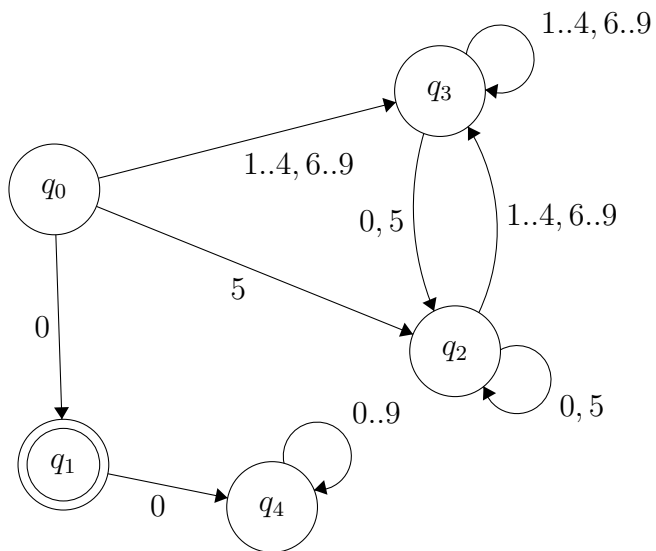
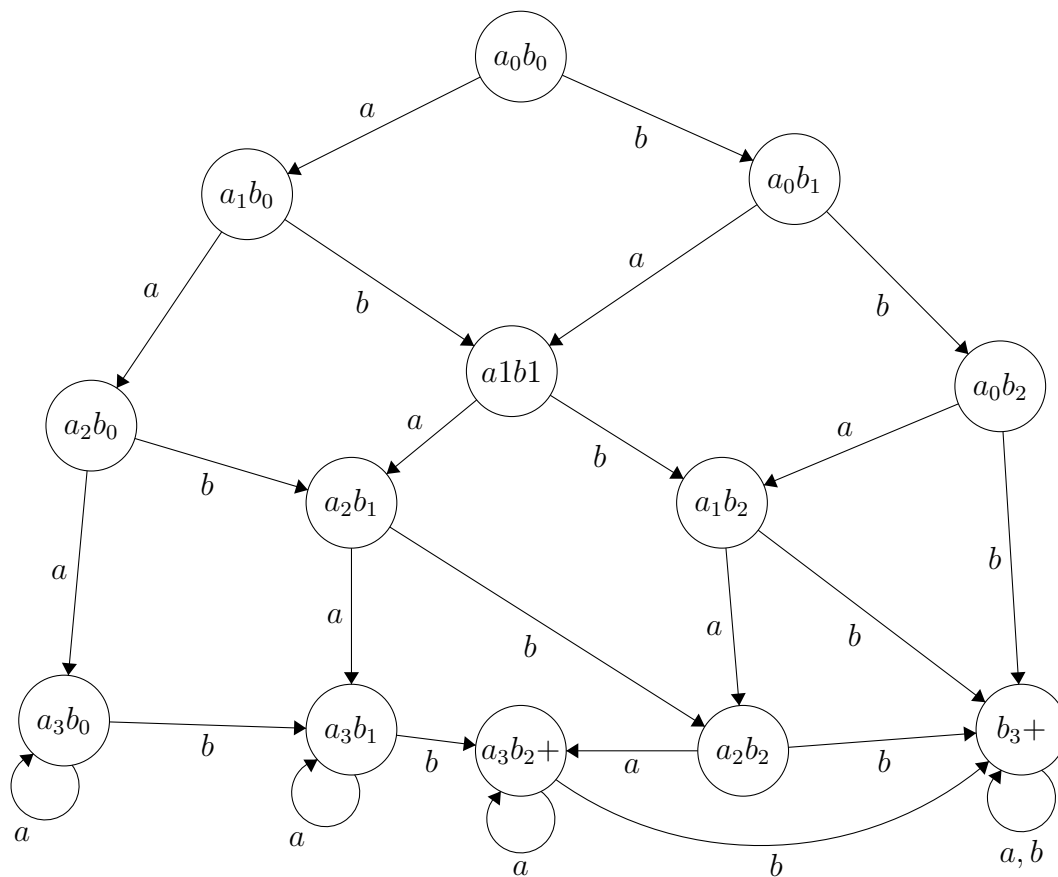


(1)



Там в  $q_2$  терминальное состояние.

