3 types of finite state automatas:

1. Deterministic finite automata : this is the simplest and most efficient type of FA
2. Nondeterministic finite automata : FAs with choices activating independent computations
3. Nondeterministic finite automata with ϵ-transitions : nondeterministic automata with special moves that do not consume the input

Deterministic finite automata

recognition devices, only "yes" or "no" answer
they don't give the result, only if there is one
can't go back

These devices read input from left to right

They can only store a quantity of information limited by a constant, using the important notion of **state**

don't have memory

They are easy to implement in a computer (table) (basically)

Example

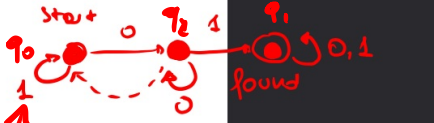
A DFA A that **accepts** the language of all strings of 0 and 1 containing the substring 01 :

$$L = \{x01y \mid x, y \in \{0,1\}^*\}$$

$$Q = \{q_0, q_1, q_2\},\ \Sigma = \{0,1\},\ F = \{q_1\}$$

δ function, specified by means of a **transition table** :

	0	1
$\rightarrow q_0$	q_2	q_0
$\star q_1$	q_1	q_1
q_2	q_2	q_1



Acceptance

(recognition device)

A DFA **accepts** a string $w = a_1 a_2 \cdots a_n$ if there is a path in the transition diagram that

- starts in the initial state
- ends in some final state
- has a sequence of transitions with labels $a_1 a_2 \cdots a_n$

Note that the above **is not** a mathematical definition

Extended transition function

(case of the DFA)

$\delta : Q \times \Sigma \to Q$

$\hat{\delta} : Q \times \Sigma^* \to Q$

$\hat{\delta} \neq \delta$

The δ transition function can be **extended** to function $\hat{\delta}$ defined over state and string pairs (as opposed to state and alphabet symbol pairs)

Base $\hat{\delta}(q, \epsilon) = q$ → don't change state

Induction $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$, $a \in \Sigma$, $x \in \Sigma^*$

} recursive definition

prefix — last symbol — current state — symbol to move to

Deterministic finite automata

Example

Given a string w over the alphabet Σ and a symbol a, let $\#_a(w)$ denote the number of **occurrences** of a in w

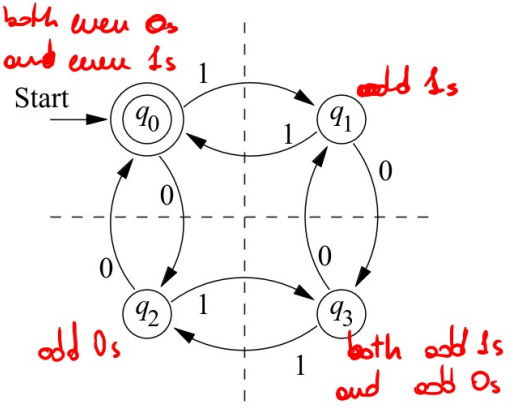
Specify a DFA A accepting all and only the strings in the following language

$$L = \{w \mid w \in \{0,1\}^*,\ \#_0(w) \text{ even},\ \#_1(w) \text{ even}\}$$

In words, L contains all and only the binary strings with an even number of 0's and an even number of 1's

What is the shortest string in L? Are there strings in L with odd length?

Example


Start
q_0
q_1
q_2
q_3
1
1
0
0
0
0
1
1
both even 0s and even 1s
odd 1s
odd 0s
both odd 1s and odd 0s

Deterministic finite automata

mathematical object

A **deterministic finite automaton** (DFA) is a 5-tuple

$$A = (Q, \Sigma, \delta, q_0, F)$$

where : five components :

1) Q is a **finite** set of **states**
2) Σ is the input symbol **alphabet**
3) δ is a **transition** function $Q \times \Sigma \to Q$ — deterministic, since every output is unique, not a set
4) $q_0 \in Q$ is the initial state (start)
5) $F \subseteq Q$ is a set of final states (finish)

Comment on the notion of determinism