Automatentheorie und formale Sprachen
endliche Automaten

Dozentin: Wiebke Petersen

13.5.2009

What we know so far about formal languages

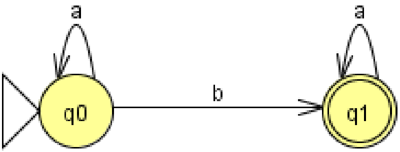
- Formal languages are sets of words (NL: sets of sentences) which are strings of symbols (NL: words).
- Everything in the set is a "grammatical word", everything else isn't.
- Some formal languages, namely the regular ones, can be described by regular expressions
  Example: $L(a^{\star} \bullet b \bullet a^{\star} \bullet b \bullet a^{\star})^{\star}$ is the regular language consisting of all words over the alphabet $\{a, b\}$ which contain an even number of b's.
- Not all formal languages are regular (We have not proven this yet!).
  Example: The formal language of all palindromes over the alphabet $\{a, b\}$ is not regular.

Deterministic finite-state automaton (DFSA)

Definition

A deterministic finite-state automaton is a tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ *with:*

1. *a finite, non-empty set of states* Q
2. *an alphabet* Σ *with* $Q \cap \Sigma = \emptyset$
3. *a partial transition function* $\delta : Q \times \Sigma \rightarrow Q$
4. *an initial state* $q_0 \in Q$ *and*
5. *a set of final/accept states* $F \subseteq Q$.



accepts: $L(a^\star b a^\star)$

Language accepted by an automaton

Definition

A *situation* *of a finite-state automaton* $\langle Q, \Sigma, \delta, q_0, F \rangle$ *is a triple* (x, q, y) *with* $x, y \in \Sigma^*$ *and* $q \in Q$.

Language accepted by an automaton

Definition

A situation of a finite-state automaton $\langle Q, \Sigma, \delta, q_0, F \rangle$ is a triple (x, q, y) with $x, y \in \Sigma^*$ and $q \in Q$.
Situation (x, q, y) produces situation (x', q', y') in one step if there exists an $a \in \Sigma$ such that $x' = xa$, $y = ay'$ and $\delta(q, a) = q'$, we write $(x, q, y) \vdash (x', q', y')$ ($(x, q, y) \vdash^* (x', q', y')$ as usual).

Language accepted by an automaton

Definition

A situation of a finite-state automaton $\langle Q, \Sigma, \delta, q_0, F\rangle$ *is a triple* (x, q, y) *with* $x, y \in \Sigma^*$ *and* $q \in Q$.
Situation (x, q, y) *produces situation* (x', q', y') *in one step if there exists an* $a \in \Sigma$ *such that* $x' = xa$, $y = ay'$ *and* $\delta(q, a) = q'$, *we write* $(x, q, y) \vdash (x', q', y')$ ($(x, q, y) \vdash^* (x', q', y')$ *as usual).*

Definition

A word $w \in \Sigma^*$ *gets accepted by an automaton* $\langle Q, \Sigma, \delta, q_0, F\rangle$ *if* $(\epsilon, q_0, w) \vdash^* (w, q_n, \epsilon)$ *with* $q_n \in F$.

Language accepted by an automaton

Definition

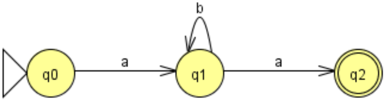
A situation of a finite-state automaton $\langle Q, \Sigma, \delta, q_0, F \rangle$ is a triple (x, q, y) with $x, y \in \Sigma^*$ and $q \in Q$.
Situation (x, q, y) produces situation (x', q', y') in one step if there exists an $a \in \Sigma$ such that $x' = xa$, $y = ay'$ and $\delta(q, a) = q'$, we write $(x, q, y) \vdash (x', q', y')$ ($(x, q, y) \vdash^* (x', q', y')$ as usual).

