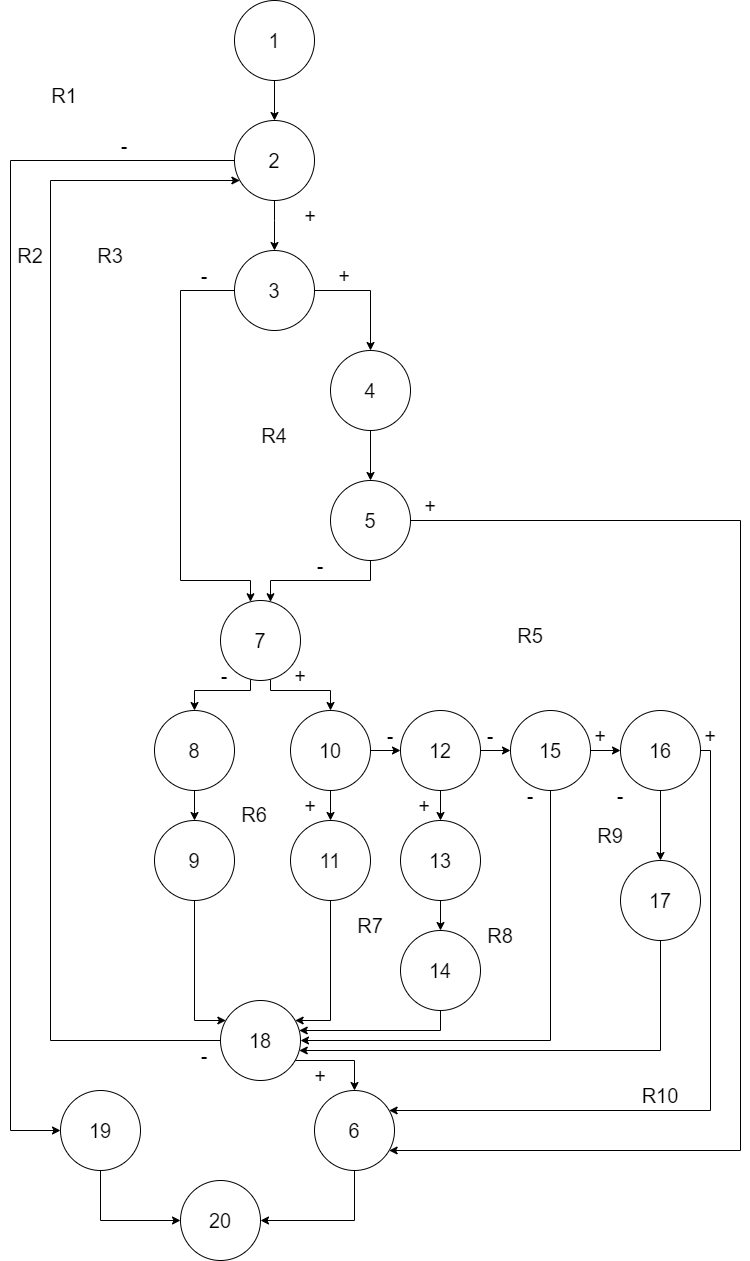


Рисунок 1 – Потоковый граф метода dfs\_paths.

**Цикломатическая сложность**

**1)** 10 регионов

**2)** 28 дуг - 20 узлов + 2 = 10

**3)** Предикатные узлы: 2,3,5,7,10,12,15,16,18

9 предикатных узла + 1 = 10

**Базовое множество**

**Путь 1:** 1-2-19-20

**Путь 2:** 1-2-3-4-5-6-20

**Путь 3:** 1-2-3-7-8-9-18-6-20

**Путь 4:** 1-2-3-7-10-11-18-6-20

**Путь 5:** 1-2-3-7-10-12-13-14-18-6-20

**Путь 6:** 1-2-3-7-10-...-12-15-18-6-20

**Путь 7:** 1-2-3-7-10-12-15-16-6-20

**Путь 8:** 1-2-3-7-10-12-15-16-17-18-19-20

**Путь 9:** 1-2-3-4-5-7-8-9-18-6-20

**Путь 10:** 1-2-3-4-5-7-10-11-18-6-20

**Тестовые варианты**

**1:**Исх.данные:

normalRegExp = “fsf\*+”

str = “”

regularExp = “fsf\*+”

Ожид.рез-т: "Нет вхождений"

**2:**Исх.данные:

normalRegExp = “\*”

str = “fdsfs”

regularExp = “\*”

Ожид.рез-т: Регулярное выражение: \* Строку: fdsfs Индекс: 0

**3:**Исх.данные:

normalRegExp = “+”

str = “fsssssssdf”

regularExp = “+”

Ожид.рез-т: Регулярное выражение: + Строку: fsssssssdf Индекс: 0

**4:** Исх.данные:

normalRegExp = “.”

str = “dsf”

regularExp = “.”

Ожид.рез-т: Регулярное выражение: + Строку: fsssssssdf Индекс: 0

**5:** Исх.данные:

normalRegExp = “@”

str = “@fds”

regularExp = “@”

Ожид.рез-т: Регулярное выражение: @ Строку: @fds Индекс: 0

**6:** Исх.данные:

normalRegExp = “+f”

str = “sdf”

regularExp = “+f”

Ожид.рез-т: Регулярное выражение: +f Строку: sdf Индекс: 0

**7:**Исх.данные:

normalRegExp = “1”

str = “e1”

regularExp = “1”

Ожид.рез-т: Регулярное выражение: 1 Строку: e1 Индекс: 1

**8:** Исх.данные:

normalRegExp = “gf”

str = “r”

regularExp = “gf”

Ожид.рез-т: нет вхождений

**9:** Исх.данные:

normalRegExp = “\*+”

str = “dfdsf”

regularExp = “\*”

Ожид.рез-т: Регулярное выражение: \*+ Строку: dfdsf Индекс: 0

**10:** Исх.данные:

normalRegExp = “\*.”

str = “dd”

regularExp = “\*.”

Ожид.рез-т: Регулярное выражение: \*. Строку: dd Индекс: 0