Automaten und Formale Sprachen

SoSe 2007 in Trier

Henning Fernau
Universität Trier
fernau@informatik.uni-trier.de

- Organisatorisches
- Einführung
- **Endliche Automaten und reguläre Sprachen**
- Kontextfreie Grammatiken und kontrextfreie Sprachen
- Chomsky-Hierarchie

Endliche Automaten und reguläre Sprachen

1. Deterministische endliche Automaten

2. Nichtdeterministische endliche Automaten

3. Reguläre Ausdrcke

4. Nichtreguläre Sprachen

5. Algorithmen mit / für endliche Automaten

Reguläre Sprachen

Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär* gdw. es ein endliches Monoid $(M, \circ, e)$, einen Monoidmorphismus $h : (\Sigma^*, \cdot, \lambda) \to (M, \circ, e)$ sowie eine endliche Menge $F \subseteq M$ gibt mit

$$L = \{w \in \Sigma^* \mid h(w) \in F\}.$$

zu abstrakt ??

Ein **deterministischer endlicher Automat** oder DEA
wird beschrieben durch ein Quintupel

$$A = (Q, \Sigma, \delta, q_0, F)$$

wobei gilt:

Q: endliche Menge von *Zuständen* (Zustandsalphabet)

Σ: endliche Menge von *Eingabezeichen* (Eingabealphabet)

$\delta : Q \times \Sigma \to Q$: *Überführungsfunktion*

$q_0 \in Q$: *Anfangszustand*

$F \subseteq Q$: *Endzustände*

Überführungstafel

Ein endlicher Automat kann vollständig durch seine *Überführungstafel* beschrieben werden.

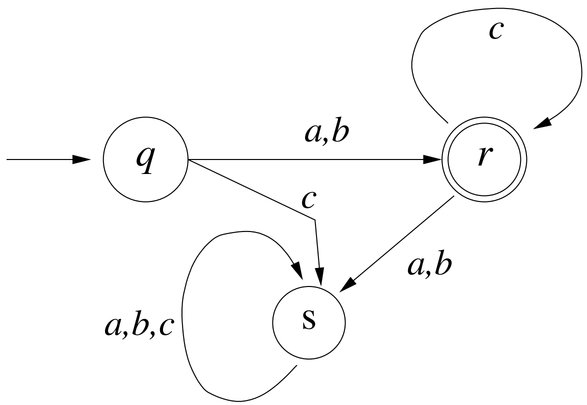
Beispiel: Betrachte:

δ	a	b	c
$\rightarrow q$	r	r	s
$r \rightarrow$	s	s	r
s	s	s	s

In der ersten Zeile ist das Eingabealphabet beschrieben, in der ersten Spalte das Zustandsalphabet; der eingehende Pfeil kennzeichnet den Anfangszustand und der ausgehende den Endzustand (es könnten auch mehrere sein).

Automatengraph

Ein gerichteter, Σ-kantenbeschrifteter Graph mit Knotenmenge Q:



“Eingangspfeile” kennzeichnen den Startzustand und doppelte Umkreisungen die Endzustände.

Wie arbeitet ein DEA ?

Es sei $w = a_1 \dots a_n \in \Sigma^n$ das *Eingabewort* von A.
Die Arbeit von A auf w kann wie folgt beschrieben werden:

1. Setze $q = q_0$.

2. Für $x = a_1$ bis a_n tue:
   Setze $q := \delta(q, x)$

3. Akzeptiere w gdw. $q \in F$ gilt.

$L(A)$ bezeichnet die von A akzeptierte Sprache.

Exkurs: Relationen

Erinnerung: $R \subseteq X \times X$ heißt *(binäre) Relation* (auf X).

Die *Diagonale* $\Delta_X = \{(x, x) \mid x \in X\}$ ist eine spezielle Relation.
$R \subseteq X \times X$ heißt *reflexiv* gdw. $\Delta_X \subseteq R$.

