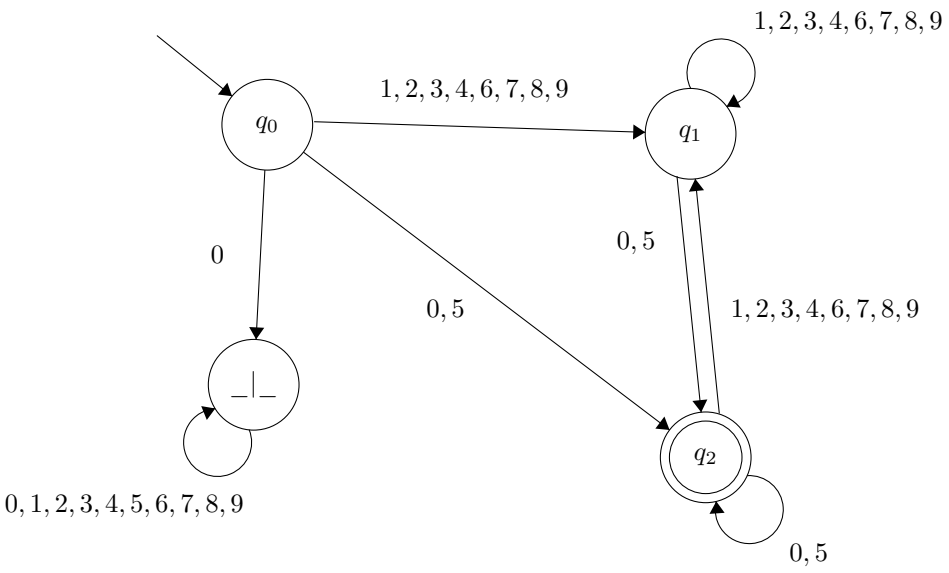
# Домашняя работа по формальным языкам - 1

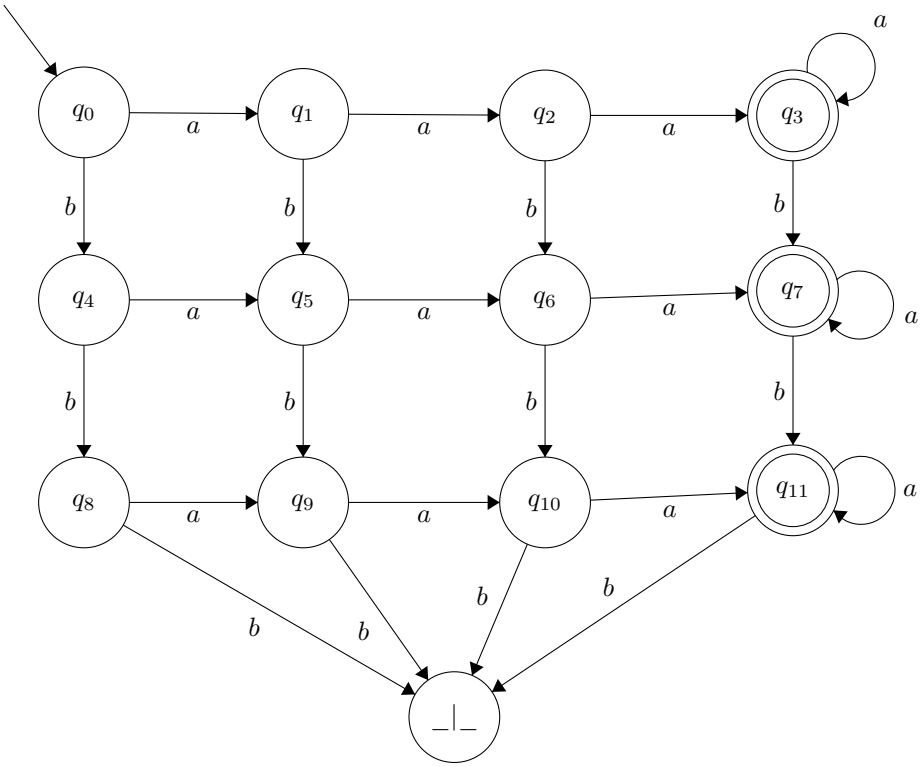
Полупанова Анна

13.09.21

## Задание №1



## Задание №2




## Задание №3

Моим вторым любимым языком программирования после c++ является python. В целом, в языке python много хороших понятных примеров необычного синтаксиса, в отличие от c++, где все гораздо сильно более формализовано. Одна из интересный вещей в python – это генерация списков (list comprehension), ниже несколько её особенностей, которые я узнала:

1. Генераторы можно прописывать не только в теле создаваемого списка, но и создавать отдельно. При создании генератора им можно воспользоваться только единожды, однако, можно создавать генератор, больше похожий на функцию, использующий ключевое слово "yeild"вместо "return". Например:

```
        def createGenerator() :
...     mylist = range(3)
...     for i in mylist :
...         yield i*i
...
```

```
>>> mygenerator = createGenerator() # создаем генератор
>>> print(mygenerator) # mygenerator является объектом!
<generator object createGenerator at 0xb7555c34>
>>> for i in mygenerator:
...     print(i)
```

Здесь применяем генератор не в list, но можно и там. Прочесть про это можно здесь. Красивый пример был найден здесь

2. С помощью конструкции "yield from"можно создавать генераторы, вызываемые из генератора, то есть субгенераторы. Это позволяет упрощать различные конструкции, где используется цикл for, а также создавать более сложные генераторы. Прочитать про это можно по ссылке из первого пункта. Пример:

```
# Обычный yield
def numbers_range(n):
    for i in range(n):
        yield i
#yield from
def numbers_range(n):
    yield from range(n)
```

## Задание №4

Опишем автомат его основными признаками: зададим алфавит, начальную вершину, терминальные состояния, переходы. На первой строке через 1 пробел зададим все символы алфавита в квадратных скобках, затем строкой "start имя вершины"зададим имя начальной вершины, затем со следующей строки записываем все терминальные вершины в формате "terminal имя вершины каждую на новой строке, затем записываем все переходы в формате "transition (имя начальной вершины -> имя конечной вершины, {символы по которым происходит переход через запятую}) каждый переход на новой строке.

Примеры:

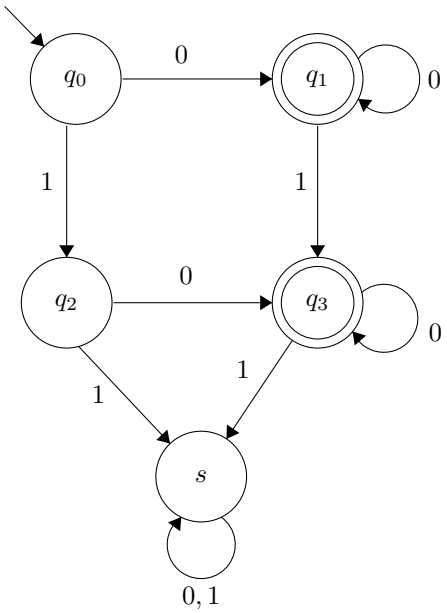
1. Автомат, распознающий язык неотрицательных чисел без лидирующих нулей

```
[0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
start q0
terminal q1
transition (q0 -> s, {0})
transition (s -> s, {0,1,2,3,4,5,6,7,8,9})
transition (q0 -> q1, {1,2,3,4,5,6,7,8,9})
transition (q1 -> q1, {0,1,2,3,4,5,6,7,8,9})
```

2. Автомат, распознающий язык неотрицательных чисел без лидирующих нулей, делящихся на 5 (см. задание №1)

```
[0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
start q0
terminal q2
transition (q0 -> s, {0})
transition (s -> s, {0,1,2,3,4,5,6,7,8,9})
transition (q0 -> q1, {1,2,3,4,6,7,8,9})
transition (q0 -> q2, {0,5})
transition (q1 -> q1, {1,2,3,4,6,7,8,9})
transition (q1 -> q2, {0,5})
transition (q2 -> q1, {1,2,3,4,6,7,8,9})
transition (q2 -> q2, {0,5})
```

3. Автомат, распознающий язык строк над алфавитом 0, 1 (битовых последовательностей), в которых 0 не меньше одного, а 1 не больше одной.




```
[0] [1]
start q0
terminal q1
terminal q3
transition (q0 -> q1, {0})
transition (q0 -> q2, {1})
transition (q1 -> q1, {0})
```

```
transition (q1 -> q3, {1})
transition (q2 -> q3, {0})
transition (q2 -> s, {1})
transition (q3 -> q3, {0})
transition (q3 -> s, {1})
transition (s -> s, {0,1})
```

## Задание №5

Для языка, описанного в предыдущем задании, я задавала лексическую подсветку в vim. Скрипт, задающий подсветку, можно посмотреть в приложенном файле. Подсвечивает ключевые слова: start, terminal, transition, имена переменных, заданные после слов start и terminal, слова алфавита при их задавании в квадратных скобках.

Как запустить:

1. Создать $.vim/syntax/$, положить туда файл с описанием подсветки.
2. Создать $.vim/ftdetect/$, положить туда файл с таким же названием, как и файл с описанием подсветки, вписать туда строку $auBufRead, BufNewFile*.txtsetfiletype = syn$. Создать в директории $.vim$ файл $filetype.vim$ и вписать туда то же самое.
3. подсветка подключится и будет работать в файлах расширения txt