

Курс: Формальные языки
Домашнее задание 1

1.

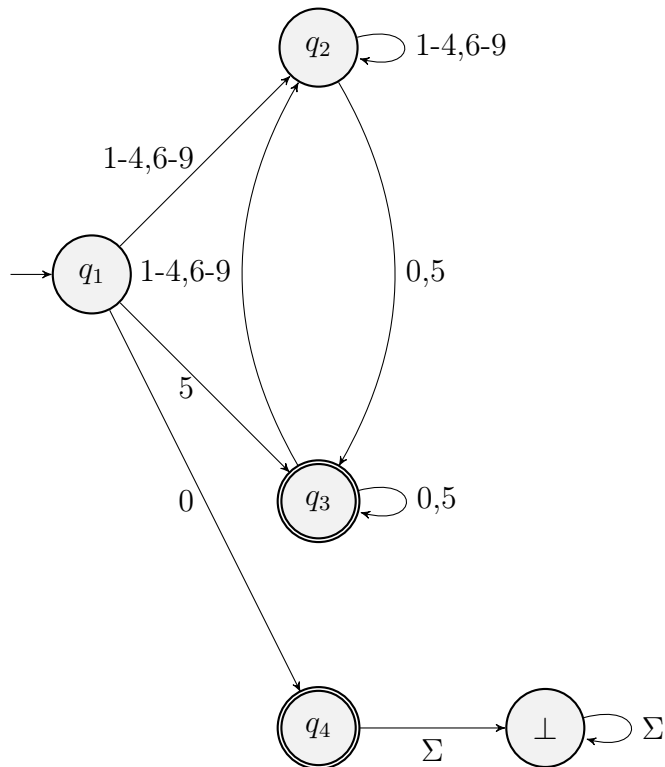


Рис. 1: задание 1

2.

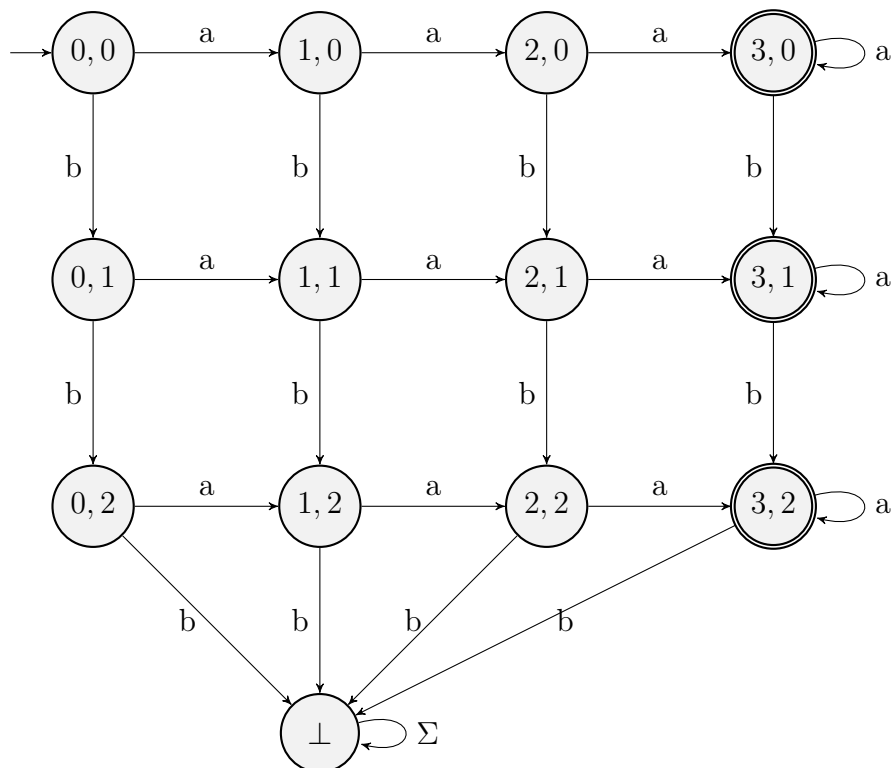


Рис. 2: задание 2

3. Я рассмотрю язык Scala, документация по лексическому синтаксису языка приведена по [ссылке](#). Я нашел интересную **особенность** в идентификаторах которая вытекает из совместимости с языком Java: есть список слов, которые конфликтуют с занятыми в языке Scala, например `Thread.yield()` не распознается, однако вместо этого можно обернуть такие идентификаторы в обратные кавычки, то есть `Thread.`yield`()`
4. Мы знаем, что такой автомат задается набором $\Sigma, Q, q_0, T, \delta$, поэтому язык описывается ровно этим набором. Рассмотрим, например, все конечные детерминированные автоматы над алфавитом $\Sigma = \{0, 1\}$, у которых три состояния.

Исправление Будем такой автомат кодировать строкой из набора элементов, составляющих алфавит, затем набор состояний Q , после начальное состояние q_0 , набор терминальных состояний T , и набор ребер перехода между состояниями δ , каждое из которых задается набором состояния, из которого есть переход, затем новое состояние и элементы алфавита, по которым можно перейти из такого состояния.

- а) автомат зададим строкой $(0, 1), (q_1, q_2, q_3), q_1, (q_1, q_2), [(q_1, q_2, (0, 1)), (q_2, q_3, (0, 1)), (q_3, q_3, (0, 1))]$

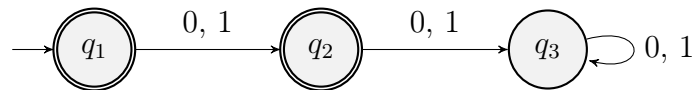


Рис. 3: автомат, принимающий все слова длины не больше 1

- б) автомат зададим строкой $(0, 1), (q_1, q_2, q_3), q_1, (q_2), [(q_1, q_2, (0, 1)), (q_2, q_3, (0, 1)), (q_3, q_3, (0, 1))]$

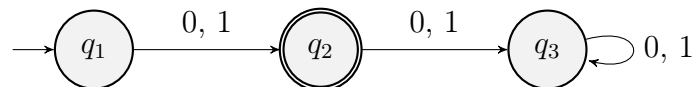


Рис. 4: автомат, принимающий все слова длины ровно 1

- с) автомат зададим строкой $(0, 1), (q_1, q_2, q_3), q_1, (q_3), [(q_1, q_2, (0)), (q_1, q_3, (1)), (q_2, q_2, (0, 1)), (q_3, q_3, (1)), (q_3, q_2, (0))]$

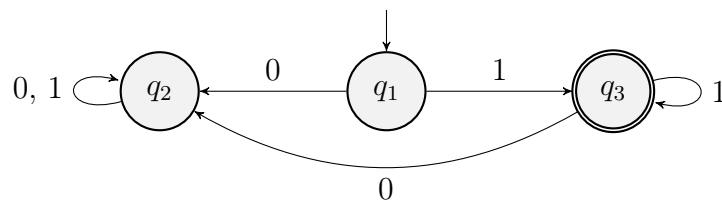


Рис. 5: автомат, принимающий только слова, состоящие только из символа 1