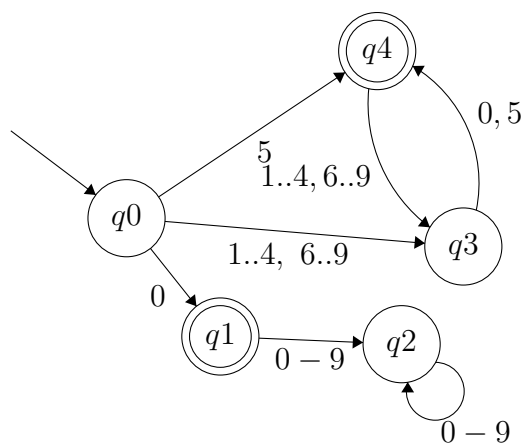


# Домашняя работа по алгебре

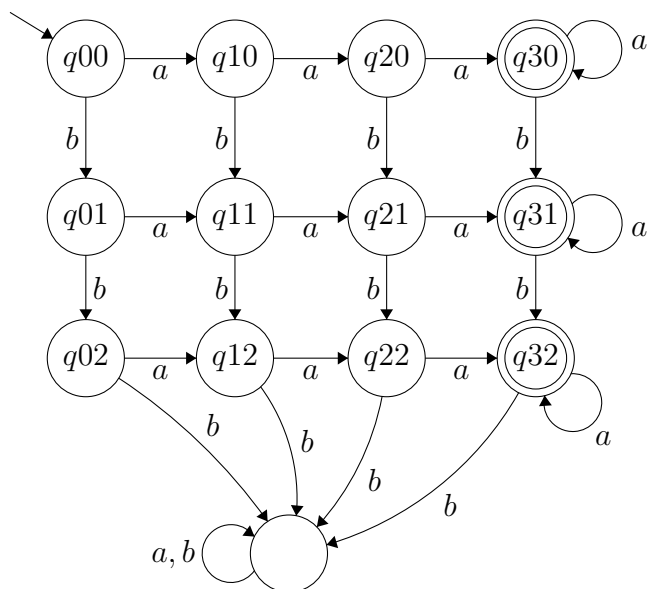
Дымашевский Сергей

13 сентября 2021 г.

1



2



3

Язык - питон.

В питоне в начале файла можно указать кодировку с помощью комментария.

Также в питоне можно делать переносы строк с помощью `\`, при этом она не разъединяет токены, кроме строковых литералов

Еще в питоне есть мнимые литералы для мнимых чисел

Остальные интересности питона, кажется, не относятся к лексическому синтаксису, который я смотрел [здесь](#)

## 4

Мы знаем, что каждый FDA задается пятью вещами, это:

1.  $\Sigma$  - алфавит
2.  $Q$  - множество состояний
3.  $q_0$  - стартовое состояние
4.  $T$  - терминалы
5.  $\delta$  - функция перехода между состояниями

Тогда давайте слово в нашем языке, описывающем автоматы и будет устроено так:

*Sigma* : "symbol1, symbol2..."

*Q* : " $q_1, q_2, \dots$ "

*q0* : " $q_0$ "

*Terminals* : " $t_1, t_2, t_3$ "

*delta* : ' $(q_1, s_1) \rightarrow q_{1s_1}, \dots$ '

где каждый параметр задается в одинаковом формате, как в дальнейших примерах. То есть мы просто задаем слова в нашем языке как вещи, определяющие конечный автомат. Язык наш будет над алфавитом всех символов, которые мы можем встретить в автоматах которые мы хотим описывать. Примеры автоматов на таком языке (в файлах оставляю их копию в виде кода):

1. *Sigma* : "0, 1"

*Q* : " $q_0, q_1$ "

*q0* : " $q_0$ "

*Terminals* : " $q_1$ "

*delta* : ' $(q_0, 1) \rightarrow q_1, (q_0, 0) \rightarrow q_0, (q_1, 1) \rightarrow q_1, (q_1, 0) \rightarrow q'_0$ '

Автомат на строках из 0 и 1, определяющий слова, кончающиеся на 1

2. *Sigma* : "a, b, c"

*Q* : " $q_0, q_1, q_2, q_3$ "

*q0* : " $q_0$ "

*Terminals* : " $q_3$ "

*delta* : ' $(q_0, a) \rightarrow q_1, (q_0, b) \rightarrow q_0, (q_0, c) \rightarrow q_0, (q_1, a) \rightarrow q_1, (q_1, b) \rightarrow q_2, (q_1, c) \rightarrow q_0, (q_2, a) \rightarrow q_1, (q_2, b) \rightarrow q_0, (q_2, c) \rightarrow q_3, (q_3, a) \rightarrow q_3, (q_3, b) \rightarrow q_3, (q_3, c) \rightarrow q'_3$ '

Автомат для строчек из букв a, b, c определяющий строчки в которых есть подстрока abc

3. (Честно говоря сложные автоматы можно помереть так описывать и потом проверять их тоже не очень, поэтому третий тоже простой)

*Sigma* : "a, b, c"

*Q* : " $q_0, q_1, q_2$ "

*q0* : " $q_0$ "

*Terminals* : " $q_0$ "

*delta* : ' $(q_0, a) \rightarrow q_0, (q_0, b) \rightarrow q_0, (q_0, c) \rightarrow q_1, (q_1, a) \rightarrow q_2, (q_1, b) \rightarrow q_2, (q_1, c) \rightarrow q_0, (q_2, a) \rightarrow q_2, (q_2, b) \rightarrow q_2, (q_2, c) \rightarrow q'_2$ '

Автомат для строк из букв abc, в которых буква c встречается только четное число раз подряд.