Determinism: computation always continues in a uniquely determined way.

Nondeterminism: have more (or none) choices

Example:

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

Nondeterminism: have more (or none) choices

Example:

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

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

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

Does there exist a (deterministic) FA recognizing this language?

\[ \text{YES but... we need a state for every combination of the last three digits} \]
Nondeterminism

Example:
\[ \{ w \in \{0\}^* \mid |w| \text{ is divisible by 2 or 3} \} \]

NFA combination of 2 automata
the new automaton accepts \( L_1 \cup L_2 \)

result of the cross-product construction
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 
- \(\delta: Q \times \Sigma \cup \{\varepsilon\} \rightarrow P(Q)\) is the transition function 
- \(q_0 \in Q\) is the start state 
- \(F \subseteq Q\) is the set of accept state
Let $N=(Q, \Sigma, \delta, q_0, F)$ be an NFA and let $w=w_1w_2\ldots w_n$ where each $w_i \in \Sigma$. Then $N$ accepts $w$ if

1) $p_0 = q_0$

2) $\delta(p_i, w_{i+1}) \subseteq p_{i+1}$

3) $p_n \in F$

For DFAs

1) $p_0 = q_0$

2) $\delta(p_i, w_{i+1}) = p_{i+1}$

3) $p_n \in F$

Note: the other sections define acceptance through extended transition functions.
Thm 1.39: Every NFA has an equivalent DFA.

Proof idea:
- for starters, no $\varepsilon$-transitions in the NFA
- example: $\Sigma = \{a, b\}$

### Example Diagram

- Possible computation paths:
  - $1 \xrightarrow{a} 2 \xrightarrow{b} 3$.
  - $1 \xrightarrow{a} 2 \xrightarrow{b} 1$.

- No need to investigate these, the paths from now on will be the same as above (**).
Thm 1.39: Every NFA has an equivalent DFA.

Proof idea:
- for starters, no $\epsilon$-transitions in the NFA
- example:

NFA:

\[ M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1) \]

\[ Q_1 = \emptyset(q_1) \]
\[ q_{01} = \{q_{01}\} \]
\[ F = \{A \in Q_1 \mid A \cap F_1 \neq \emptyset\} \]

DFA:

\[ M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2) \]
\[ \delta_2(A, \sigma) = \bigcup_{p \in A} \delta_1(p, \sigma), \quad \forall \sigma \in \Sigma, \forall A \in Q_1 \]

Construction done.

\[ \square \]
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.

\[ L(M) = L(M_1) \cup L(M_2) \]
Thm 1.47: The class of regular languages is closed under the concatenation operation.
Thm 1.49: The class of regular languages is closed under the star operation.

This construction does not work, e.g.

\[ L(M) = L(M_i)^* \]

\( M \) s.t.

Contains \( \epsilon \)

Note: we might make the old accepting states non-accepting.