Einführung in die Computerlinguistik – reguläre Sprachen und endliche Automaten

Dozentin: Wiebke Petersen

Foliensatz 3

Describing formal languages by enumerating all words

- Peter says that Mary has fallen off the tree.
- Oskar says that Peter says that Mary has fallen off the tree.
- Lisa says that Oskar says that Peter says that Mary has fallen off the tree.
- ...

Describing formal languages by enumerating all words

- Peter says that Mary has fallen off the tree.
- Oskar says that Peter says that Mary has fallen off the tree.
- Lisa says that Oskar says that Peter says that Mary has fallen off the tree.
- ...

The set of strings of a natural language is infinite.

The enumeration does not gather generalizations.

Describing formal languages by grammars

Grammar

- A formal grammar is a generating device which can generate (and analyze) strings/words.
- Grammars are finite rule systems.
- The set of all strings generated by a grammar is the formal language generated by the grammar.

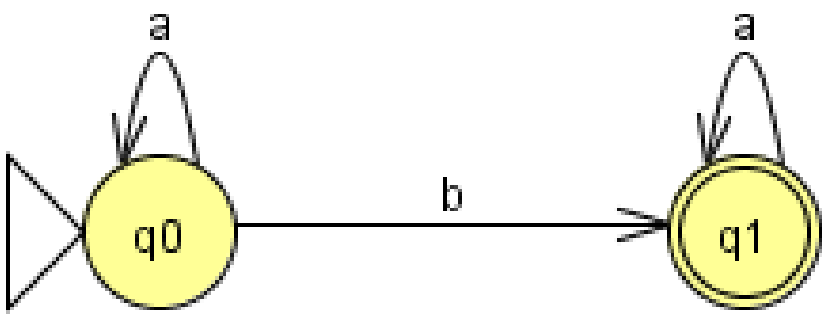
S	$\rightarrow$	NP VP	VP	$\rightarrow$	V	NP	$\rightarrow$	D N
D	$\rightarrow$	the	N	$\rightarrow$	cat	V	$\rightarrow$	sleeps

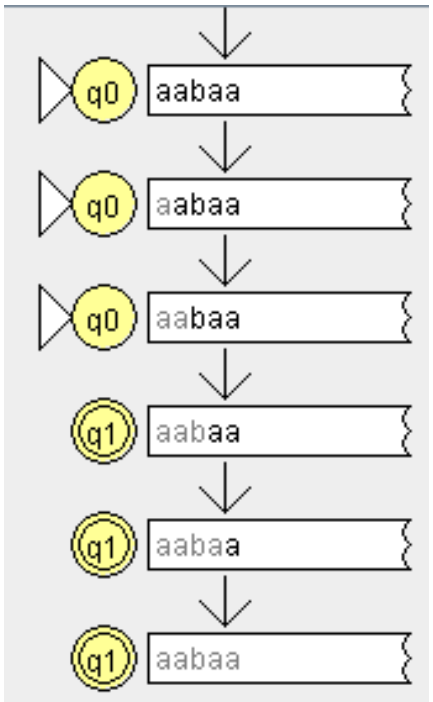
Generates: the cat sleeps

Describing formal languages by automata

Automaton

- An automaton is a recognizing device which accepts strings/words.
- The set of all strings accepted by an automaton is the formal language accepted by the automaton.





Sprachbeschreibung

Zusammenhang nach Klabunde 1998

"Formale Sprachen besitzen strukturelle Eigenschaften.

Grammatiken sind Erzeugungssysteme für formale Sprachen.

Automaten sind Erkennungssysteme für formale Sprachen."

Vorsicht: per Definition besitzen formale Sprachen keine strukturellen Eigenschaften; uns interessieren aber nur solche mit strukturellen Eigenschaften, die von einer Grammatik erzeugt werden können. Außerdem können Grammatiken auch für die Analyse (Erkennung) formaler Sprachen und endliche Automaten für ihre Erzeugung genutzt werden.

