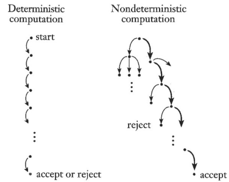
# Nondeterministic finite automata

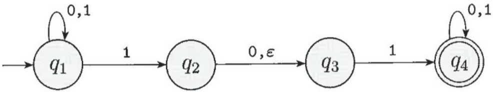
**Definition 1.7** (Nondeterministic finite automaton) A **nondeterministic finite automaton** (NFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. $Q$ is a finite set of states,
2. $\Sigma$ is a finite alphabet,
3. $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ is the transition function, with $\Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.



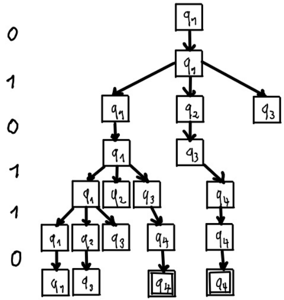

1 1 1

ε 1 ε 1 1 ε




| $\delta$ | 0 | 1 | $\varepsilon$ |
|---|---|---|---|
| $q_1$ | $\{q_1\}$ | $\{q_1, q_2\}$ | $\emptyset$ |
| $q_2$ | $\{q_3\}$ | $\emptyset$ | $\{q_3\}$ |
| $q_3$ | $\emptyset$ | $\{q_4\}$ | $\emptyset$ |
| $q_4$ | $\{q_4\}$ | $\{q_4\}$ | $\emptyset$ |

$N_1 = (Q, \Sigma, \delta, q_1, F)$, $\quad Q = \{q_1, q_2, q_3, q_4\}$, $\quad \Sigma = \{0, 1\}$, $\quad F = \{q_4\}$




**Definition 1.8** (Strings accepted by NFA N)
Let $N = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic finite automaton and $w$ be a string over alphabet $\Sigma$.

$N$ **accepts** $w$ if we can write $w$ as $w = y_1 y_2 \dots y_m$, $y_i \in \Sigma_\varepsilon$ and if there exists a sequence of states $r_0, r_1, \dots, r_m$ (in $Q$), such that all following three conditions hold:

1. $r_0 = q_0$ *(N starts in start state.)*
2. $r_{i+1} \in \delta(r_i, y_{i+1})$, for $i = 0, \dots m-1$ *(State change follows transition function.)*
3. $r_m \in F$ *(N ends up in accept state)*

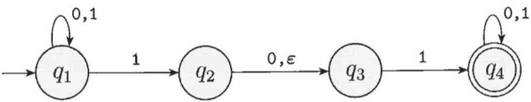
If $N$ does not accept $w$, it **rejects** it.

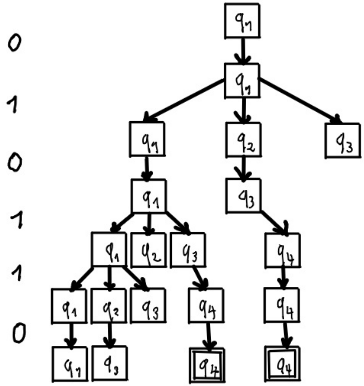
**Definition 1.9** (Computation branch of an NFA) Let $N = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic finite automaton and $w = w_1 w_2 \cdots w_n$ be a string over alphabet $\Sigma$.

A **computation branch of $N$ on $w$** allows for a modification of the input string $w$ to a string $w = y_1 y_2 \cdots y_m$, $y_i \in \Sigma_\varepsilon$ and is a sequence of states $c = r_0, r_1, \ldots, r_m$, such that the following two conditions hold:

1. $r_0 = q_0$ *(N starts in start state.)*
2. $r_{i+1} \in \delta(r_i, w_{i+1})$, for $i = 0, \ldots m-1$ *(State change follows transition function.)*

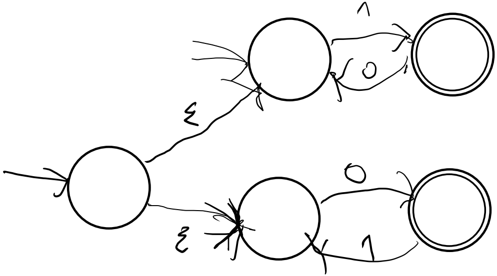
We call $c$ an **accepting computation branch**, if $r_n \in F$, otherwise we call it a **rejecting computation branch**.



