Nondeterminism

Determinism: computation always continues in a uniquely determined way.

Nondeterminism: have more (or none) choices

Example:

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

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

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

An example of a NFA for \( L_{001} \):

- Merge the two starting states (yellow and green).
- Example input: 1101000111001110 is accepted (two possible accepting computation paths)
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:
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?
Example:
\[ \{ w \in \{0\}^* \mid |w| \text{ is divisible by 2 or 3} \} \]
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
- \(\delta: Q \times \Sigma \cup \{\epsilon\} \rightarrow \mathcal{P}(Q)\) is the transition function
- \(q_0 \in Q\) is the start state
- \(F \subseteq Q\) is the set of accept state

**DFA:**

\[\delta: Q \times \Sigma \rightarrow Q\]
Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA and let $w = w_1 w_2 \ldots w_n$ where each $w_i \in \Sigma^*$. Then $N$ accepts $w$ iff there exists a sequence of states $r_0, r_1, r_2, \ldots, r_n$ where $\forall i \in \{0, \ldots, n\}: r_i \in Q$ such that:

- $r_0 = q_0$
- $r_{k+1} \in \delta(r_k, \sigma_{k+1})$ \quad $\forall k \in \{0, 1, \ldots, n-1\}$
- $r_n \in F$

Rewriting the top of the slide to account for $\varepsilon$-transitions:

$N$ accepts $w$ if we can split $w = \sigma_1 \sigma_2 \ldots \sigma_n$ where each $\sigma_i \in \Sigma^*$ and there exists a sequence of states $r_0, r_1, r_2, \ldots, r_n$ s.t.:
Thm 1.39: Every NFA has an equivalent DFA.

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

```
Equivalence of NFAs and DFAs

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. We want to construct a DFA $M = (Q', \Sigma, \delta', q'_0, F')$ as follows:

- $Q' = \mathcal{P}(Q)$
- $q'_0 = \{q_0\}$
- $F' = \{S \in Q' : S \cap F \neq \emptyset\}$

Let $\delta'(S, \sigma) = \bigcup_{q \in S} \delta(q, \sigma)$, where $S \in Q', \sigma \in \Sigma$.
```
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:
  \[
  E(R) = \{ q \in Q \mid q \text{ can be reached from } R \text{ by traveling along 0 or more } \varepsilon \text{-arrows} \}
  \]

Example:

\[
E(\{q_0,q_3\}) = \{q_0,q_1\}
\]
\[
E(\{q_0,q_2\}) = \{q_0,q_2,q_1,q_3\}
\]
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:

\[
E(R) = \{ q \in Q \mid q \text{ can be reached from } R \text{ by traveling along 0 or more } \varepsilon\text{-arrows} \}
\]

Example:

Have an NFA \( N = (Q, \Sigma, \delta, q_0, F) \)

Want to construct a DFA \( M = (Q', \Sigma, \delta', q'_0, F') \)

Let:

\[
Q' = \rho(Q)
\]

\[
q'_0 = \bigcup_{q \in F} E(\{q\})
\]

\[
F' = \{ S \subseteq Q' \mid S \cap F \neq \emptyset \}
\]

\[
\delta'(s, \sigma) = \bigcup_{q \in S} \delta(q, \sigma)
\]

where \( S \subseteq Q', \sigma \in \Sigma \)
Thm 1.45 (revisited): The class of regular languages is closed under the union operation.

Suppose we have two regular lang. \( L_1, L_2 \), we want to show that \( L_1 \cup L_2 \) is regular.

Let \( N_1, N_2 \) be NFAs for \( L_1, L_2 \).

\[ N_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \]
\[ N_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2) \]

Assume \( Q_1 \cap Q_2 = \emptyset \)

Let \( N = (Q, \Sigma, \delta, q_0, F) \) where \( L(N) = L(N_1) \cup L(N_2) \)

\[ Q = Q_1 \cup Q_2 \cup \{ \emptyset \} \]
\[ \Sigma = \Sigma_1 \cup \Sigma_2 \]
\[ q_0 = \emptyset \]
\[ F = F_1 \cup F_2 \]

\[ \delta(q, \sigma) = \begin{cases} 
\delta_1(q, \sigma) & \text{if } q \in Q_1, \sigma \in \Sigma_1, \\
\delta_2(q, \sigma) & \text{if } q \in Q_2, \sigma \in \Sigma_2, \\
[q, q_1] & \text{if } q = q_0, \sigma = \epsilon \\
\emptyset & \text{otherwise}
\end{cases} \]
Thm 1.47: The class of regular languages is closed under the concatenation operation.

do the formal description on your own 😊
**Thm 1.49:** The class of regular languages is closed under the star operation.

**Formally:**
- **Given:** NFA $N = (Q, \Sigma, \delta, q_0, F)$
- **Want:** NFA $N' = (Q', \Sigma', \delta', q_0', F')$ such that $L(N') = L(N)^*$

Let $Q' = Q \cup \{q'_0\}$ and $F' = \{q'_0\} \cup F$.