September 2nd

<u>Example:</u> The language of all even binary words

$\Sigma = \{0, 1\}$

$1^* \cdot 0$? $L(1^* 0) = 1 \ldots 1\, 0$ ✗

$\overset{\uparrow}{0}$ ✗

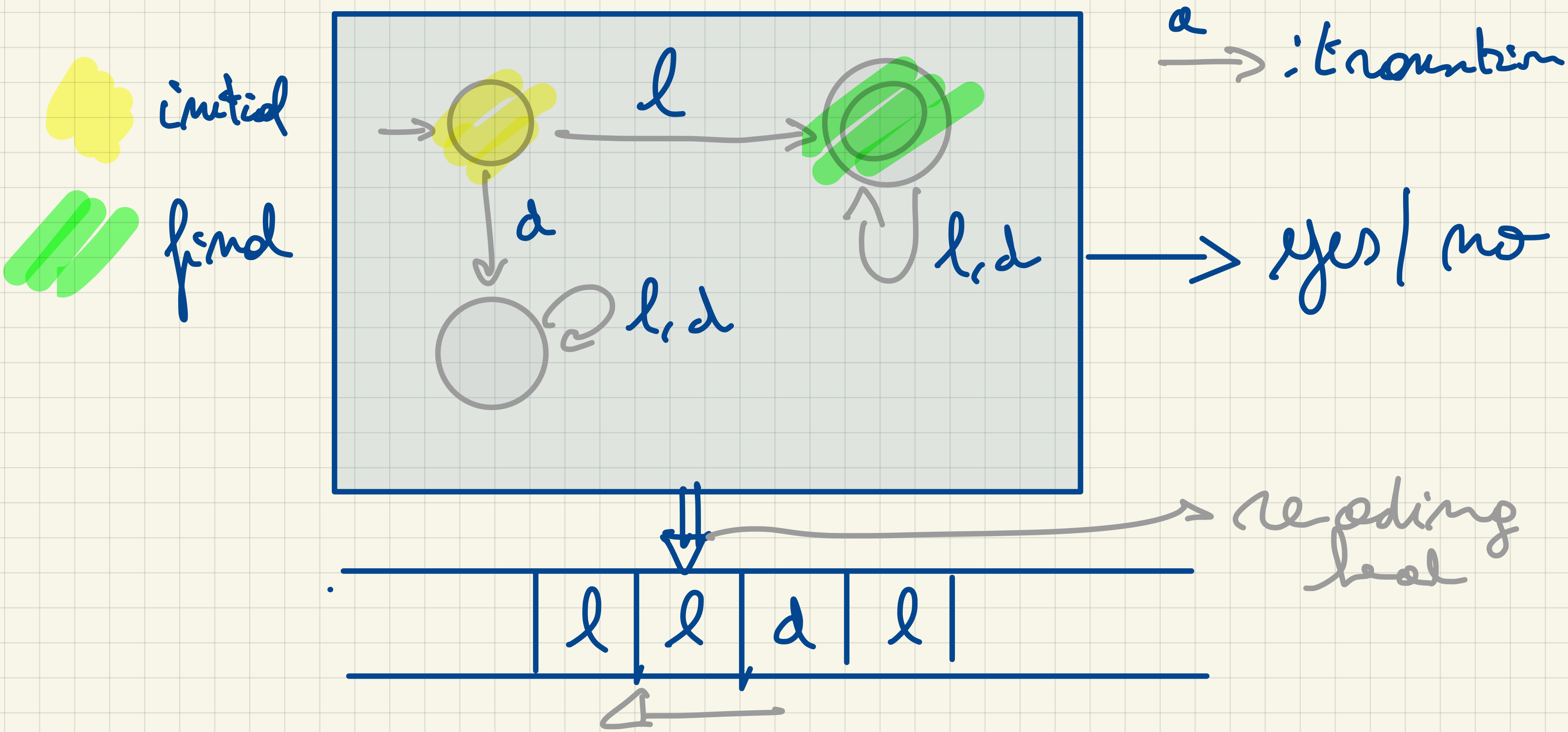
$(1+0)^* \cdot 0$ ✓