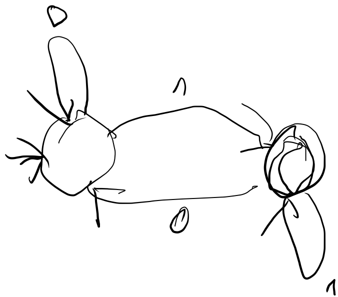




**Definition 1.10** (Language of NFA N) Let $N = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic finite automaton.

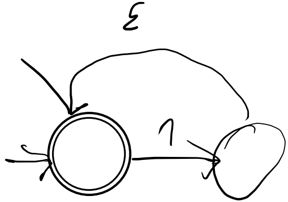
The **language of N** $L(N)$ is the set of all strings that are accepted by $N$. We say: $N$ **recognizes** $L(N)$.



ε
ε
1
0
0
1

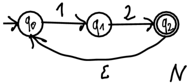
0
1
0
1

ε
1

# 1 Reading and understanding NFAs

You are given the following state transition diagram:

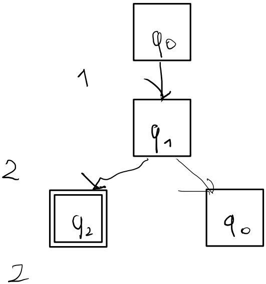



1. Give the formal description of the corresponding *NFA* as 5-tuple.
2. Let the NFA “run” on the input strings 12, 122, 1212.
3. Find the language recognized by this automaton.

1) $N = (Q, \Sigma, \delta, q_0, F)$ $\quad Q = \{q_0, q_1, q_2\}$ $\quad F = \{q_2\}$

$\Sigma = \{1, 2\}$

| | 1 | 2 | ε |
|---|---|---|---|
| $q_0$ | $\{q_1\}$ | $\emptyset$ | $\emptyset$ |
| $q_1$ | $\emptyset$ | $\{q_2\}$ | $\emptyset$ |
| $q_2$ | $\emptyset$ | $\emptyset$ | $\{q_0\}$ |




2) 12 $\Rightarrow$ accept

122 $\Rightarrow$ reject 2

1212 $\Rightarrow$ accept

3) $L = \{ w \in \Sigma^* \mid w \text{ repeats "12" } n \text{ times}, n \geq 1\}$

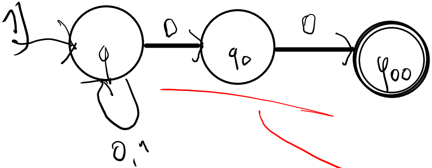
## 2 Constructing an FA

You are given the following regular languages over $\Sigma = \{0, 1\}$.

1. $L_1 = \{w | w \text{ ends with } 00\}$
2. $L_2 = \{w | w \text{ is given by an arbitrary number of 0s}\}$ $\Sigma = \{0\}$
3. $L_3 = \{w | w \text{ has length at least 3 and its third symbol is a } 0\}$

Find for each of the languages the nondeterministic finite automaton the recognizes $L$ it and describe it by an STD.

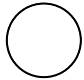
1)

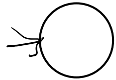



000|00

1001|00

111|00

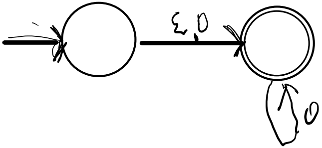


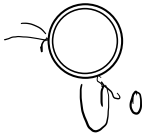


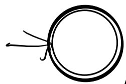
$N_4$

$L(N_4) = \emptyset$

2)



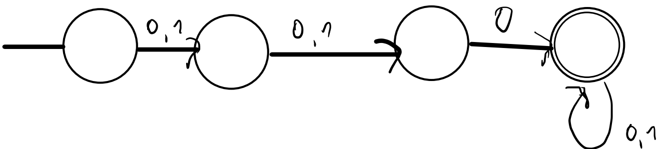



$N_5$

$L(N_5) = \{\varepsilon\}$

3)