Definition

A word $w \in \Sigma^*$ gets accepted by an automaton $\langle Q, \Sigma, \delta, q_0, F \rangle$ if $(\epsilon, q_0, w) \vdash^* (w, q_n, \epsilon)$ with $q_n \in F$.
An automaton accepts a language iff it accepts every word of the language.

partial/total transition function

FSA with partial transition function



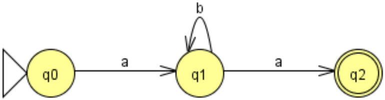
accepts $ab^{\star}a$

	a	b
q0	q1	
q1	q2	q1
q2		

transition table

partial/total transition function

FSA with partial transition function

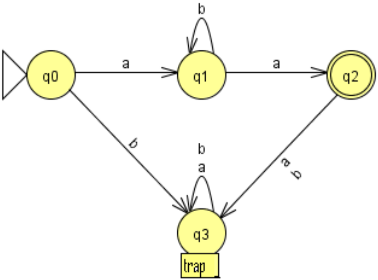


accepts $ab^{\star}a$

	a	b
q0	q1	
q1	q2	q1
q2		

transition table

FSA with complete transition function



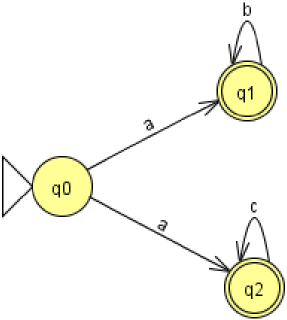
accepts $ab^{\star}a$

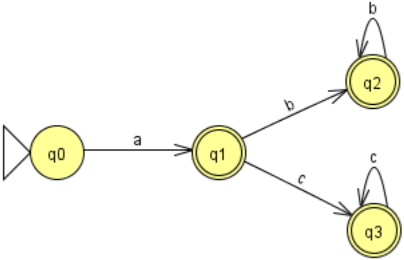
	a	b
q0	q1	q3
q1	q2	q1
q2	q3	q3
q3	q3	q3

transition table

Example DfSA / NDFSA

The language $L(ab^{\star} + ac^{\star})$ is accepted by





Nondeterministic finite-state automaton NDFSA

Definition

A nondeterministic finite-state automaton is a tuple $\langle Q, \Sigma, \Delta, q_0, F \rangle$ *with:*

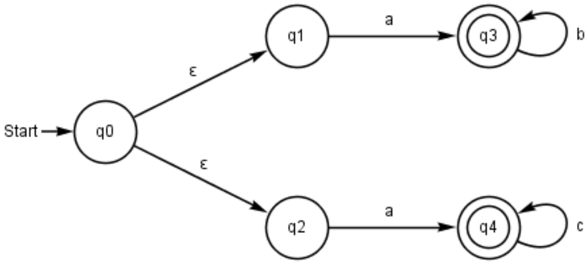
1. *a finite non-empty set of states* Q
2. *an alphabet* Σ *with* $Q \cap \Sigma = \emptyset$
3. **a transition relation** $\Delta \subseteq Q \times \Sigma \times Q$
4. *an initial state* $q_0 \in Q$ *and*
5. *a set of final states* $F \subseteq Q$.

Theorem

A language L can be accepted by a DFSA iff L can be accepted by a NFSA.

Note: Even automatons with ϵ-transitions accept the same languages like NDFSA's.

Automaton with ϵ-transition



Exercise 1

Überlegen sie sich, wie ein nichtdeterministischer endlicher Automat mit ϵ-Übergängen systematisch in einen nichtdeterministischen endlichen Automaten ohne ϵ-Übergänge überführt werden kann.

oder

Exercise 2

Schreiben sie ein Prolog-Programm für deterministische und nichtdeterministische endliche Automaten. (Prädikate in der Faktenbasis: `transition/3`, `initial/1`, `final/1`).

Vergessen sie bitte nicht, den auf der Webseite verlinkten Text aus Klabunde 1998 zu lesen.