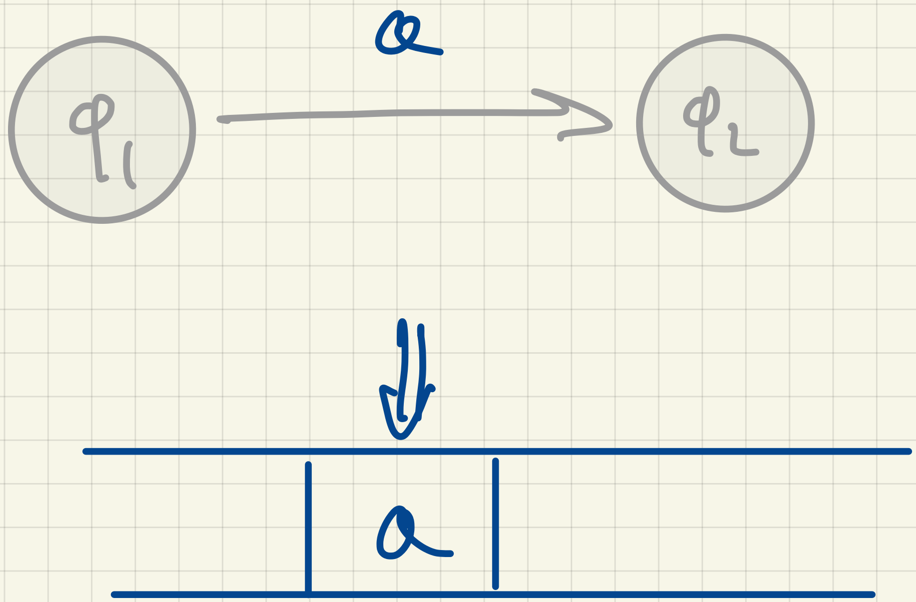
$(1+0)^*$ $\quad L((1+0)^*) = \{ 1, \ldots \}$ ✗ not even

$(0+1)^*$

Finite automata

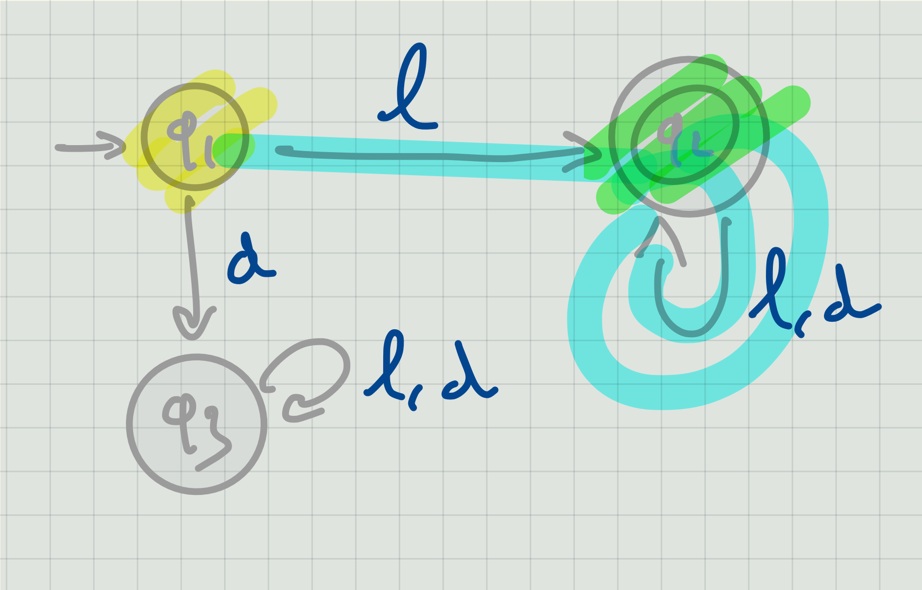
Mathematical model of the notion of computation $\rightarrow$ membership problem





