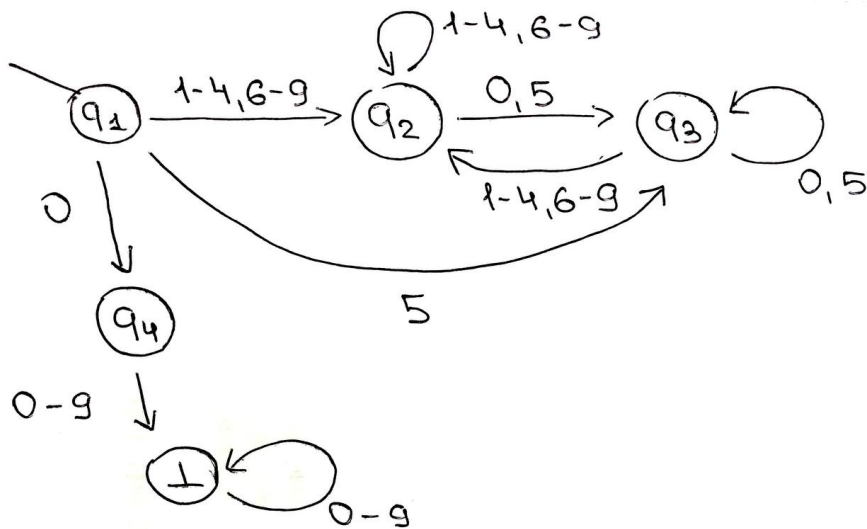


# Формальные языки

Евгения Ву

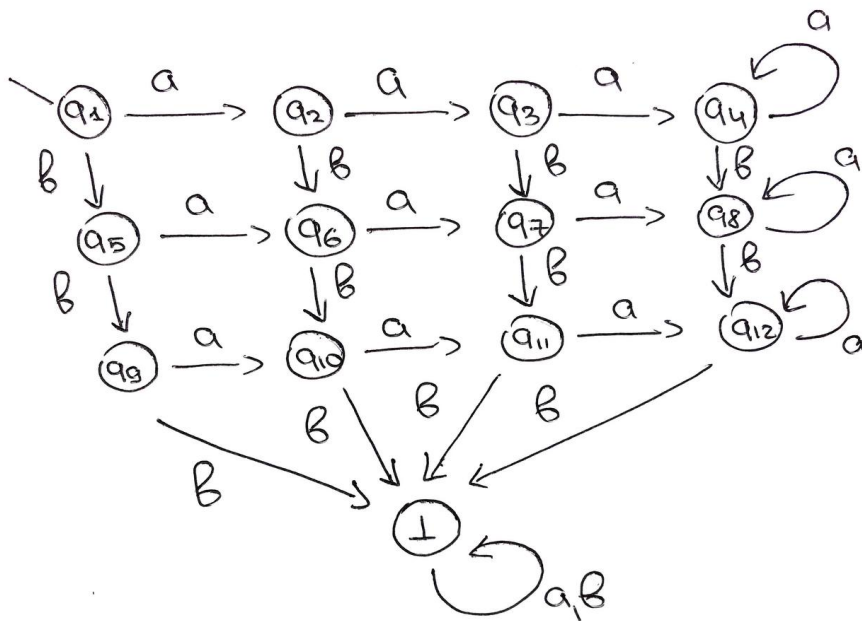
13 сентября 2021 г.

## Задание 1



Заметим, что в данном автомате  $q_1$  – начальное состояние, а  $q_3$  и  $q_4$  – терминальные состояния.

## Задание 2



Заметим, что в данном автомате  $q_1$  – начальное состояние, а  $q_4$ ,  $q_8$  и  $q_{12}$  – терминальные состояния.

### Задание 3

- 1) Мы можем использовать псевдонимы типа [https://en.cppreference.com/w/cpp/language/type\\_alias](https://en.cppreference.com/w/cpp/language/type_alias)
- 2) Существует негатив функции, который выдает дополнение к объявленной [https://en.cppreference.com/w/cpp/utility/functional/not\\_fn](https://en.cppreference.com/w/cpp/utility/functional/not_fn)
- 3) Если хочешь задать несколько шаблонных значений, но не знаешь сколько, то можно использовать многоточие [https://en.cppreference.com/w/cpp/language/template\\_parameters](https://en.cppreference.com/w/cpp/language/template_parameters)

## Задание 4

Рассмотрим, что нам нужно для записи автомата в виде графа: алфавит, на котором построен автомат, состояние – вершины (с характеристикой какое оно: терминальное, начальное или промежуточное), и функции перехода (ребра).

Тогда, чтобы описать какой-нибудь автомат на нашем языке нам потребуется:

В первой строчке через пробел указать алфавит, на котором строится автомат  
 Далее начиная с каждой новой строки, опишем каждое состояние в виде:  $q_{num} \text{ state}$ , где  $q$  означает, что далее последует описание вершины, вместо  $num$  требуется указать номер данного состояния, вместо  $state$  – какого типа наше состояние:  $s$  – начальное,  $t$  – терминальное,  $n$  – обычное.

После этого будем описывать функции перехода (ребра), они будут записываться в виде:  $e \ q_{out} \ q_{in} \ symbols$ , где  $e$  означает что последует описание ребра, далее  $q$  – что это какая-то вершина, вместо  $out$  указать номер состояния, откуда действует функция. Аналогично описываем вторую вершину – состояние, куда приходит функция, после этого  $symbols$  – список символов алфавита, по которым мы можем осуществить данный переход, если символов много, то передаем их через запятую

**Пример 1.** Автомат, определяющий натуральное ли это число без ведущих нулей. (0 – не натуральное)

```
0 1 2 3 4 5 6 7 8 9
q1 s
q2 t
q3 t
q4 n
e q1 q2 {1, 2, 3, 4, 5, 6, 7, 8, 9}
e q1 q3 {0}
e q2 q2 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
e q3 q4 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
e q4 q4 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

**Пример 2.** Автомат из первого с лекции, определяющий нечетное количество нулей и единиц в числе состоящем только из нулей и единичек.

```
0 1
q1 s
q2 n
q3 n
q4 t
e q1 q2 {0}
e q2 q1 {0}
```

```
e q1 q3 {1}
e q3 q1 {1}
e q4 q3 {0}
e q3 q4 {0}
e q2 q4 {1}
e q2 q4 {1}
```

**Пример 3.** Автомат, определяющий, состоит ли наше слово из других символов, кроме 1, на алфавите 0 и 1

```
0 1
q1 s
q2 t
q3 n
e q1 q2 {1}
e q2 q2 {1}
e q1 q3 {0}
e q2 q3 {0}
e q3 q3 {0}
```

## Задание 5

В данном задании я использовала среду разработки **sublime text**. Чтобы настроить подсветку я воспользовалась [этим туториалом](#). В моей задумке было подсветить мой язык описания dfa. Так, розовым подсвечивается  $e$  – обозначение, что дальше будет идти описание функции перехода, оранжевым подсвечиваются элементы нашего алфавита, как в самом перечислении, так и в описании, по каким символам происходит переход, и зеленым подсвечивается название состояния с его характеристикой (какое это состояние)

Файл примера кода и самой подсветки синтаксиса прикреплен