Einführung in die Computerlinguistik

Endliche Automaten,
rechtslineare Grammatiken
und reguläre Sprachen (1)

Vokabular der Theorie der formalen Sprachen (1)

Alphabet Σ: eine nicht-leere Menge von *Zeichen*

Wort: eine endliche Folge/Kette $x_1 \dots x_n$ von Zeichen.

leeres Wort: das aus null Zeichen bestehende Wort ϵ. (Vorsicht: $\emptyset \neq \epsilon$)

Länge eines Worts $|w|$: Zahl der Zeichen in einem Wort w.

Sternbildung: Die Menge aller Wörter über einem Alphabet Σ wird mit Σ^* bezeichnet. ($\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$)

Konkatenation von Wörtern: Seien $w_1 = abc \in \Sigma^*$ und $w_2 = de \in \Sigma^*$ dann ist $w_1 \circ w_2 = abcde \in \Sigma^*$ die Konkatenation von w_1 und w_2.

Potenz eines Wortes: a^i ist die i-fache Konkatenation des Wortes a mit sich selbst. Beispiel: $a^3 = a \circ a \circ a$

formale Sprachen

Eine **formale Sprache** L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot -, - \cdots, \ldots, - - \cdot\cdot\}$
- Sprache L_{arith} der vollständig geklammerten arithmetischen Ausdrücke über dem Alphabet $\Sigma_Z \cup \{+, -, \cdot, :\}$.

Sprachbeschreibung durch Angabe einer Grammatik

Unter einer Grammatik verstehen wir einen generativen Mechanismus, der es gestattet Zeichenketten zu erzeugen.

Die Menge aller durch die Grammatik erzeugbaren Zeichenketten ist die von der Grammatik generierte Sprache.

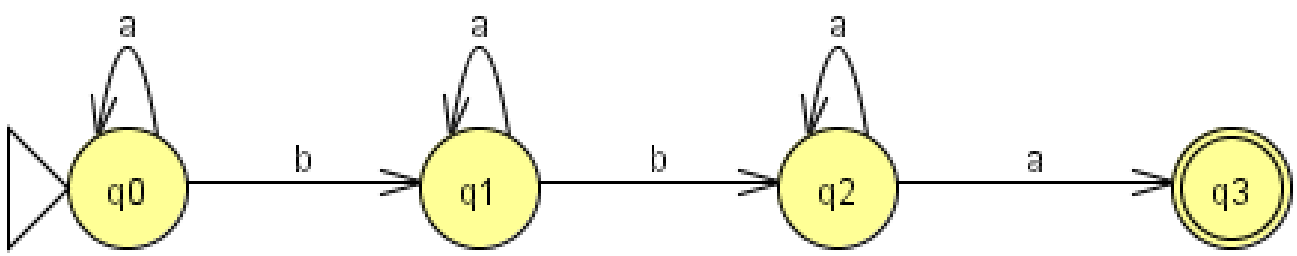
Grammatiken sind endliche Regelsysteme.

Sprachbeschreibung durch Automaten

Automaten sind Erkennungsmechanismen die Zeichenketten analysieren und entscheiden, ob sie diese Zeichenkette akzeptieren.

Die Menge der durch einen Automaten akzeptierten Zeichenketten ist die von dem Automaten akzeptierte Sprache.

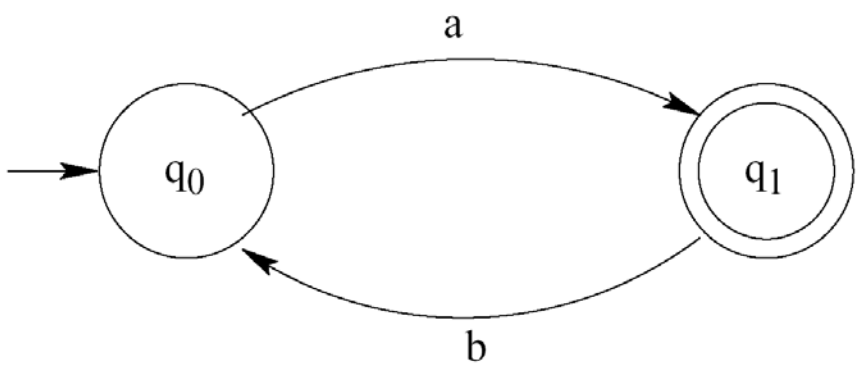
Beispiel (endlicher Automat):



deterministische endliche Automaten DEA

Definition 4. *Ein* **deterministischer endlicher Automat** *ist ein 5-Tupel* $\langle \Phi, \Sigma, \delta, S, F \rangle$ *bestehend aus:*