We can move from q_1 to q_2 iff the reading head is reading on a

$\Sigma = \{l, d\}$



$l\,l\,d$; $\quad q_1 \xrightarrow{l} q_2 \xrightarrow{l} q_2 \xrightarrow{d} q_2$ ✓

$d\,l$; $\quad q_1 \xrightarrow{d} q_3 \xrightarrow{l} q_3$ ✗

$$l\,(l+d)^*$$

(Deterministic) Automaton → Program

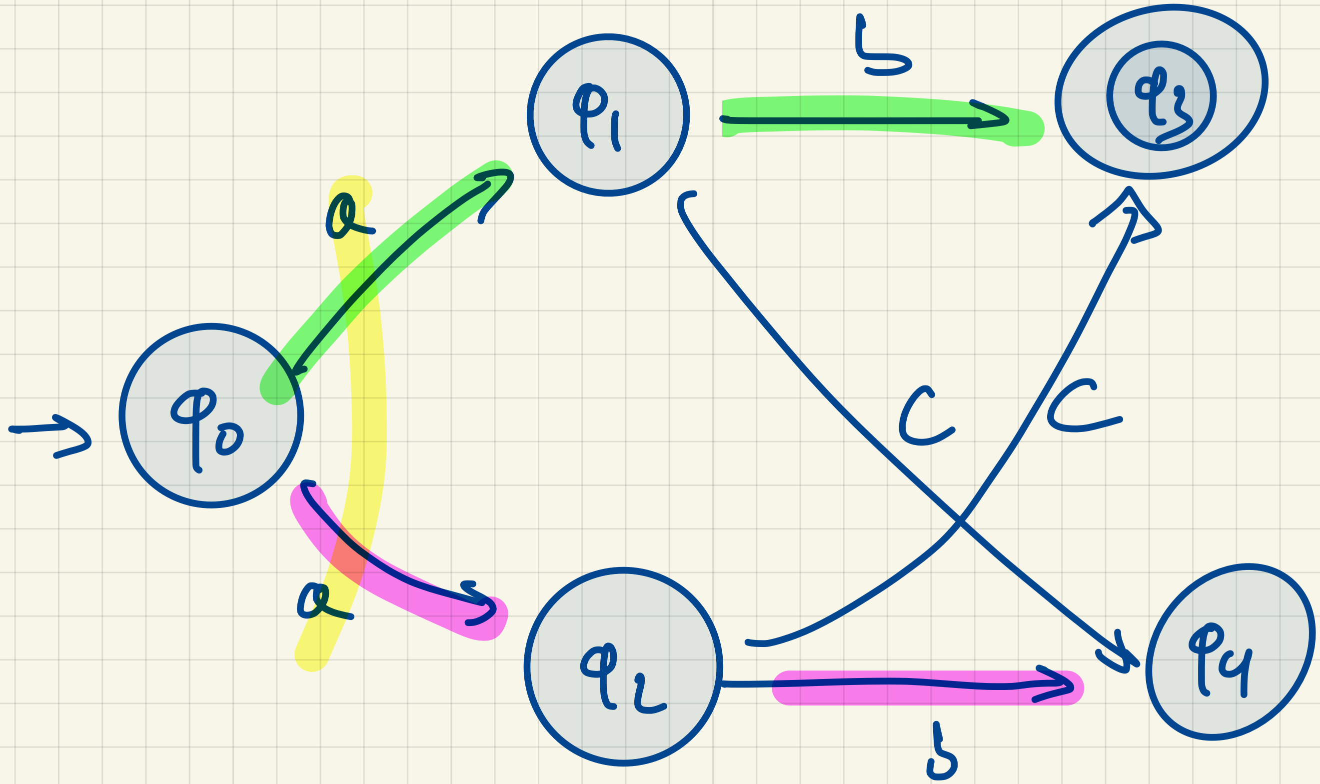
```
currentState = 1;
while (true)
{ letter = read();
  switch (currentstate)
     case 1:
        switch (letter)          input
           case l:
              currentstate = 2;
           case d:
              currentstate = 3;
   :
}
```

Another example

? ab

accepting

not accepting.