(2)



В  $a_3b_{2+}$  терминальное состояние.

(3)

(а) В Python не используются символы "?" и "\$", их использование вне строки вызывает ошибку. Зато используется "@" в декораторах и даже есть конструкция "@= если

перегрузить "@" как оператор.

- (b) Специальные литералы для комплексных чисел

```
3.14e-10j
```

- (c) Синтаксис лямбд

```
f = lambda x, y: x * y
```

- (d) Синтаксис тернарного оператора

```
b = x if condition else y
```

- (e) Синтаксис assert

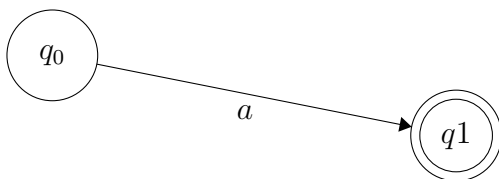
```
assert len(arr) >= 0, "размер массива не может быть отрицательным"
```

(4)

В первой строке после слова "alphabet" и двоеточия идет алфавит в фигурных скобках через запятую. Далее в каждой новой строчке идет название вершины, двоеточие и все ребра, которые из нее исходят в формате  $q_0 \{q_1 \text{ with } a, q_1 \text{ with } b\}$  - значит, что из  $q_0$  можно перейти в  $q_1$  по ребру с символом а и по ребру с символом b. Если состояние терминальное, добавим перед названием вершины слово terminal.

Примеры:

(a)

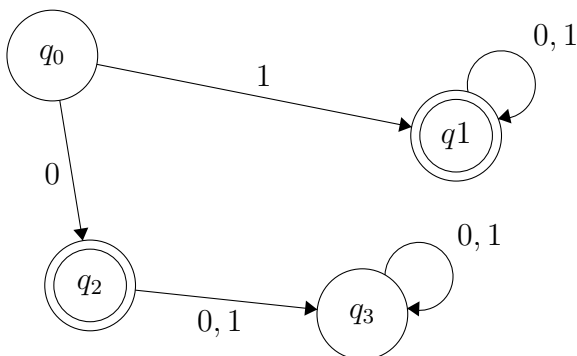


$alphabet : \{a\}$

$q_0 : \{q_1 \text{ with } a\}$

terminal  $q_1 : \{\}$

(b)



$alphabet : \{0,1\}$

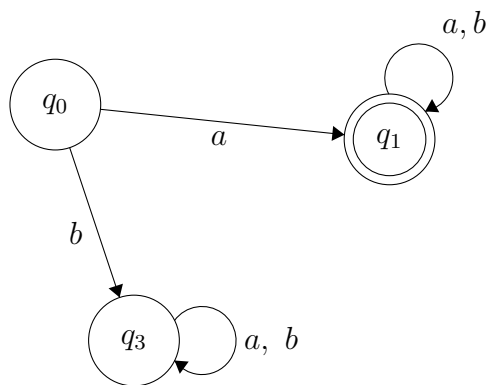
$q_0: \{q_1 \text{ with } 1, q_2 \text{ with } 0\}$

terminal  $q_1: \{q_1 \text{ with } 1, q_1 \text{ with } 0\}$

*terminal*  $q_2$ :  $\{q_3 \text{ with } 1, q_3 \text{ with } 0\}$

$q_3$ :  $\{q_3 \text{ with } 1, q_3 \text{ with } 0\}$

(с)



*alphabet* :  $\{a, b\}$

$q_0$ :  $\{q_1 \text{ with } a\}$

*terminal*  $q_1$ :  $\{q_1 \text{ with } a, q_1 \text{ with } b\}$

$q_3$ :  $\{q_3 \text{ with } a, q_3 \text{ with } b\}$

(5)

Подсветила слова *terminal*, *with*, *alphabet* и вершины, считая, что вершины и только вершины имеют название вида  $q_n$ , где n - последовательность цифр.