1. *einem* **Zustandsalphabet** Φ
2. *einem* **Eingabealphabet** Σ *mit* $\Phi \cap \Sigma = \emptyset$
3. *einer* **Übergangsfunktion** $\delta : \Phi \times \Sigma \to \Phi$
4. *einem* **Startzustand** q_0 *und*
5. *einer Menge von* **Endzuständen** $F \subset \Phi$.



Sprache, die von einem Automaten akzeptiert wird

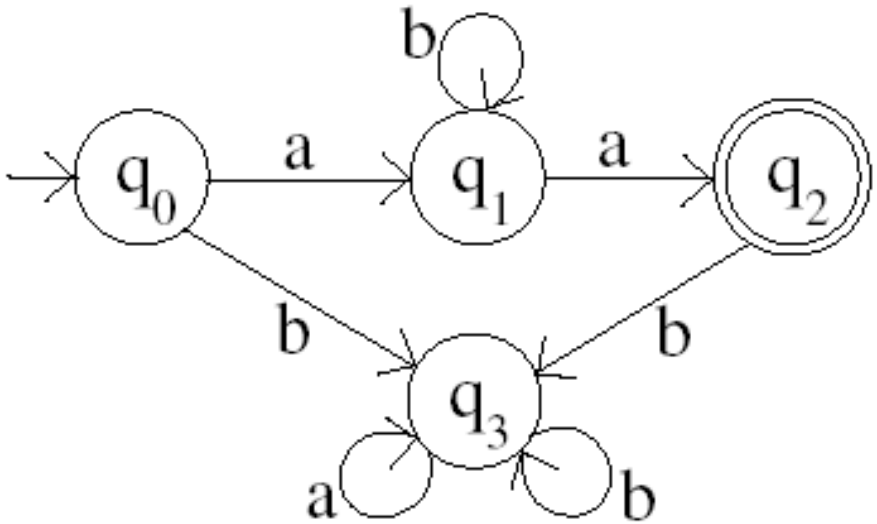
Definition 5. *Eine* **Situation** *eines Automaten* $\langle \Phi, \Sigma, \delta, q_0, F \rangle$ *ist ein Tripel* (x, q, y) *mit* $x, y \in \Sigma^*$ *und* $q \in \Phi$.

Situation (x, q, y) **produziert** *Situation* (x', q', y') **in einem Schritt**, *wenn* $\exists a \in \Sigma$ *so daß* $x' = xa$, $y = ay'$ *und* $\delta(q, a) = q'$, *wir schreiben* $(x, q, y) \vdash (x', q', y')$ $((x, q, y) \vdash^* (x', q', y')$ *wie üblich).*

Definition 6. *Ein Wort* $w \in \Sigma^*$ *wird von einem Automaten* $\langle \Phi, \Sigma, \delta, q_0, F \rangle$ **akzeptiert**, *wenn* $(\epsilon, q_0, w) \vdash^* (w, q_n, \epsilon)$ *wobei* $q_n \in F$.