Choice when reading a from q_0

non-determinism.

In a non-deterministic automaton,

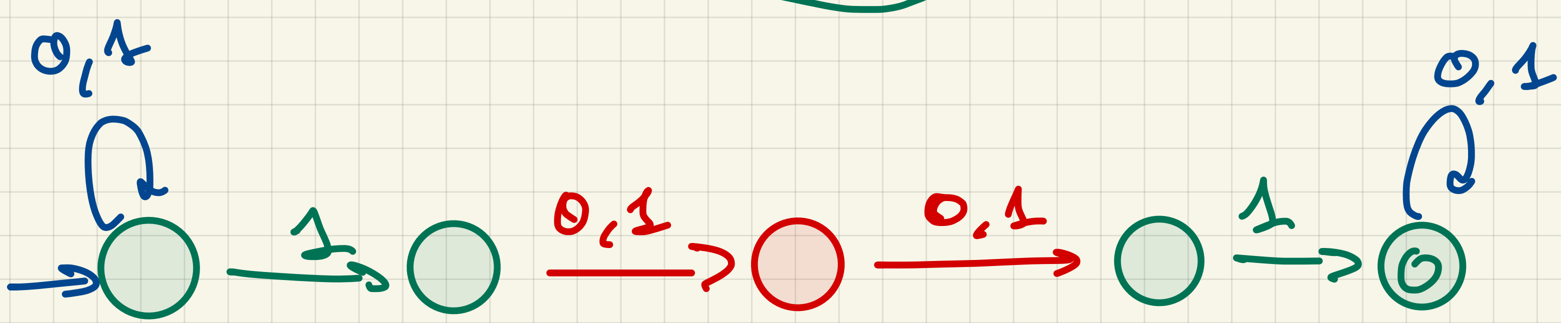
A word is accepted iff

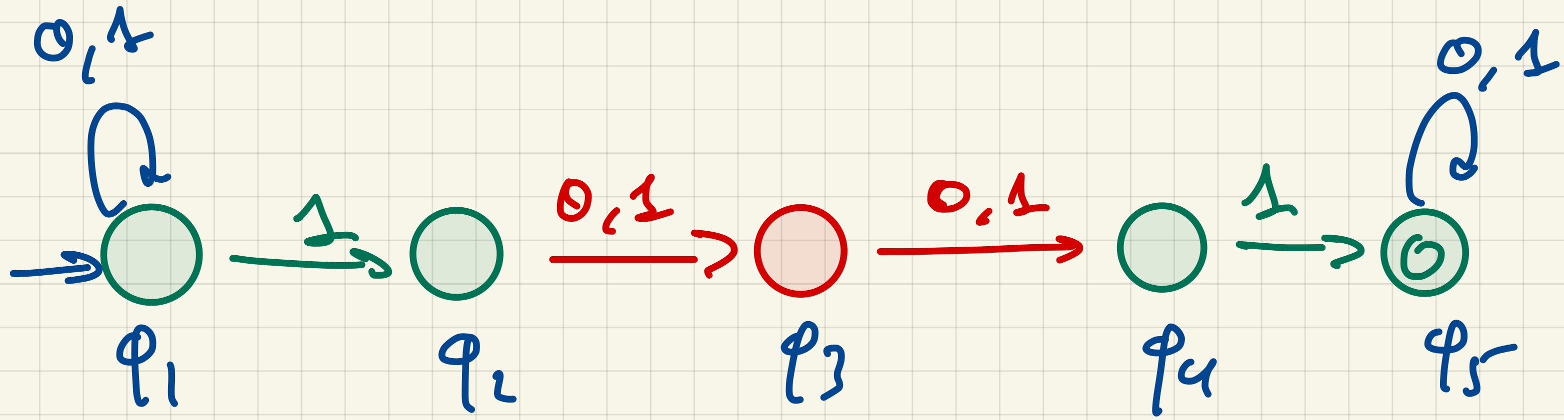
$\exists$ an accepting path

Why non-determinism?