Regular expressions

RE: syntax

The set of regular expressions RE_Σ over an alphabet $\Sigma = \{a_1, \ldots, a_n\}$ is defined by:

- $\underline{\emptyset}$ is a regular expression.
- ε is a regular expression.
- $a_1, \ldots, a_n$ are regular expressions
- If a and b are regular expressions over Σ then
  - $(a + b)$
  - $(a \bullet b)$
  - $(a^\star)$

  are regular expressions too.

(The brackets are frequently omitted w.r.t. the following dominance scheme: $\star$ dominates $\bullet$ dominates $+$)

Regular expressions

RE: semantics

Each regular expression r over an alphabet Σ describes a formal language $L(r) \subseteq \Sigma^*$.

Regular languages are those formal languages which can be described by a regular expression.

The function L is defined inductively:

- $L(\underline{\emptyset}) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$, $L(a_i) = \{a_i\}$
- $L(a + b) = L(a) \cup L(b)$
- $L(a \bullet b) = L(a) \circ L(b)$
- $L(a^\star) = L(a)^*$

Aufgaben für Übungssitzung (1)

Exercise 1

Find a regular expression which describes the regular language L (be careful: at least one language is not regular!)

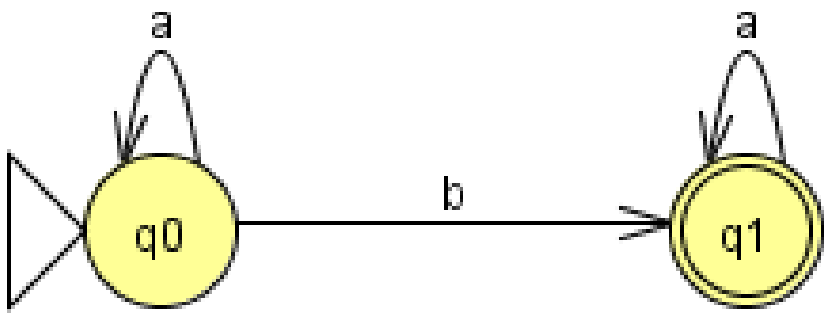
- *L is the language over the alphabet* $\{a, b\}$ *with* $L = \{aba, \varepsilon, aa, bbb\}$.
- *L is the language over the alphabet* $\{a, b\}$ *which consists of all words which start with a nonempty string of b's followed by at least one a followed by any number of b's*
- *L is the language over the alphabet* $\{a, b\}$ *such that every a has a b immediately to its left.*
- *L is the language over the alphabet* $\{a, b\}$ *which consists of all words which contain an uneven number of a's.*
- *L is the language of all palindromes over the alphabet* $\{a, b\}$.

Deterministic finite-state automaton (DFSA)

Definition

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

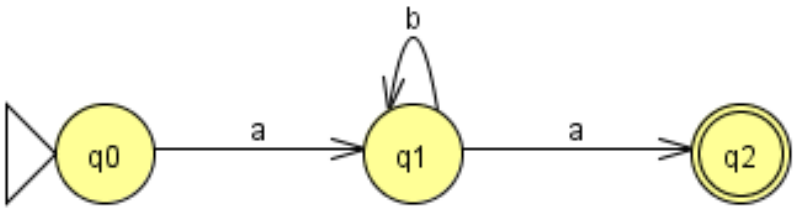
1. *a finite, non-empty set of states* Q
2. *an alphabet* Σ *with* $Q \cap \Sigma = \emptyset$
3. *a partial transition function* $\delta : Q \times \Sigma \rightarrow Q$
4. *an initial state* $q_0 \in Q$ *and*
5. *a set of final/accept states* $F \subseteq Q$.