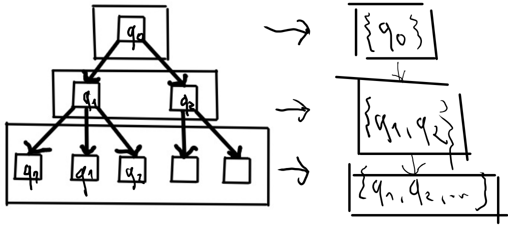


**Theorem 1.3** Every nondeterministic finite automaton has an equivalent deterministic finite automaton, i.e. there exists for every NFA $N$ a DFA $M$ such that $L(M) = L(N)$.

**Central idea:**
M "simulates" the computation of N by simultaneously going through all computation branches of N.

N NFA states $Q$

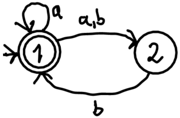
M DFA $Q' = \mathcal{P}(Q)$




**Corollary 1.1** A language is regular if and only if some nondeterministic finite automaton recognizes it.

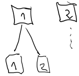
### 3 Example for proof of Theorem 1.3

Let the NFA $N$ be given by its STD:



$N = (Q, \Sigma, \delta, 1, \{1\})$

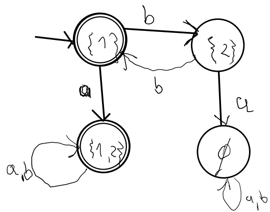
$Q = \{1, 2\} \qquad \Sigma = \{a, b\}$

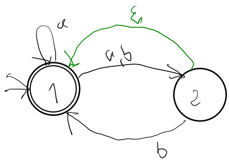


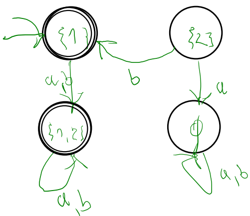
Use the construction from the proof of Theorem 1.3 to find (as an STD) the FA $M$ such that $L(M) = L(N)$.

$M = (Q', \Sigma, \delta', \ , \ )$

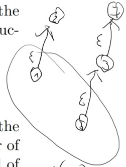
$Q' = \mathcal{P}(Q) = \{ \emptyset, \{1\}, \{2\}, \{1,2\} \}$







er the
struc-



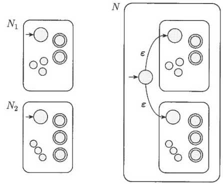
ll the
ber of
od of

$E(R)$

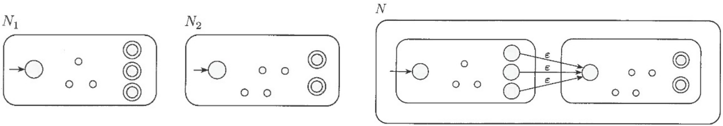
$\|$

$\{1, 5, 3, 2, 7$

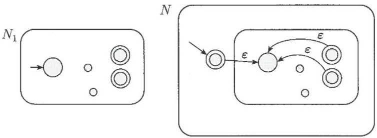
**Theorem** The class of regular languages is closed under the union operation. In other words, if $A_1$ and $A_2$ are regular languages, so is $A_1 \cup A_2$.



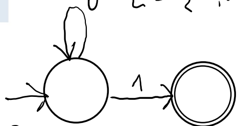

**Theorem** The class of regular languages is closed under the concatenation operation. In other words, if $A_1$ and $A_2$ are regular languages, then $A_1 \circ A_2$ is regular.

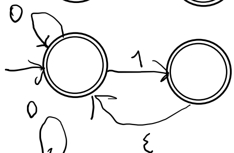



**Theorem 1.4** The class of regular languages is closed under the star operation.




$L = \{1, 01, 001, 0001, \ldots\}$

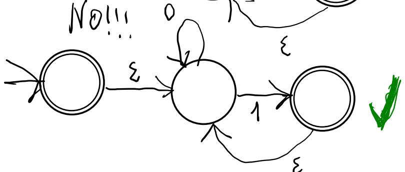



$L = \{0, 00, \ldots$

$L^*$ ? →

No!!!




$\varepsilon \in L^*$

100

1,00