The language of all binary words that contain 2 "1" that are separated by 2 symbols

0 1 ... 0 1 1 0 1 1 0 .. 1





1 1 0 0 1 0

$q_1 \to \cdots \to q_1$ ✗

$q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_4 \to x$ ✗

$q_1 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_5 \xrightarrow{0} q_5$ ✓

If we read an accepting state
then we read the pattern, so
the word is in the language

If the word has the pattern, we
can accept it by building
the proper path.

Spontaneous transitions

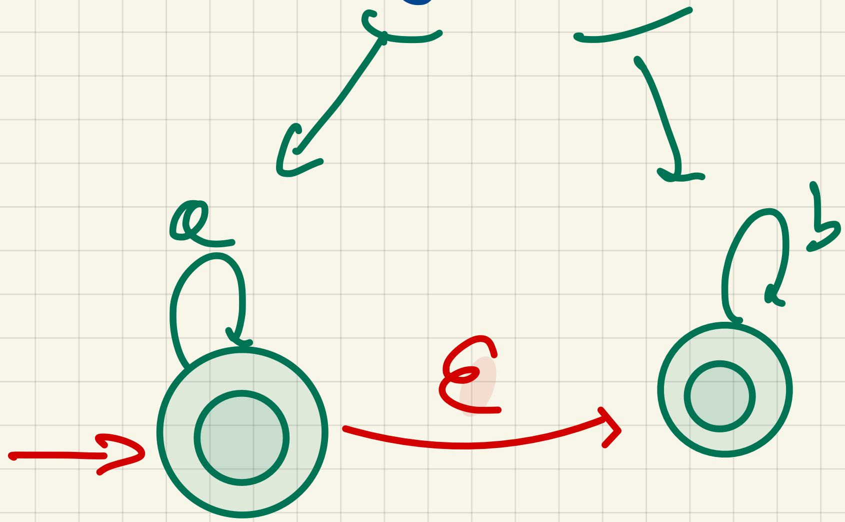
$$q_1 \xrightarrow{\varepsilon} q_2$$

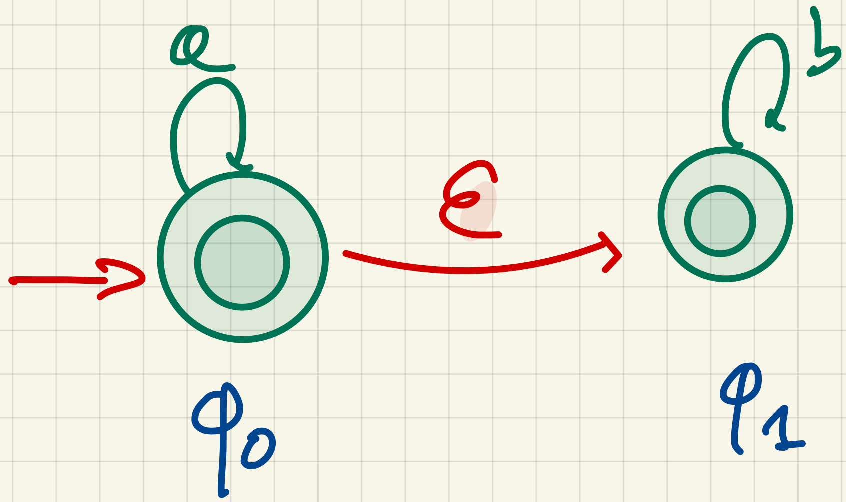
We can change from q_1 to q_2 without reading a letter.

why ???

