

## ОТЧЕТ ПО ДОМАШНЕЙ РАБОТЕ 3

**ЗАДАНИЕ 1.1** Пусть у нас есть слово  $a^n abb^n$ . Разобьем это слово на  $x y z$ , так как  $|x y| \leq n$ , то  $y$  имеет вид  $a^c$ . Тогда возьмем  $k = 2$ , тогда наше слово будет иметь вид  $a^{n+c} a b b^n$ . Так как у нас возможность  $u$  и  $v$  должна совпадать, то заметим, что по середине не стоит  $a b$  в полученном слове. Значит, слово не принадлежит нашему языку. Тогда исходя из леммы о накачке такой язык будет являть нерегулярным.

**ЗАДАНИЕ 1.2** Возьмем слово вида  $a^n c^{2n} e^{n-1}$ , Разобьем это слово на  $xyz$ , так как  $|xy| \leq n$ , то  $y$  имеет вид  $a^b$ . Тогда возьмем  $k = 2$ , тогда наше слово будет иметь вид  $a^{n+b} c^{2n} e^{n-1}$ . Заметим, что если слово входит в наш язык, то  $n+b+n-1+1 = 2n \rightarrow b = 0$ , но  $y$  не должно быть пустым. Тогда исходя из леммы о накачке такой язык будет являть нерегулярным.

**ЗАДАНИЕ 1.3** Рассмотрим множество, которое состоит из таких  $p$ , где  $p$  – простое и  $p+2$  тоже простое. Если множество бесконечно, то наш алфавит можно описать регулярным выражением вида " $a^+$ ". Если наше множество конечно, то возьмем  $p_{max}$ . Мы хотим, чтобы  $n \leq p_{max}$ , тогда у нас должно быть  $a$  от 1 до  $p_{max}$ . В регулярном выражении тогда это будет записываться как  $a1, p_{max}$ . Тогда в любом случае мы можем построить регулярное выражение, следовательно, наш язык регулярный.

**ЗАДАНИЕ 2.1** Тесты описаны в файле `main.cpp`. Реализация парсера в файлах `parser.cpp`. Тесты для проверки работы парсера закомментированы. Проверяют работу каждого отдельного оператора, а так же их совместную работу.

Рассмотрим различие время работы на примере регулярного выражения:  $a * (d|c)$  и строки "aaaaaaaaad".

Время работы без оптимизации: 461.785 ms

Время работы с оптимизацией: 1.70363 ms. Происходит это из-за того, что каждый раз, когда мы матчим по звездочке, наша регулярка разрастается в экспоненциальном размере, таким образом при увеличении количество  $a$  время работы сильно ухудшается.

Регулярное выражение  $a * d^*$ , строка `aaaaaaaaaaddddd`

Время работы без оптимизации: Не вычисляется даже за несколько секунд

Время работы с оптимизацией: 3.06496 ms.

Тест на котором работает долго даже с нашими оптимизациями: регулярное выражение `a*d*c*`, строка "aaaaaaaaaaaaaaaaaaaaadddddddddddddddcccccccccccccccccccccccc".

Время работы с оптимизацией: 10545.7 ms это 10 s.

Время работы такое большое, так как на каждой звездочке наша регулярка увеличивается

в два раза.

## Исправление

**Задание 1.2** Возьмем слово вида  $a^n c^{2n+1} e^n$ , Разобьем это слово на  $xyz$ , так как  $|xy| \leq n$ , то  $y$  имеет вид  $a^b$ . Тогда возьмем  $k = 2$ , тогда наше слово будет иметь вид  $a^{n+b} c^{2n+1} e^n$ . Заметим, что если слово входит в наш язык, то  $n + b + n + 1 = 2n + 1 \rightarrow b = 0$ , но  $y$  не должно быть пустым. Тогда исходя из леммы о накачке такой язык будет являть нерегулярным.

\*Изменила  $n - 1$  вхождений  $e$  до  $n$ , теперь подходит для любого  $n^*$