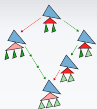


## Лабораторная работа 2

- ❶ (общее) 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  $10^5$  символов.
- ❷ Реализовать алгоритм решения системы регулярных выражений с беззвездными коэффициентами.
- ❸ (a) Реализовать алгоритм преобразования RG в regex (чет в списке группы).
- ❹ (b) Реализовать алгоритм преобразования DFA в regex (нечет в списке группы).



## Синтаксис входных данных

Здесь и на следующем слайде чёрным обозначены элементы метаязыка, красным — элементы языка входных данных. Т.е. запись вида **(|)** обозначает альтернативу между закрывающей и открывающей скобками, а вида **(||)** — слово, состоящее из двух значков альтернативы. Чтение данных осуществляется из файла.

Синтаксис входных данных для решения системы уравнений:

$\langle \text{system} \rangle$	$::=$	$\langle \text{equation} \rangle^+$
$\langle \text{equation} \rangle$	$::=$	$\langle \text{var} \rangle = (\langle \text{regex} \rangle \langle \text{var} \rangle (+ \langle \text{regex} \rangle \langle \text{var} \rangle)^* +)? \langle \text{regex} \rangle$
$\langle \text{regex} \rangle$	$::=$	$(\langle \text{expr} \rangle   \langle \text{alt.regex} \rangle)   \langle \text{expr} \rangle$
$\langle \text{alt.regex} \rangle$	$::=$	$\langle \text{expr} \rangle   \langle \text{alt.regex} \rangle   \langle \text{expr} \rangle$
$\langle \text{expr} \rangle$	$::=$	$\langle \text{letter} \rangle \langle \text{expr} \rangle   \langle \text{letter} \rangle$
$\langle \text{var} \rangle$	$::=$	$[A-Z]$
$\langle \text{letter} \rangle$	$::=$	$[a-z]$



## Синтаксис входных данных

Синтаксис RG — каждое правило с новой строки, стартовый нетерминал всегда S.

```
 $\langle \text{grammar} \rangle ::= \langle \text{rule} \rangle^+$   
 $\langle \text{rule} \rangle ::= \langle \text{nterm} \rangle \rightarrow (\langle \text{letter} \rangle \langle \text{nterm} \rangle \mid \langle \text{letter} \rangle)$   
 $\langle \text{nterm} \rangle ::= [\text{A-Z}]$   
 $\langle \text{letter} \rangle ::= [\text{a-z}]$ 
```

Синтаксис DFA описан ниже. Первое состояние в списке — начальное. Список состояний в фигурных скобках задаёт множество конечных состояний. Список всех состояний явно не перечисляется, он восстанавливается по правилам перехода.

```
 $\langle \text{DFA} \rangle ::= \langle \text{state} \rangle, \{ \langle \text{transition} \rangle^+ \}, \{ \langle \text{state} \rangle (, \langle \text{state} \rangle)^* \} \rangle$   
 $\langle \text{transition} \rangle ::= \langle \text{state} \rangle, \langle \text{letter} \rangle, \langle \text{state} \rangle$   
 $\langle \text{state} \rangle ::= [\text{A-Z}][0-9]?$   
 $\langle \text{letter} \rangle ::= [\text{a-z}]$ 
```



## Пример входных данных

Пример корректной системы уравнений приведён ниже.  
Расстановка пробелов свободная.

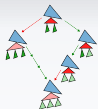
$$X = (a|b)X + bY + aab$$

$$Y = aX + a$$

Уравнение  $Z = (a|(b|a))X$  должно распознаваться как некорректное: каждый коэффициент может иметь только один уровень вложенности скобок. Некорректно и уравнение  $Z = a|b Z$ , потому что в нём нет скобок вокруг альтернативы. Уравнение  $X = Y + a$  некорректно, потому что все коэффициенты должны быть не пусты (в том числе в альтернативе). Нижеследующая система тоже некорректна — в ней нет уравнения на  $Z$ .

$$X = (a|b)X + bY + aab$$

$$Y = aZ + a$$



## Пример входных RG и DFA

В RG могут быть нетерминалы-ловушки, не выходящие на останов. Например,

$$\begin{array}{lll} S \rightarrow bN & S \rightarrow aM & S \rightarrow b \\ N \rightarrow aL & L \rightarrow aN & M \rightarrow aS \end{array}$$

Здесь N, L — ловушки. Аналогичная ситуация может быть и в DFA:

$$\langle Q_0, \{ \langle Q_0, a, Q_0 \rangle \langle Q_0, b, Q_1 \rangle \langle Q_1, a, Q_1 \rangle \langle Q_1, b, Q_2 \rangle \langle Q_2, a, Q_2 \rangle \langle Q_2, b, Q_2 \rangle \}, \{ Q_1 \} \rangle$$

Здесь ловушка — состояние Q2. В DFA нужно проверить наличие переходов по всем буквам из всех состояний.

Алфавит языка DFA восстанавливается как множество всех букв, по которым есть переходы хотя бы из одного состояния.



## Отчёт по ЛР

---

В отчёте должны быть описаны:

- краткое обоснование эквивалентности трёх регулярных выражений;
- способ порождения тестов для регулярных выражений;
- время распознавания тестов (в виде таблицы) для трёх версий регулярных выражений;
- кратко — возникшие сложности при реализации 2 и 3 задачи и способ их решения.



---

## Доп.баллы +1

- Чётные: удаление всех избыточных скобок из результирующего регулярного выражения.
- Нечётные: на вход алгоритма может приниматься NFA, а не только DFA. Выводить об этом предупреждение при анализе, но регулярное выражение всё равно строить.