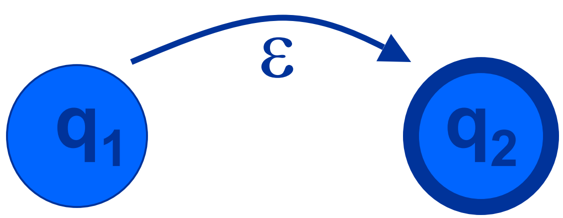
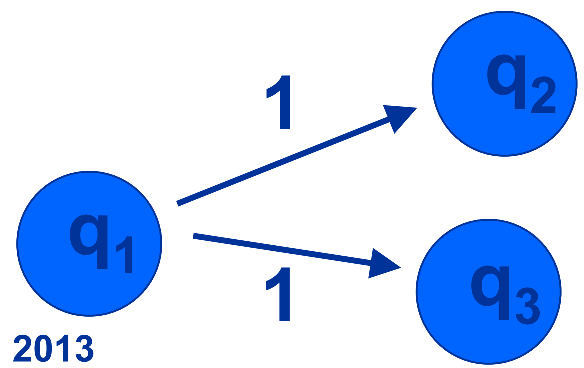
Nondeterminism

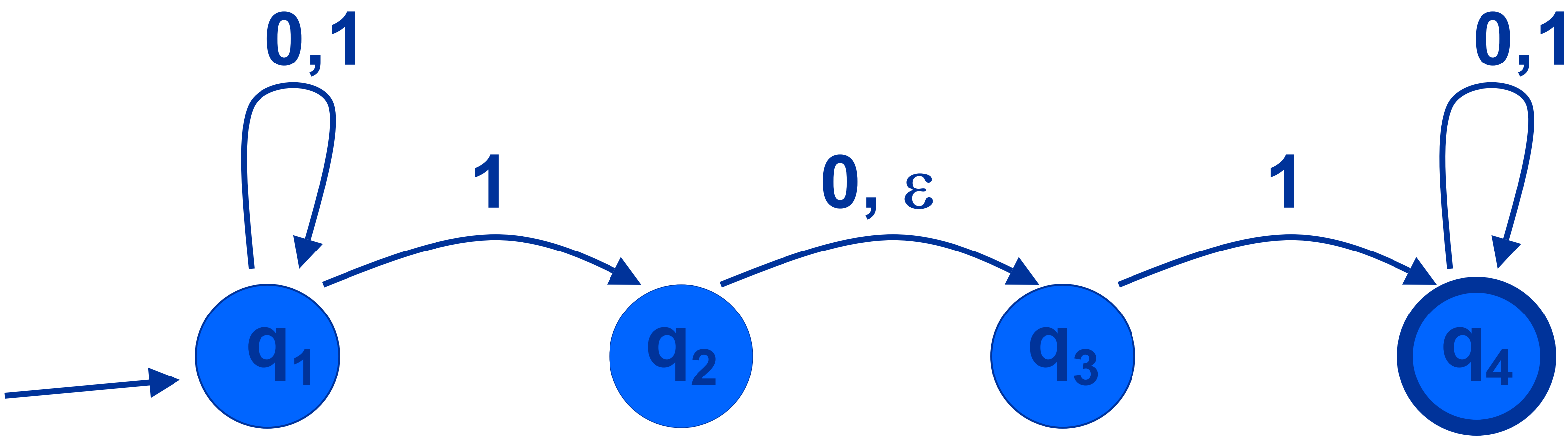
Nondeterministic machines are capable of being lucky, no matter how small the probability.