I want to build an automaton for

$$a^* \quad b^*$$





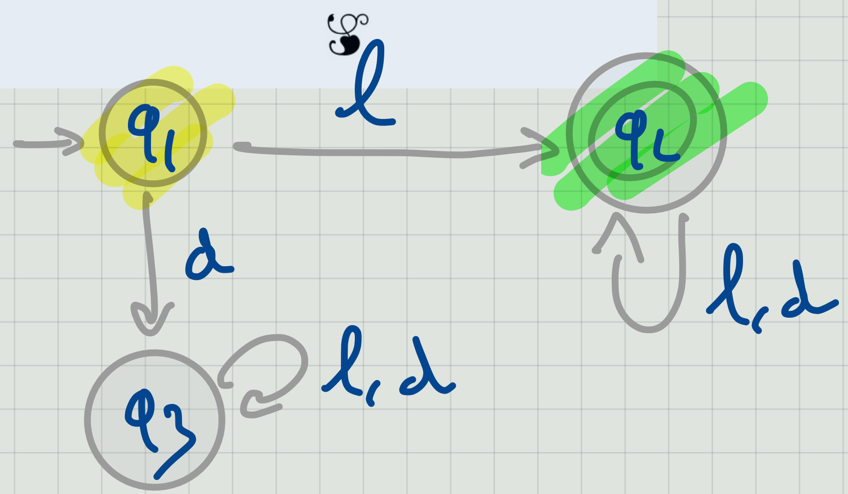
b : $q_0 \xrightarrow{\varepsilon} q_1 \xrightarrow{b} q_1$ ✓

Definition 2.5 (Finite automaton)**.** A finite automaton is a tuple:

$$A = \langle Q, \Sigma, \delta, q_0, F \rangle$$

where:

1. Q is a finite set of states;
2. Σ is the (finite) input alphabet;
3. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \mapsto 2^Q$ is the transition function;
4. $q_0 \in Q$ is the initial state;
5. $F \subseteq Q$ is the set of accepting states.



