II.3.1. String Recognition (13.1)

II.3.2. Nondeterministic Finite State Automata (13.2)

II.3.3. Examples of Automata (13.3)

II.3.4. Automata with Empty Move Transitions (13.4)

II.3.5. Deterministic Finite State Automata (13.6)

Example

Automaton with empty move transitions are like NFAs, but have the possibility to make transitions without consuming a letter. These transitions are labelled as $\epsilon$.
II.3.4. Automata with Empty Move Transitions (13.4)

### Definition NFA with Empty Moves

A **non-deterministic finite state automaton with empty moves** \((Q, q_0, F, T, \rightarrow)\) is given by:

- A finite set \(Q\) of **states**.
- A single **initial state** \(q_0\).
- A set \(F \subseteq Q\) of **accepting states**.
- A finite set of **terminal symbols** \(T\).
- A relation \(\rightarrow\) between \(q \in Q\), \(a \in T \cup \{\epsilon\}\) and \(q' \in Q\).

We write \(q \rightarrow a \rightarrow q'\) iff \(q, a, q'\) are in this relation.

### The Extended Transition Function

We extend the function \(\rightarrow^*\) to NFA with empty moves as follows:

**Definition**

Let \(A = (Q, q_0, F, T, \rightarrow)\) be an NFA with empty moves. Then we define a relation \(\rightarrow^*\) between \(q \in Q\), \(w \in T^*\), \(q' \in Q\).

We write \(q \rightarrow^* w \rightarrow q'\) iff \(q, w, q'\) are in this relation.

- \(q \rightarrow^* q'\) iff

  \[
  q = q_0 \rightarrow_0 \rightarrow q_1 \rightarrow_1 \epsilon \rightarrow q_2 \rightarrow_2 \epsilon \rightarrow \cdots \epsilon \rightarrow q_n = q'
  \]

  for some \(n \in \mathbb{N}\), \(q_i \in Q\).

  The special case \(n = 0\) is allowed, so we have always \(q \rightarrow^* q\).
Informal Understanding

So $q^{a_1 \cdots a_n} \xrightarrow{\epsilon} q'$ if we can reach from $q$ state $q'$ using finitely many $\epsilon$-transitions, then one $a_1$-transition, then again finitely many $\epsilon$-transitions, an $a_2$ transition, ... , finitely many $\epsilon$-transitions, and an $a_n$ transition:

$q_0 \xrightarrow{\epsilon} \cdots \xrightarrow{\epsilon} q_1 \xrightarrow{a_1} \cdots \xrightarrow{\epsilon} q_2 \xrightarrow{\epsilon} \cdots \xrightarrow{\epsilon} q_{n-1} \xrightarrow{\epsilon} \cdots \xrightarrow{\epsilon} q_n \xrightarrow{\epsilon} \cdots \xrightarrow{\epsilon} q'$

Language Accepted

Definition

Let $A = (Q, q_0, F, T, \rightarrow)$ be an NFA with empty moves. Then $L(A) = \{w \in T^* | \exists q' \in F. q_0 \xrightarrow{w \epsilon}* q'\}$.

NFA with Empty Moves are Equivalent to NFAs

We show that for every NFA $A$ with empty moves we can find an NFA $A'$ without empty moves s.t. $L(A') = L(A)$.

This is done as follows:

Let $A = (Q, q_0, F, T, \rightarrow)$.

We define $A' = (Q, q_0, F', T, \rightarrow')$ where

- $q \xrightarrow{\alpha} q'$ iff $q \xrightarrow{\alpha \epsilon} q'$,
- $F' := \{q \in Q | \exists q' \in F. q \xrightarrow{\epsilon} q'\}$.

NFA with Empty Moves are Equivalent to NFAs

So the transitions of $A'$ are obtained by allowing first finitely many empty moves and then one proper transition and again finitely many empty moves.

At the end we might need to make finitely many empty moves before reaching the accepting state, therefore the set of accepting states is the set of states from which we can reach an accepting state of $A$ using empty moves.
Correctness of the Translation

- One can now easily see that for $w \neq \epsilon$ we have
  \[ q \xrightarrow{w^*} q' \text{ iff } q \xrightarrow{w^*} q' \]
- Note that $q \xrightarrow{\epsilon^*} q' \text{ iff } q' = q$.

We get

\[ L(A') = \{ w \in T^* | \exists q' \in F'.q \xrightarrow{w^*} q' \} \]
\[ = \{ w \in T^* | \exists q' \in Q.\exists q'' \in F. ( (w = \epsilon \land q' = q) \lor (w \neq \epsilon \land q \xrightarrow{w^*} q') \land q' \xrightarrow{\epsilon^*} q'' ) \} \]
\[ = \{ w \in T^* | \exists q' \in F.q \xrightarrow{w^*} q' \} = L(A) \]

Example

Consider the NFA with empty moves from above:

\[ \begin{align*}
q_0 & \xrightarrow{\epsilon} \xrightarrow{a} q_1 \\
q_1 & \xrightarrow{b} \xrightarrow{a,b} q_1
\end{align*} \]

The tranformed automaton is as follows:

\[ \begin{align*}
q_0 & \xrightarrow{a} \xrightarrow{a,b} q_0 \\
q_1 & \xrightarrow{b} \xrightarrow{a,b} q_1
\end{align*} \]
Theorem II.3.4.1.

**Theorem (II.3.4.1.)**

For any NFA $A$ with empty moves there exist an NFA without empty moves s.t. $L(A') = L(A)$. $A'$ can be computed from $A$.

**Proof Idea of Theorem II.3.5.1.**

- We will define a new automaton $A' = (Q', q_0', F', T', \rightarrow')$.
- $Q'$ is the set of all subsets of $Q$, i.e. $P(Q)$.
  - Note that since $Q'$ is finite $P(Q)$ will be finite too.
  - Having reached state $\{q_1, \ldots, q_k\}$ means that $\{q_1, \ldots, q_k\}$ are the set of states we could have reached in $A$ by making different choices, but following the same word.
- $q_0' := \{q_0\}$.
  - Initially the states we have reached are the elements of $\{q_0\}$.

- Let $B \subseteq Q$.
  - $B \rightarrow' C$

  where
  
  $C = \{q \in Q \mid \exists q' \in B. q' \xrightarrow{a} q\}$

  - If we could have reached any of the states in $B$, then after reading $a$ in addition, we could have reached any of the states we can reach from a state in $B$ by an $a$-transition.
II.3.5. Deterministic Finite State Automata (13.6)

Proof Idea for Theorem II.3.5.1.

- The accepting states are the set of states containing at least one accepting state.
  - If having read word \( w \) we can reach the states \( \{q_1, \ldots, q_k\} \), then the word \( w \) can be accepted, if one of \( q_1, \ldots, q_k \) is an accepting state.

Resulting DFA

\[
\begin{align*}
\text{automaton} & \quad A' \\
\text{states} & \quad \mathcal{P}(Q) \\
\text{terminals} & \quad T \\
\text{start} & \quad \{q_0\} \\
\text{final} & \quad B \in \mathcal{P}(Q) \text{ s.t. } B \cap F \neq \emptyset \\
\text{transitions} & \quad B \xrightarrow{a} C \\
& \text{where } C = \{q' | \exists q \in B.q \xrightarrow{a} q'\}.
\end{align*}
\]

Simplification

- Usually only some states of \( A' \) are reachable.
- We can omit all unreachable states and get an equivalent automaton.
- We can construct the reachable states of \( A' \) by starting with \( \{q_0\} \), and constructing from there systematically all transitions and the states reached.
- Furthermore, there will be a state \( \emptyset \).
  - When we have reached that state we have consumed a word for which there is no complete run of \( A \).
  - \( \emptyset \notin F' \), \( \emptyset \xrightarrow{a} \emptyset \).
  - So \( \emptyset \) is a sink, a state from which we cannot escape, and which doesn’t accept anything.
  - If we omit \( \emptyset \), we obtain a DFA with the same language.

Proof of Theorem II.3.5.1.

- Consider the DFA as given above.
- Define for \( B \subseteq Q, w \in T^*, \delta(B, w) \subseteq Q \) by
  \[
  \delta(B, w) := \{q \in Q | \exists q' \in B.q \xrightarrow{w} q\}
  \]
- So we have
  \[
  B \xrightarrow{a} \delta(B, a)
  \]
Proof of Theorem II.3.5.1.

We show for \( B \subseteq Q \) that
\[
B \xrightarrow{w^*} \delta(B, w)
\]
by induction on the length of \( w \):

- Case \( w = \epsilon \):
  
  \[
  \delta(B, \epsilon) = \{ q \in Q \mid \exists q' \in B. q' \xrightarrow{\epsilon} q \} = \{ q \in Q \mid \exists q' \in B. q' = q \} = B
  \]
  
  and
  
  \[
  B \xrightarrow{\epsilon^*} B = \delta(B, \epsilon)
  \]

- Case \( w = w'a \):
  
  By IH
  
  \[
  B \xrightarrow{w'a} \delta(B, w')
  \]
  
  Furthermore
  
  \[
  \delta(B, w') \xrightarrow{a'} \{ q' \mid \exists q \in \delta(B, w'). q \xrightarrow{a} q' \}
  = \{ q' \mid \exists q'' \in B. \exists q \in Q. q'' \xrightarrow{w'a} q \wedge q \xrightarrow{a} q' \}
  = \{ q' \mid \exists q'' \in B. q'' \xrightarrow{w'a} q' \}
  = \delta(B, w'a)
  \]
  
  Therefore
  
  \[
  B \xrightarrow{w'a} \delta(B, w'a)
  \]

We obtain now

\[
L(A') = \{ w \in T^* \mid \exists B \in F'. \{ q_0 \} \xrightarrow{w^*} B \}
= \{ w \in T^* \mid \delta(\{ q_0 \}, w) \cap F \neq \emptyset \}
= \{ w \in T^* \mid \exists q \in F. q \in \delta(\{ q_0 \}, w) \}
= \{ w \in T^* \mid \exists q \in F. q_0 \xrightarrow{w^*} q \}
= L(A)
\]