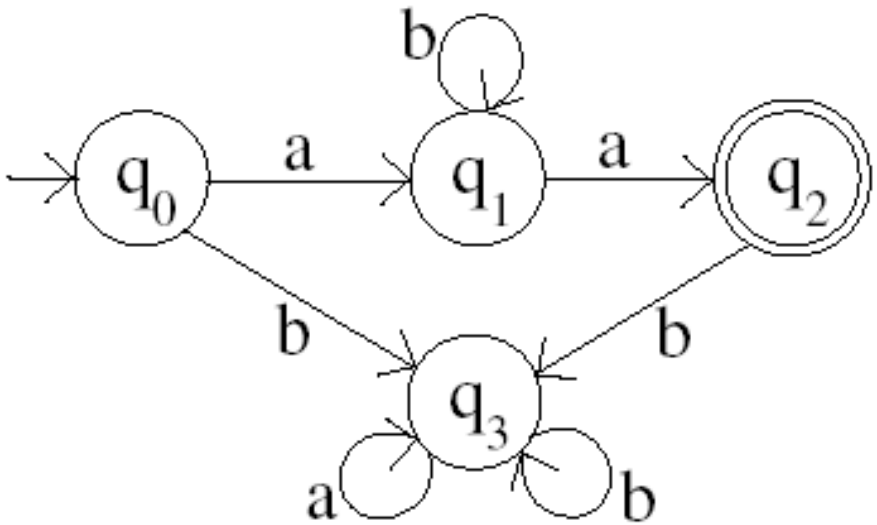
Ein Automat **akzeptiert eine Sprache**, *wenn es jedes Wort dieser Sprache akzeptiert.*

Beispiel



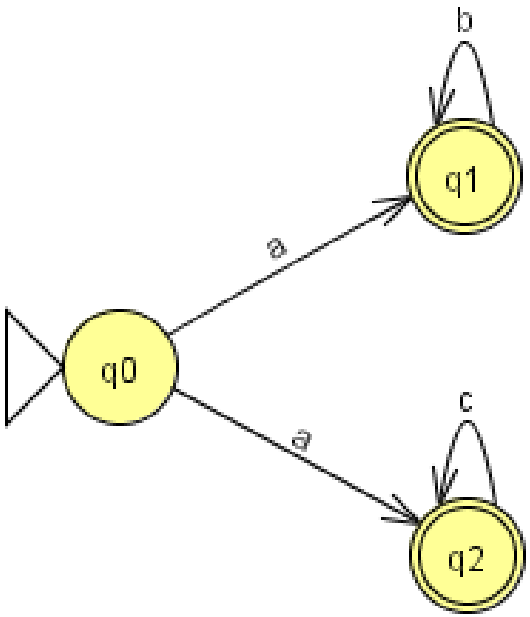
Ein endlicher Automat ist deterministisch, wenn es, egal in welchem Zustand des Automaten man sich gerade befindet, für jede Eingabe aus dem Alphabet, immer einen eindeutigen Zustand gibt, in den man übergehen muß.

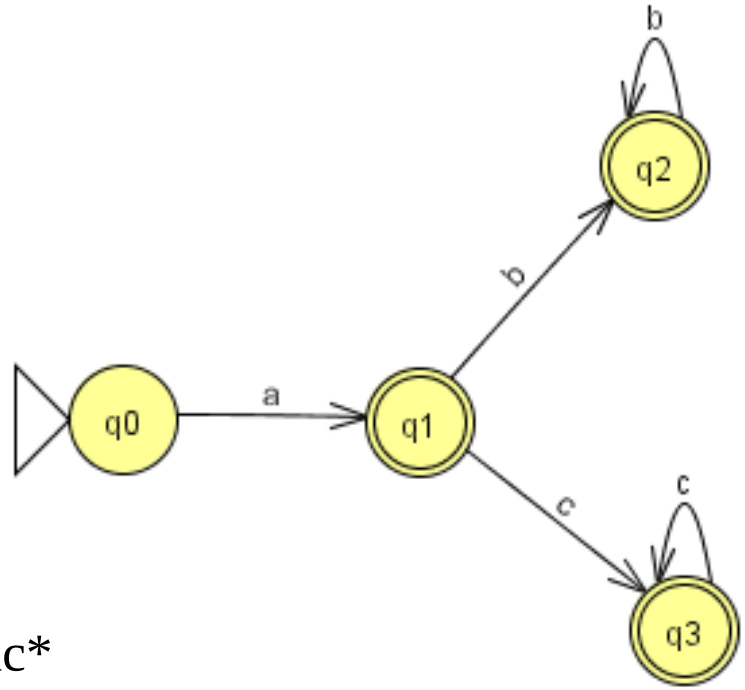
Beispiel



akzeptiert ab^*a

schwach determinstische endliche Automaten





ab* ∪ ac*

Ein endlicher Automat ist schwach deterministisch, wenn es für keine Situation des Automaten zwei verschiedene Situationen gibt, die in einem Schritt produziert werden können.