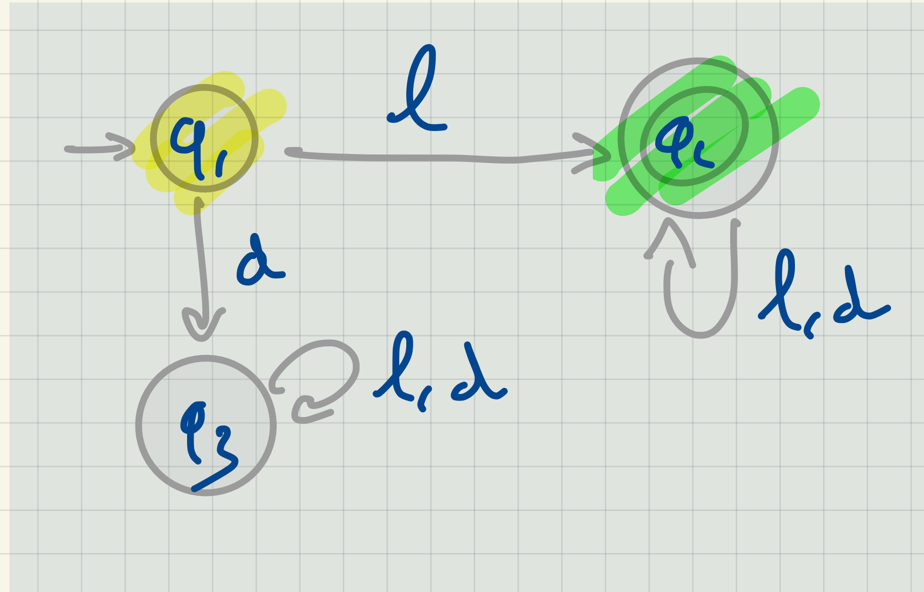
$Q = \{q_1, q_2, q_3\}$

$\Sigma = \{l, d\}$

initial: q_1

$F = \{q_2\}$

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \mapsto 2^{Q}$$

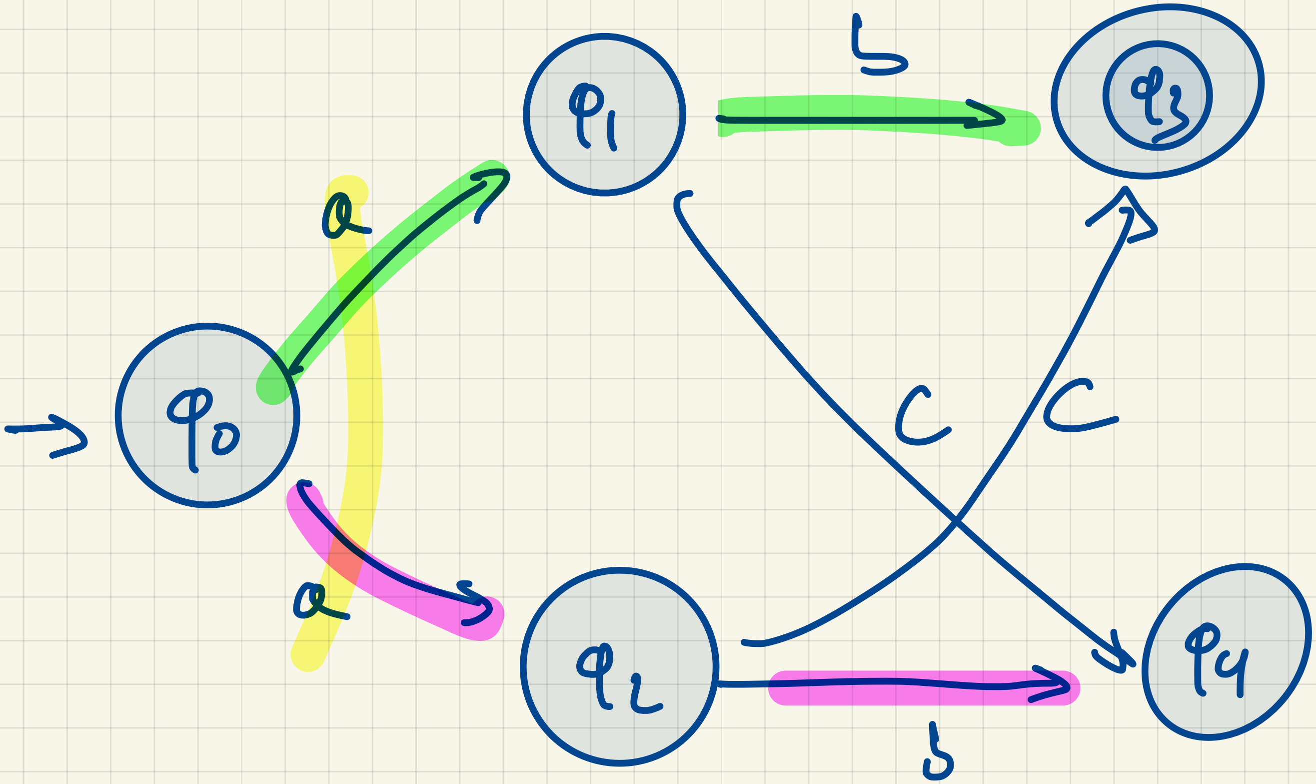


input state
letter or ε

output = set of states.

$$\delta(q_2, l) = \{q_2\}, \quad \delta(q_1, d) = \{q_3\}$$

$$2^{Q} = \text{all possible subsets of } Q$$



$$F = \{q_3\}$$

$$\delta(q_0, a) = \{q_1, q_2\}$$

Several classes of FA

non-deterministic

- ε- NFA
  = most general class
  non-det and ε-transitions are allowed.

- NFA = ε-transitions are not allowed
  $\delta(q, \varepsilon) = \emptyset$ for all state q.

deterministic

- DFA = no ε-transition, no non-det
  $|\delta(q, a)| = 1$ for all $q \in Q$, $a \in \Sigma$

3 classes of automata