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

partial/total transition function

FSA with partial transition function



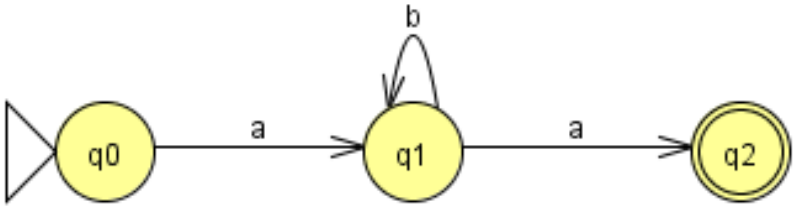
accepts $ab^{\star}a$

	a	b
q0	q1	
q1	q2	q1
q2		

transition table

partial/total transition function

FSA with partial transition function

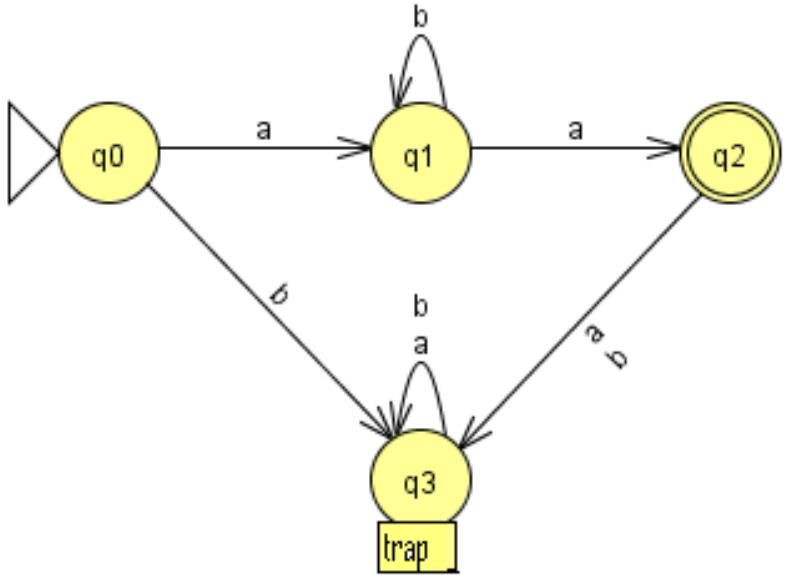


accepts $ab^{\star}a$

	a	b
q0	q1	
q1	q2	q1
q2		

transition table

FSA with complete transition function



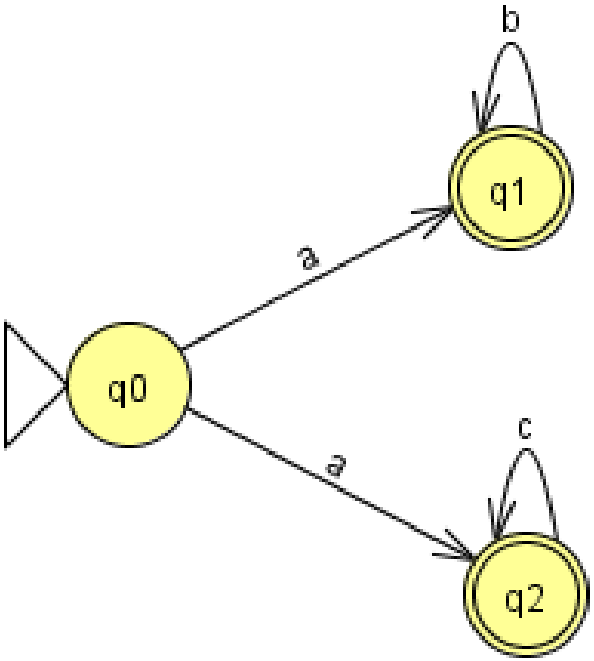
accepts $ab^{\star}a$

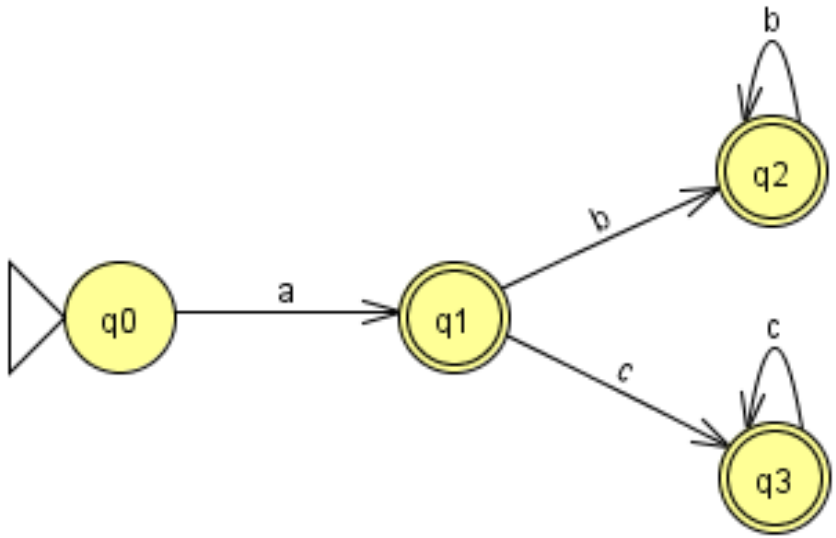
	a	b
q0	q1	q3
q1	q2	q1
q2	q3	q3
q3	q3	q3

transition table

Example DFSA / NFSA

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





Nondeterministic finite-state automaton NFSA

Definition

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

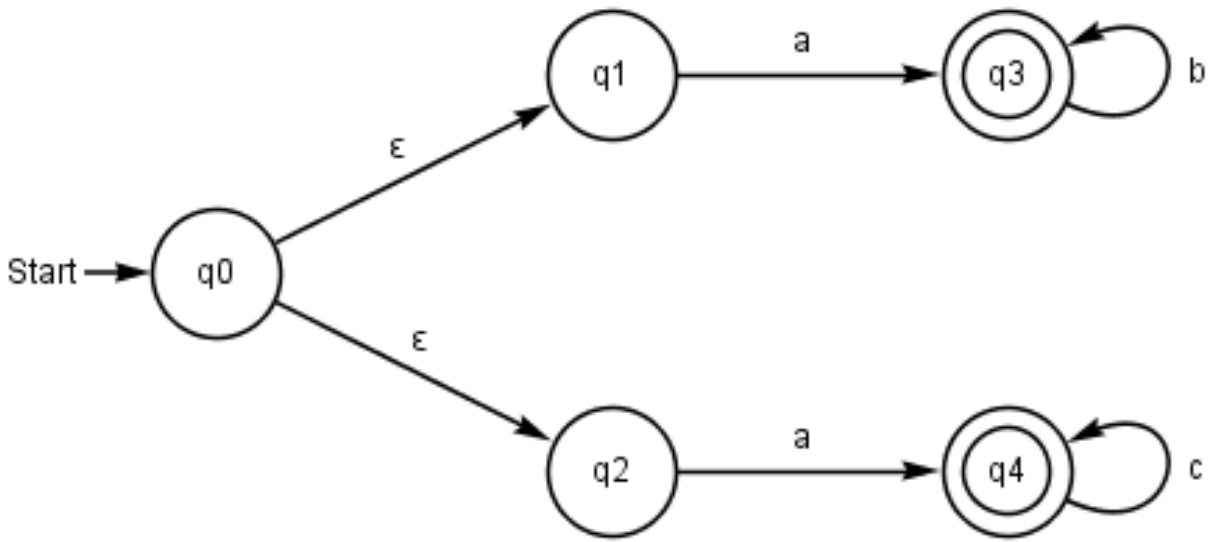
1. *a finite non-empty set of states* Q
2. *an alphabet* Σ *with* $Q \cap \Sigma = \emptyset$
3. **a transition relation** $\Delta \subseteq Q \times \Sigma \times Q$
4. *an initial state* $q_0 \in Q$ *and*
5. *a set of final states* $F \subseteq Q$.

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 DFSA's.

Automaton with ε-transition



Aufgaben für Übungssitzung (2)

Exercise 2

Give an FSA for each of the following languages over the alphabet $\{a, b\}$ *(and try to make it deterministic):*

- $L = \{w|$ *between each two 'a's in w there are at least three 'b's*$\}$
- $L = \{w|w$ *is any word except "bab"*$\}$
- $L = \{w|w$ *does not contain the infix "ab"*$\}$
- $L = \{w|w$ *contains at most two 'a's*$\}$
- $L = \{w|w$ *contains an uneven number of 'a's*$\}$
- $L((b^\star a)^\star ab^\star)$
- $L(a^\star(ab)^\star)$
- $L(aa^\star b)$.
- $L((bb^\star + ab^\star a))$