Nichtdeterministische endliche Automaten NDEA

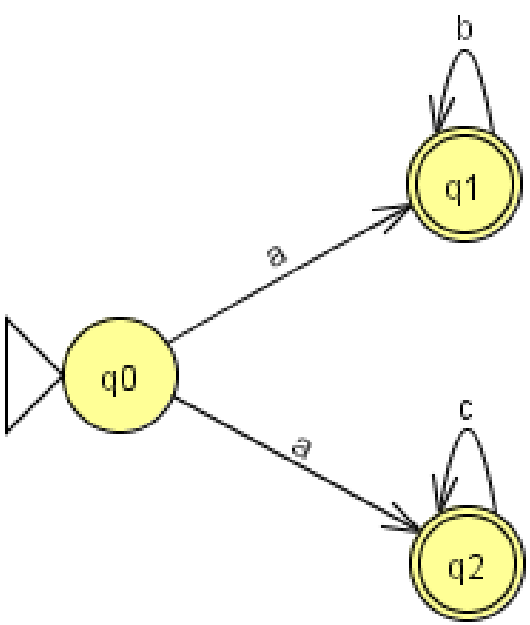
Definition 7. *Ein* ***nichtdeterministischer endlicher Automat*** *ist ein 5-Tupel* $\langle \Phi, \Sigma, \Delta, q_0, F \rangle$ *bestehend aus:*

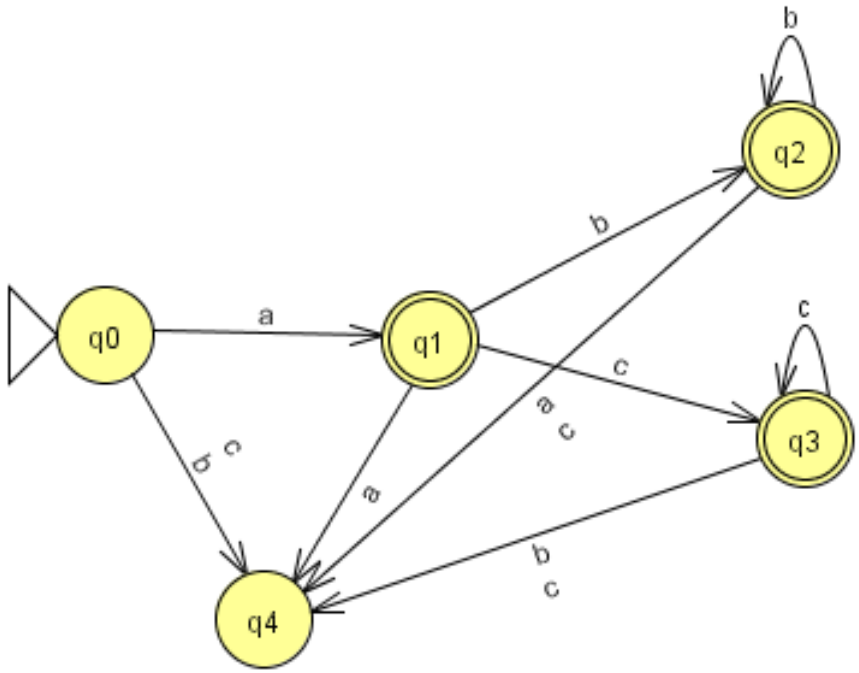
1. *einem* ***Zustandsalphabet*** Φ
2. *einem* ***Eingabealphabet*** Σ *mit* $\Phi \cap \Sigma = \emptyset$
3. *einer* ***Übergangsrelation*** $\Delta \subseteq \Phi \times \Sigma \times \Phi$
4. *einem* ***Startzustand*** q_0 *und*
5. *einer Menge von* ***Endzuständen*** $F \subset \Phi$.

Zu jedem NDEA gibt es einen DEA, der die gleiche Sprache akzeptiert.

Beispiel DEA / NDEA

Die Sprache $ab^* \cup ac^*$ wird akzeptiert von





Übrigens: auch Automaten mit ϵ-Übergängen akzeptieren die gleichen Sprachen wie DEA's und NDEA's.