Determinism: computation always continues in a uniquely determined way.

Nondeterminism: have more (or none) choices

Example:

\[ \{ w \in \{0,1\}^* \mid w \text{ contains } 001 \text{ or } 0101 \text{ as a substring} \} \]
Determinism: computation always continues in a uniquely determined way.

Nondeterminism: have more (or none) choices

Example:

\[ \{ w \in \{0,1\}^* \mid w \text{ contains 001 or 0101 as a substring} \} \]

Nondeterministic FA can also use $\varepsilon$-transitions:
Example:

\[ L = \{ w \in \{0,1\}^* \mid w \text{ contains 1 in the third position from the end} \} \]

Does there exist a (deterministic) FA recognizing this language?
Example:
\[ \{ w \in \{0\}^* \mid |w| \text{ is divisible by 2 or 3} \} \]

with an NFA with \( \leq 8 \) states.
Formal definition:

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

- $Q$ is a finite set of states,
- $\Sigma$ is a (finite) alphabet,
- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ is the transition function,
- $q_0 \in Q$ is the start state,
- $F \subseteq Q$ is the set of accept states.

For DFAs: $\delta: Q \times \Sigma \rightarrow Q$
Let $N=\langle Q, \Sigma, \delta, q_0, F \rangle$ be an NFA and let $w=w_1w_2\ldots w_n$ where each $w_i \in \Sigma \cup \{\epsilon\}$. Then $N$ accepts $w$ if there exists a sequence of states $r_0, r_1, \ldots, r_n \in Q$ such that:

1) $r_0 = q_0$

2) $r_{i+1} \in \delta(r_i, w_{i+1}) \forall i \in \{1, \ldots, n\}$

3) $r_n \in F$.

For DFA:

2) $\delta(r_i, w_{i+1}) = r_{i+1} \forall i \in \{0, \ldots, n-1\}$
Thm 1.39: Every NFA has an equivalent DFA.

Proof idea:
- for starters, no ε-transitions in the NFA
- example:
Thm 1.39: Every NFA has an equivalent DFA.

Proof idea, part 2 (getting rid of ε-transitions in the NFA):
- for $R \subseteq Q$ let:
  $$E(R) = \{ q \in Q \mid q \text{ can be reached from } R \text{ by traveling along 0 or more } \varepsilon \text{-arrows} \}$$
Thm 1.45 (revisited): The class of regular languages is closed under the union operation.

want to show: the union of any two regular languages is also a regular language

i.e. given two (D)FA $N_1, N_2$, we can construct a DFA $M$ s.t. $L(M) = L(N_1) \cup L(N_2)$

i.e. (using Thm 1.39):

given two NFAs $N_1, N_2$, we can construct an NFA $N$ s.t. $L(N) = L(N_1) \cup L(N_2)$
Thm 1.47: The class of regular languages is closed under the concatenation operation.

Given two NFA $N_1, N_2$, we want to construct a NFA $N$ s.t. $L(N) = L(N_1) \cdot L(N_2)$.
Closure under regular operations

Thm 1.49: The class of regular languages is closed under the star operation.

\[ N_i = (Q_i, \Sigma, \delta_i, q_{i1}, F_i) \quad \text{we want to construct an NFA } N = (Q, \Sigma, \delta, q_0, F) \]

\[ \text{s.t. } L(N) = L(N_i)^* \]

Given an NFA \( N_i \), we want to construct an NFA \( N \) such that \( L(N) = L(N_i)^* \).

Let:

\[ Q = Q_i \cup \{ q_{\text{new}} \} \]

assuming \( q_{\text{new}} \notin Q_i \)

\[ q_0 = q_{\text{new}} \]

\[ F = F_i \cup \{ q_{\text{new}} \} \]

\[ \delta(q, \sigma) = \begin{cases} 
\delta_i(q, \sigma) & \text{if } q \in Q_i, F_i \text{ and } \sigma \in \Sigma_e \\
\delta_i(q, \sigma) & \text{if } q \in F_i \text{ and } \sigma \in \Sigma_e \\
\{ q_{\text{new}} \} & \text{if } q \in F_i \text{ and } \sigma = \varepsilon \\
\cup \delta_i(q, \sigma) & \text{if } q = q_{\text{new}} \text{ and } \sigma = \varepsilon \\
\{ q_i \} & \text{if } q = q_{\text{new}}, \sigma \in \Sigma \\
\emptyset & \text{if } q = q_{\text{new}}, \sigma \in \Sigma \end{cases} \]