A nondeterministic finite automaton has transition rules/possibilities like





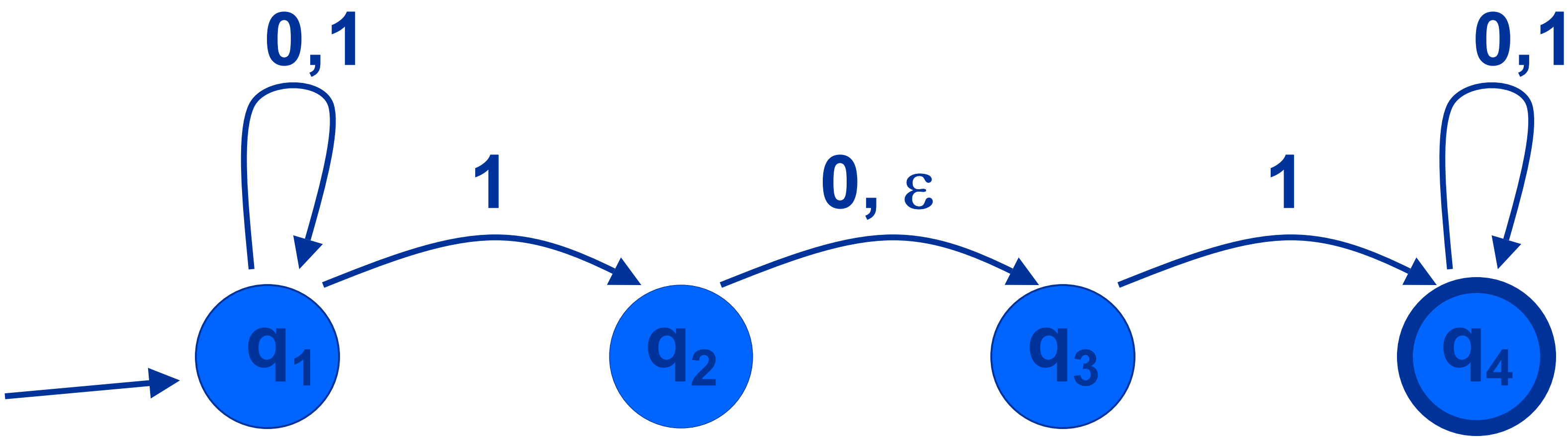
A Nondeterministic Automaton



This automaton accepts "0110", because there is a possible path that leads to an accepting state, namely:

$q_1 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_4$

A Nondeterministic Automaton



The string 1 gets rejected: on "1" the automaton can only reach: $\{q_1,q_2,q_3\}$.

Nondeterminism ~ Parallelism

For any (sub)string w, the nondeterministic automaton can be in a set of possible states.

If the final set contains an accepting state, then the automaton accepts the string.

"The automaton processes the input in a parallel fashion. Its computational path is no longer a line, but a tree." (Fig. 1.16)

Nondeterministic FA (def.)

- A nondeterministic finite automaton (NFA) M is defined by a 5-tuple $M=(Q,\Sigma,\delta,q_0,F)$, with
  - Q: finite set of states
  - Σ: finite alphabet
  - δ: transition function $\delta:Q\times\Sigma_\varepsilon\rightarrow\mathcal{P}(Q)$
  - $q_0\in Q$: start state
  - $F\subseteq Q$: set of accepting states

Nondeterministic $\delta: Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$

The function $\delta: Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ is the crucial difference. It means:
"When reading symbol "a" while in state q, one can go to one of the states in $\delta(q,a) \subseteq Q$."

The ε in $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ takes care of the empty string transitions.

Recognizing Languages (def)

A nondeterministic FA $\mathbf{M = (Q,\Sigma,\delta,q,F)}$ <u>accepts</u> a string $\mathbf{w} = w_1 \ldots w_n$ if and only if we can rewrite w as $y_1 \ldots y_m$ with $y_i \in \Sigma_\varepsilon$ and there is a sequence $r_0 \ldots r_m$ of states in Q such that:

1) $r_0 = q_0$

2) $r_{i+1} \in \delta(r_i, y_{i+1})$ for all $i=0,\ldots,m-1$

3) $r_m \in F$

Exercises

[Sipser 1.5]: Give NFAs with the specified number of states that recognize the following languages over the alphabet $\Sigma=\{0,1\}$:

1. $\{ w \mid w$ ends with 00$\}$, three states
2. $\{0\}$; two states
3. $\{ w \mid w$ contains even number of 0s, or exactly two 1s$\}$, six states
4. $\{0^n \mid n \in N \}$, one state

Exercises - 2

Proof the following result:
"If L_1 and L_2 are regular languages, then $L_1 \cap \overline{L_2}$ is a regular language too."

Describe the language that is recognized by this nondeterministic automaton: