CSE2315 Slides week 2 - NFAs

Matthijs Spaan Stefan Hugtenburg

Algorithmics group

Delft University of Technology

19 February 2020

This lecture

- Introduction to nondeterministic finite automata
- Formal definition NFAs
- Acceptation and recognition by NFAs
- Equivalence of automata
- Construction of DFAs from NFAs without ε-transitions
- Construction of DFAs from NFAs with ε-transitions
- So languages recognized by NFAs are regular!
- Regular operations and closure properties of regular languages

Nondeterministic finite automata (NFAs)

(see also videos Nondeterministic Finite State Machines: Introduction and Nondeterministic Finite State Machines: Formal Definition)

Three differences with deterministic finite automata:

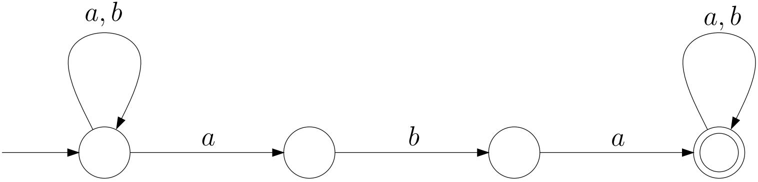
1. Multiple transitions possible for one symbol from a state;
2. Possibility that no transition is defined for a symbol from a state;
3. ε transitions: reading nothing but transitioning to another state.

Result: In an NFA, it is possible for there to be zero, one or more ways to consume a string w *in its entirety*.

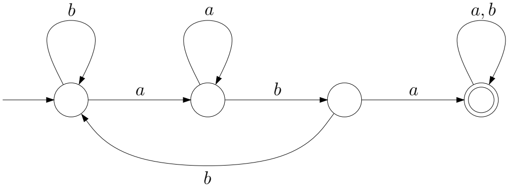
Adaptation of notion of acceptation (*informally*): A string w is accepted by an NFA iff at least one path label by w exists that ends in an accepting state.

Example

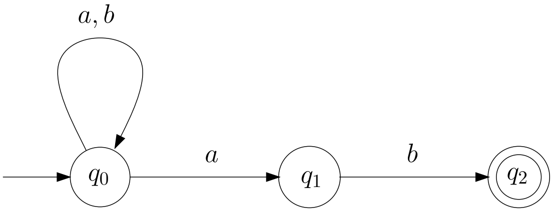
An NFA that accepts every word that contains the substring aba:



Note that this NFA is simple than the DFA that recognizes the same language:



Example



An NFA that accepts every word with the suffix ab.

Nondeterminism and NFAs

- Adds the notion of choice.
- Can be seen as a form of (unlimited) parallelism.
- NFAs form a good starting point for studying the phenomenon of nondeterminism
- Higher order, declarative description of processes that can be simulated by deterministic processes.
- Some theorems are easier to prove for NFAs than for DFAs.
- Gives natural descriptions of certain languages, for instance of the form $L_1 \cup L_2$. See also previous slides.
- *DFAs have nicer formal properties, while NFAs are easier to construct.*

Nondeterministic Finite Automata (NFAs)

Definition 1.37 A nondeterministic finite automaton (*NFA*) is an ordered quintuple $N = (Q, \Sigma, \delta, q_0, F)$, with:

- Q, Σ, q_0 and F are defined as in a DFA;
- however: $\boxed{\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathscr{P}(Q).}$

Here $\mathscr{P}(Q)$ is called the power set of Q.

Notation: Instead of $\Sigma \cup \{\varepsilon\}$, we usually write Σ_ε.

Meaning of δ

Given an NFA $N = (Q, \Sigma, \delta, q_0, F)$. What does it mean if:

$$\delta(q, w) = P$$

with $q \in Q$, $w \in \Sigma_\varepsilon$ and $P \in \mathscr{P}(Q)$?

Answer: First, $P \in \mathscr{P}(Q)$ means P is a *subset* of Q, written as $P \subseteq Q$. Next, $\delta(q, w)$ is therefore a set, namely the set consisting of precisely those possible states which N can end up in from q in one step by reading one symbol from Σ or by taking one ε step.

Acceptation of Words by NFAs

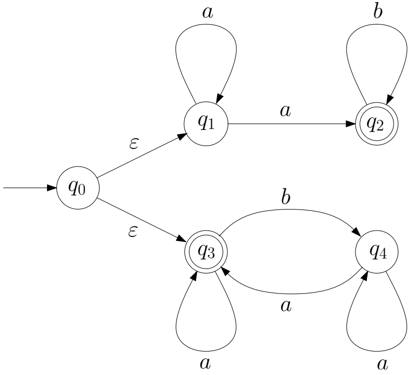
Definition: Let $N = (Q, \Sigma, \delta, q_0, F)$ an NFA and $w \in \Sigma^*$.
N accepts (recognizes) $w = a_1 a_2 \cdots a_n$ iff there are $r_0, \dots, r_n \in Q$ such that:

1. $r_0 = q_0$,
2. $r_{i+1} \in \delta(r_i, a_{i+1})$, for each $i \geq 0$,
3. $r_n \in F$.

$$\boxed{\delta^*(q, w) = \{q' \in Q \mid \text{there is a path from } q \text{ to } q' \text{ labeled by } w\}}$$

NB 1: Some a_i can indicate the empty string ε!
NB 2: "Neat" recursive definition of δ^* is possible but is complicated by ε transitions.

Example



$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_3, q_4\} \\ \Sigma &= \{a, b\} \\ F &= \{q_2, q_3\} \end{aligned}$$

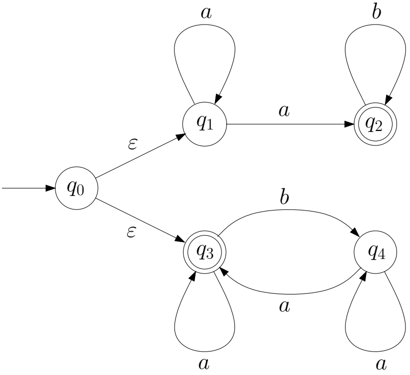
δ	a	b	ε
q_0	$\emptyset$	$\emptyset$	$\{q_1, q_3\}$
q_1	$\{q_1, q_2\}$	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\{q_2\}$	$\emptyset$
q_3	$\{q_3\}$	$\{q_4\}$	$\emptyset$
q_4	$\{q_3, q_4\}$	$\emptyset$	$\emptyset$

Language recognized by an NFA

Definition: Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. The language $L(N)$ recognized by N is defined as:

$$\begin{aligned} L(N) &= \{w \in \Sigma^* \mid N \text{ accepts } w\} \\ &= \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}. \end{aligned}$$

Example



Examples: $\delta^*(q_0, a) = \{q_1, q_2, q_3\}$

$\delta^*(q_0, bba) = \emptyset$

$\delta^*(q_1, abb) = \{q_2\}$

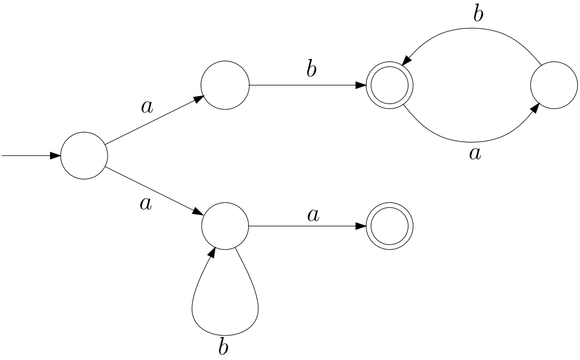
$\delta^*(q_1, aaa) = \{q_1, q_2\}$

Exercise

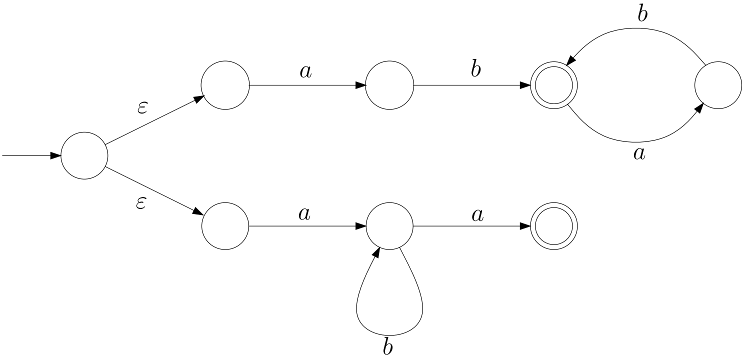
Construct an NFA that accepts the following language with $\Sigma = \{a, b\}$:

$$\{(ab)^n \mid n > 0\} \cup \{ab^m a \mid m \geq 0\}.$$

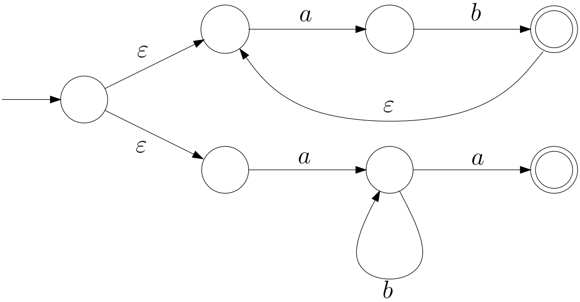
Solution



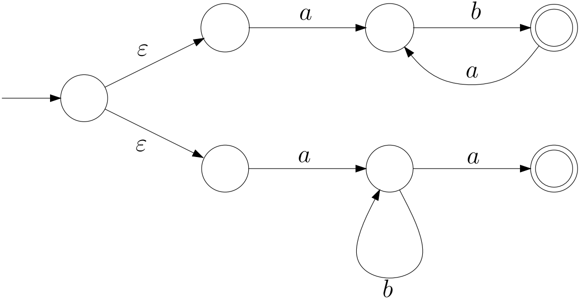
Solution



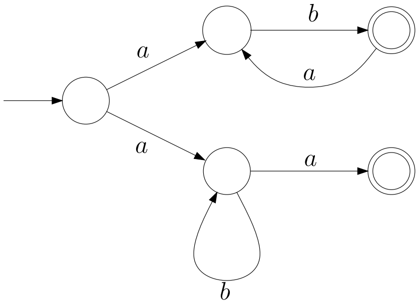
Solution



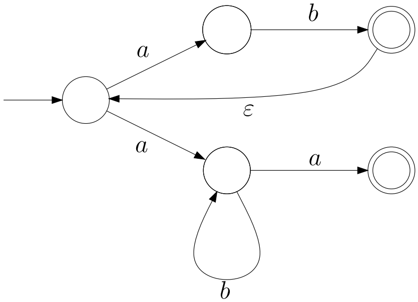
Solution



Solution



Not a solution!



This is NOT a correct solution! Why not?

Which languages are recognized by NFAs?

Definition (again): A language L is regular iff there exists a DFA M such that $L = L(M)$.

Question: Which languages are recognized by NFAs?

Remarkable answer: The class of languages recognized by NFAs is also the class of regular languages!

Equivalence of NFAs and DFAs

(see also video Equivalence of Deterministic and Nondeterministic FSMs)

Definition: Automata M_1 and M_2 are equivalent iff $L(M_1) = L(M_2)$.

Theorem: For every NFA, an equivalent DFA exists.

Corollary: The class of languages recognized by NFAs is the class of regular languages.

Proof sketch

- First we prove that for every NFA N without ε transitions a DFA M can be constructed such that $L(M) = L(N)$.
  This is done by having M execute all possible transitions that N may do from a set of states in parallel.
- Then we treat the case of NFAs with ε transitions.
  While doing this, we take into account that during these transitions, ε might be "read". Besides all states that N can reach by reading a symbol, M also determines all states that can be reached from these states by following them by ε transitions.

NFAs without ε transitions

Definition: An NFA $(Q, \Sigma, \delta, q_0, F)$ without ε transitions is an NFA such that:

$$\delta : Q \times \Sigma \to \mathscr{P}(Q).$$

The function δ^* is inductively defined as:

$$\begin{aligned}
\delta^*(q, \varepsilon) &= \{q\} \\
\delta^*(q, wa) &= \delta(p_0, a) \cup \cdots \cup \delta(p_n, a) \\
&\qquad\qquad \text{with } \{p_0, \ldots, p_n\} = \delta^*(q, w) \\
&= \bigcup_{p \in \delta^*(q,w)} \delta(p, a).
\end{aligned}$$

Exercise

Let $a \in \Sigma$. Do we generally have $\delta(q,a) = \delta^*(q,a)$ for:

1. DFAs?
2. NFAs without ε transitions?
3. NFAs with ε transitions?

From NFA without ε transitions to DFA

Theorem: For every NFA without ε transitions there exists an equivalent DFA.

Idea: Construct a DFA M that for each input word w performs all computation paths of N in parallel. The word w is accepted by M if at least one of the computation paths of N ends in an accepting state.

Constructing a DFA from an NFA

Let $N = (Q_N, \Sigma, \delta_N, q_N, F_N)$ be an NFA without ε transitions.

Define $M = (Q_M, \Sigma, \delta_M, q_M, F_M)$ such that:

$$\begin{aligned} Q_M &= \mathscr{P}(Q_N) \\ q_M &= \{q_N\} \\ F_M &= \{X \subseteq Q_N \mid X \cap F_N \neq \emptyset\} \end{aligned}$$

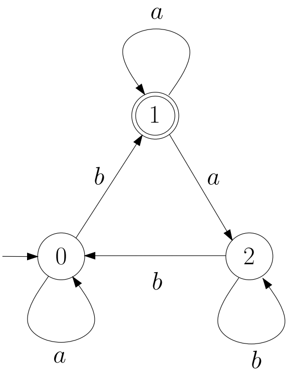
For each $X = \{x_0, \ldots, x_n\} \subseteq Q_N$ and each $a \in \Sigma$:

$$\begin{aligned} \delta_M(X, a) &= \delta_N(x_0, a) \cup \cdots \cup \delta_N(x_n, a) \\ &= \bigcup_{x \in X} \delta_N(x, a) \end{aligned}$$

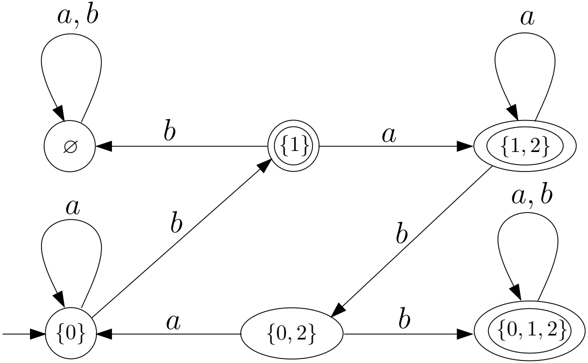
Note: $\delta_M(\emptyset, a) = \emptyset$ for all $a \in \Sigma$!

Exercise

Transform the following NFA to an equivalent DFA:



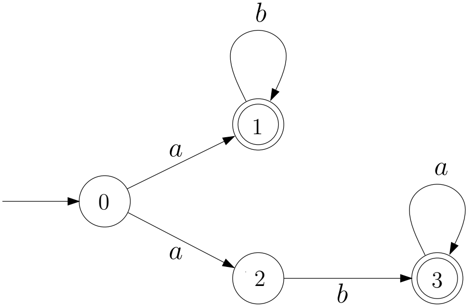
Solution



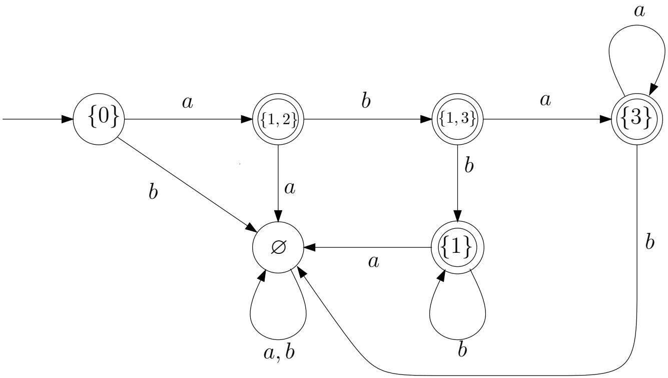
Note: States $\{2\}$ and $\{0,1\}$ have been left out. Why?

Exercise

Transform the following NFA into a DFA.



Solution



Overview definitions

Let $N = (Q_N, \Sigma, \delta_N, q_N, F_N)$ be an NFA without ε transitions. Then:

- $$\begin{array}{rcl} \delta_N^*(q, \varepsilon) & = & \{q\} \\ \delta_N^*(q, wa) & = & \displaystyle\bigcup_{p \in \delta_N^*(q,w)} \delta_N(p, a) \end{array}$$

- $M = (Q_M, \Sigma, \delta_M, q_M, F_M)$ is defined as:

$$\begin{array}{rclrcl} Q_M & = & \mathscr{P}(Q_N) & F_M & = & \{X \subseteq Q_N \mid X \cap F_N \neq \emptyset\} \\ q_M & = & \{q_N\} & \delta_M(X, a) & = & \displaystyle\bigcup_{x \in X} \delta_N(x, a) \end{array}$$

- $$\begin{array}{rcl} \delta_M^*(X, \varepsilon) & = & X \\ \delta_M^*(X, wa) & = & \delta_M(\delta_M^*(X, w), a) \end{array}$$

ε closure

Definition: Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Then $E(q)$ is the ε closure of a state $q \in Q$, and $E(X)$ of a set of states $X \subseteq Q$ is defined as follows:

$$
\begin{aligned}
E(q) &= \{q' \in Q \mid q' \text{ can be reached from } q \\
&\qquad\qquad \text{via } 0 \text{ or more } \varepsilon \text{ transitions}\} \\
E(X) &= \{q' \in Q \mid \text{there exists a } q \in X \text{ such that } q' \in E(q)\}
\end{aligned}
$$

Note: It always holds that $q \in E(q)$; after all, q can be reached from q in 0 ε transitions.

DFA for an NFA with ε transitions

Let $N = (Q_N, \Sigma, \delta_N, q_N, F_N)$ be an NFA.

Define $M = (Q_M, \Sigma, \delta_M, q_M, F_M)$ such that:

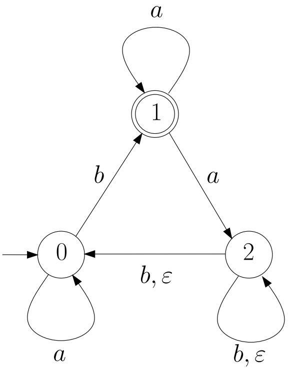
$$\begin{aligned} Q_M &= \mathscr{P}(Q_N) \\ q_M &= E(q_N) \\ F_M &= \{X \subseteq Q_N \mid X \cap F_N \neq \emptyset\} \end{aligned}$$

and for each $X = \{x_0, \ldots, x_n\} \subseteq Q_N$ and each $a \in \Sigma$:

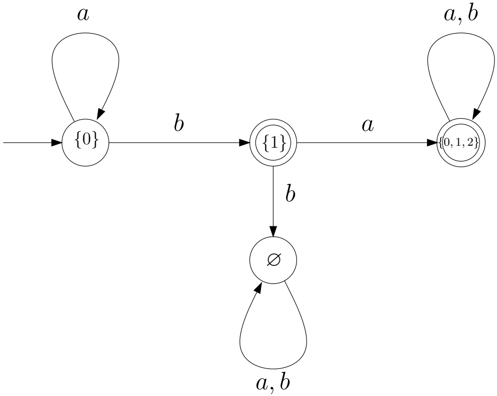
$$\begin{aligned} \delta_M(X, a) &= E(\delta_N(x_0, a)) \cup \ldots \cup E(\delta_N(x_n, a)) \\ &= \bigcup_{x \in X} E(\delta_N(x, a)) \end{aligned}$$

Exercise

Transform the following NFA into an equivalent DFA:



Soluton



Regular operations

Definition: The language operations *union* ($\cup$), *concatenation* ($\circ$) and *star* (*) are called the regular operations.

If L_1 and L_2 are languages, we write these operations as $L_1 \cup L_2$, $L_1 \circ L_2$ en L_1^*.

Note: We usually write $L_1 L_2$ instead of $L_1 \circ L_2$.

Closure under Union

Theorem 1.45: The class of regular languages is closed under union.

Sipser, pp. 59, 60.

(Also see for closure under $\cup$, $\circ$ and * the video Closure of the Regular Operations)

Closure under Concatenation

Theorem 1.47: The class of regular languages is closed under concatenation.

Sipser, pp. 60, 61.

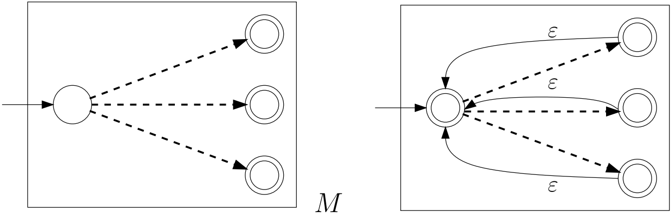
Closure under Star

Theorem 1.49: The class of regular languages is closed under star.

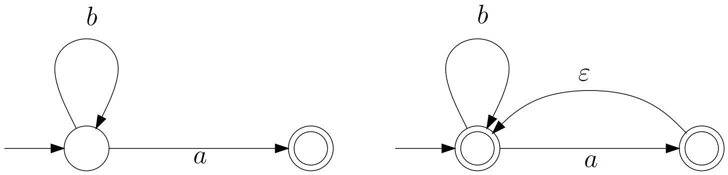
Sipser, pp. 62, 63.

Exercise (1.15)

Let M be an NFA. Why does the following construction for star NOT generally work?



Counterexample



Let N be the left NFA and N' the right NFA.
Note that $b \notin L(N)^*$, while $b \in L(N')$

Exercise

Prove or disprove the following statement:

Let L be a regular language. Then the language

$$L' = \{wv \mid w \in L \text{ and } |v| = 2\}$$

is also regular.

Exercise

Let M be an NFA. Show that there exists an equivalent NFA M' such that:

1. M' has precisely one start state,
2. M' has precisely one accept state,
3. no transition ends in the start state,
4. no transition starts in the accept state.

Solution