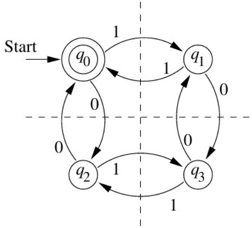
Example



start state

Is string $w = 0101$ accepted by A ?

- $\hat{\delta}(q_0, \epsilon) = q_0$ → base case
- $\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0) = \delta(q_0, 0) = q_2$
- $\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_2, 1) = q_3$
- $\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0) = \delta(q_3, 0) = q_1$
- $\hat{\delta}(q_0, 0101) = \delta(\hat{\delta}(q_0, 010), 1) = \delta(q_1, 1) = q_0 \in F$

Language recognized by a DFA

The language **recognized** by DFA A is

→ set of accepted strings

Describes the behaviour of an automaton ←

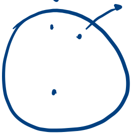
$$L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}$$

→ set of final states

The languages accepted by the class of DFAs are called **regular languages** $\{L(A) \mid A \text{ is a DFA}\}$

languages

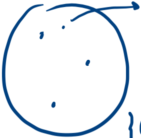
string



$\{w \mid \dots\}$

families / classes

language



$\{L \mid \dots\}$

Notational conventions

Commonly used notation for DFAs

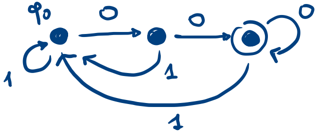
- $a, b, c, \ldots$ alphabet symbols
- u, v, w, x, y, z strings over input alphabet
- $p, q, r, s, q_0, q_1, q_2, \ldots$ states

Test

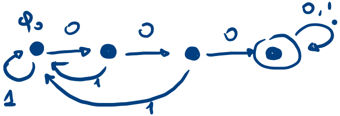
Specify DFAs for the following languages over the alphabet $\{0, 1\}$:

- set of all strings ending in 00
- set of all strings with three consecutive 0's
- set of all strings with 011 as a substring
- set of all strings that start or end (or both) with 01

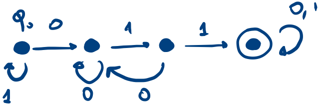
1) $\{x00 \mid x \in \{0,1\}^*\}$



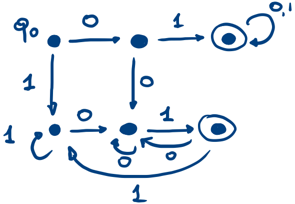
2) $\{x000y \mid x,y \in \{0,1\}^*\}$



3) $\{x011y \mid x,y \in \{0,1\}^*\}$



4) $\{xyz \mid x,y,z \in \{0,1\}^*,\ x = 01 \vee z = 01\}$



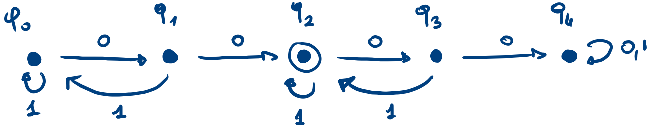
Exercise

Consider the language L of strings over the alphabet $\{0, 1\}$ with **exactly** one occurrence of string 00

Carry out the following points :

- draw the transition diagram of a DFA A such that $L(A) = L$
- state the meaning of each of A's states (i.e. for each state of A describe the strings leading to it)

Hint: define a "failure state" that can never reach any final state



q_0: I've not encountered the 00

q_1: I've possibly encountered 00, next step tells me if I did

q_2: I've encountered one 00, final state

q_3: I've possibly encountered another 00, next step tells me if I did

q_4: I've encountered at least two 00, I know already this string doesn't belong to this language

Nondeterministic finite automata

These automata accept only regular languages

Easier to design than DFAs

Later on we will see several examples of this fact

Very useful for implementing the **search** for a pattern in a text

Nondeterministic finite automata

find branching points somewhere

A nondeterministic finite automaton can **simultaneously** be in different states

not sure which of them he is in

The automaton **accepts** if at least one final state is reached at the end of the scan of the input string

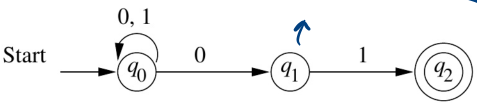
Equivalently, in a given state the automaton can **guess** which next state will lead to acceptance

This interpretation is not in the textbook

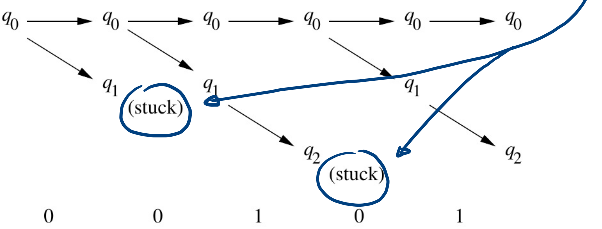
Example

Nondeterministic automaton N accepting all and only the strings ending in 01

no 0 arrow



Simultaneous computations of N on input string 00101



Nondeterministic finite automaton

A **nondeterministic finite automata** (NFA) is a 5-tuple

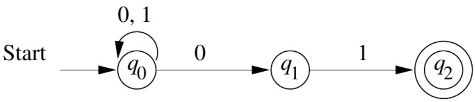
$$A = (Q, \Sigma, \delta, q_0, F)$$

where :

- Q is a **finite** set of **states**
- Σ is the **alphabet** of input symbols
- δ is a **transition** function $Q \times \Sigma \rightarrow 2^Q$, where 2^Q is the set of all subsets of Q (power set)
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of final states

Example

The transition diagram



represents the nondeterministic automaton

$$A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

with transition function δ

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star q_2$	$\emptyset$	$\emptyset$

Extended transition function $\hat{\delta}$

it's a set

Base $\hat{\delta}(q, \epsilon) = \{q\}$

it's a set

it's a set

Induction

$$\hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)$$

it's a state

it's a set

Notice the difference with the case of DFA in the induction part. Can you explain this?