|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**ЛАБОРАТОРНАЯ РАБОТА № 2**

**ПО КУРСУ:**

***«ТЕОРИЯ ФОРМАЛЬНЫХ ЯЗЫКОВ»***

Студент *Сербин Д.А.*

Преподаватель *Непейвода А.Н.*

*Москва, 2021 г.*

Оглавление

[1. Первая задача 3](#_Toc85412791)

[2. Вторая задача 5](#_Toc85412792)

[3. Третья задача (чётный вариант) 7](#_Toc85412793)

# 1. Первая задача

Задание: написать три эквивалентных (описывающих один и тот же язык) регулярных выражения: академическое; с использованием отрицания; с использованием ленивой итерации Клини. Минимальная длина regex — 10 символов. Сравнить производительность этих regex на 10 тестах длиной от 100 до символов.

Был выбран язык, описывающий двоичные записи чисел, делящихся на 3. Будем считать, что алфавитом на вход является двоичная строка.

Оригинальное (академическое) регулярное выражение выглядит так:

(0|1(01\*0)\*1)+

С использованием отрицания:

([^1]|[^0]([^1][^0]\*[^1])\*[^0])+

С использованием ленивой итерации Клини:

(0|1(01\*?0)\*?1)+

На десяти тестах програма показала следующие результаты:

Test1:

academic regex:

Total execution time: 1062868ns

negative regex:

Total execution time: 411761ns

klini regex:

Total execution time: 466417ns

Test2:

academic regex:

Total execution time: 1426387ns

negative regex:

Total execution time: 761443ns

klini regex:

Total execution time: 384342ns

Test3:

academic regex:

Total execution time: 571578ns

negative regex:

Total execution time: 2036120ns

klini regex:

Total execution time: 467763ns

Test4:

academic regex:

Total execution time: 591587ns

negative regex:

Total execution time: 1750448ns

klini regex:

Total execution time: 456342ns

Test5:

academic regex:

Total execution time: 606178ns

negative regex:

Total execution time: 762585ns

klini regex:

Total execution time: 829121ns

Test6:

academic regex:

Total execution time: 421293ns

negative regex:

Total execution time: 465342ns

klini regex:

Total execution time: 434170ns

Test7:

academic regex:

Total execution time: 829715ns

negative regex:

Total execution time: 509320ns

klini regex:

Total execution time: 569597ns

Test8:

academic regex:

Total execution time: 518417ns

negative regex:

Total execution time: 575797ns

klini regex:

Total execution time: 310960ns

Test9:

academic regex:

Total execution time: 538057ns

negative regex:

Total execution time: 475955ns

klini regex:

Total execution time: 720647ns

Test10:

academic regex:

Total execution time: 374474ns

negative regex:

Total execution time: 332755ns

klini regex:

Total execution time: 442200ns

Среднее время выражений:

academic times: 694055

negative times: 808152

klini times: 508155

Из данных тестов и среднего времени можно сделать вывод,что ленивая итерация Клини в среднем работает быстрее, на 2 месте оказалось академическое выражение,а на последнем месте оказалось регулярное выражение с использованием отрицания.

**2.** **Вторая задача**

Задание: реализовать алгоритм решения системы регулярных выражений с беззвездными коэффициентами.

В ходе решения задачи был обнаружен баг в условной конструкции, код не правильно проверял длину одного из выражений. Баг был исправлен дополнительным условием. Так же был баг с когда Y = bZ + a, в котором нет переменной Y в правой части уравнения, баг был исправлен путем копирования части кода из 3 задача, где это было пофикшено.

Test1:

X = (a|b)X+bY+aab  
Y=aX + a

Ответ: X = (a|b)\*((b(a(a|b)\*)b\*((a(a|b)\*)aab+a))+aab)

Y = (a(a|b)\*)b\*((a(a|b)\*)aab+a)

Test2:

X = aY + bX + e  
Y = aZ + bY  
Z = aX + bZ

Ответ: X = b\*((ab\*((a(((ab\*)ab\*)a+b)\*((ab\*)e))))+e)

Y = b\*((a(((ab\*)ab\*)a+b)\*((ab\*)e)))

Z = (((ab\*)ab\*)a+b)\*((ab\*)e)

Test3:

X = (a|b)X +bY+aab  
Y=aZ + a

Ответ: Система не корректна

Test4:

X = a|bX+bY+aab  
Y=aX + a

Ответ: Система не корректна

Test5:

X = Y + a  
Y=aX + a

Ответ: Система не корректна

Test6:

X = (a|a)X+aaY+aaaaa  
Y=aaaaX + (a |aa)

Ответ: X = (a|a)\*((aa(aaaa(a|a)\*)aa\*((aaaa(a|a)\*)aaaaa+(a|aa)))+aaaaa)

Y = (aaaa(a|a)\*)aa\*((aaaa(a|a)\*)aaaaa+(a|aa))

Test 7:

X = (a|b)X+bY+aab  
Y=bZ + a  
Z = aZ+a

Ответ: X = (a|b)\*((b((ba\*(a))+a))+aab)

Y = ((ba\*(a))+a)

Z = a\*(a)

# 3. Третья задача (чётный вариант)

Задание: реализовать алгоритм преобразования RG в regex.

Для решения этой задачи было принято решение воспользоваться алгоритмом 2 задачи. Мы составляем систему уравнений регулярных выражений из нашей грамматики, а ловушки убираем из нашей системы. Исправлен баг с выходом из границ массив в 4 тесте с помошью дополнительного условия.

Тесты, которые прошла программа:

Test1:

N → aL L → aN M → aS  
S → bN S → aM S → b

Ответ: (aa)\*(b)

Test2:

N → aL L → aN M → aS  
S → bN S → aM S → b M → a

Ответ: (b|(a(aa)\*(a|ab)))

Test3:

N → aL L → aN L → bM M → aS  
S → bN S → aM S → b

Ответ: (ba(aa)\*ba|aa)\*(b)

Test4:

N → aL  
L → aN  
M → aL  
S → aM  
S → b  
S → a

Ответ: (a|b)