Das *Produkt* zweiter Relationen R_1, R_2 auf X ist definiert durch:

$$R_1 \circ R_2 = \{(x, z) \in X \times X \mid \exists y \in X : (x, y) \in R_1 \wedge (y, z) \in R_2\}.$$

Satz: $(2^{X \times X}, \circ, \Delta_X)$ ist ein Monoid.

Eine Relation R auf X heißt *transitiv* gdw. $R \circ R \subseteq R$.

Wir können auch *Relationenpotenz*en rekursiv definieren:
$R^1 := R$ und $R^n := R \circ R^{n-1}$ für $n > 1$.

Satz: Die so genannte *transitive Hülle* $R^+ := \bigcup_{n \geq 1} R^n$ ist die kleinste R umfassende transitive Relation auf X für gegebenes $R \subseteq X \times X$.

Mit $R^0 := \{\Delta_X\}$ können wir genauso festhalten:

Satz: Die so genannte *reflexive transitive Hülle* $R^* := \bigcup_{n \geq 0} R^n$ ist die kleinste R umfassende reflexive und transitive Relation auf X für gegebenes $R \subseteq X \times X$.

Die **von einem DEA $A = (Q, \Sigma, \delta, q_0, F)$ akzeptierte Sprache** kann man formal wie folgt beschreiben.

Ein Element aus $C = Q \times \Sigma^*$ heißt *Konfiguration* von A.
Definiere eine binäre Relation $\vdash_A$ auf C durch $(q, w) \vdash_A (q', w')$ gdw.
$\exists a \in \Sigma : w = aw'$ und $q' = \delta(q, a)$.
Die zweite Komponente einer Konfiguration ist die "übrige Eingabe".
$\vdash_A$ beschreibt den Konfigurationsübergang in einem Schritt.
Entsprechend beschreibt $\vdash_A^n$ n Schritte von A.
Daher können wir definieren:

$$L(A) = \{w \in \Sigma^* \mid \exists q \in F : (q_0, w) \vdash_A^* (q, \lambda)\}.$$

Exkurs: Zustandsraum allgemein

Zustandsraum / Konfigurationsraum $\equiv$ Landkarte

Konfigurationsbergänge $\equiv$ erlaubte “Einzelschritte”

Konfigurationsbergangsfolge $\equiv$ Weg auf der Karte

Ein bekanntes Beispiel

Drei Missionare und drei Kannibalen wollen einen Fluss queren.
Das verfügbare Boot befördert höchstens zwei Personen.
Nebenbedingung: Die Kannibalen dürfen auf keinem Ufer in der Mehrheit sein.

Zustandsraum: $\{(i, j) \mid 0 \leq i, j \leq 3\}$ i Miss. / j Kann. am linken Ufer
Problem: Übergang durch die Mitte

	3	2	1	0
3	Start			
2	X		X	X
1	X	X		X
0				Ziel

Was tut also “unser” Automat ?

$$L(A) = \{w \in \Sigma^* \mid (q,w) \vdash_A^* (r,\lambda)\}.$$

$$\{w \in \Sigma^* \mid (q,w) \vdash_A^* (r,\lambda)\} = \{u \in \Sigma^* \mid (q,u) \vdash_A (r,\lambda)\} \cdot \{v \in \Sigma^* \mid (r,v) \vdash_A^* (r,\lambda)\}$$

$\Rightarrow$

$$\{w \in \Sigma^* \mid (q,w) \vdash_A^* (r,\lambda)\} = \{a,b\}\{c^n \mid n \in \mathbb{N}\}$$

Dualzahlen

Beispiel: Wir wollen einen Automaten A diskutieren, der Dualzahlen der Form 1010010 bzw. *Dualbrüche* der Form $10010 * 01110$ erkennt.
(Der $*$ soll der Deutlichkeit dienen für das "Dualzeichen.")
Wir wollen zur Erleichterung auch "leere Zahlen" akzeptieren und Zahlen der Formen $xxxxx*$ und $xxxxx * 0$.
Das heißt, der Automat soll alle Wörter über dem Alphabet $\{0, 1, *\}$ akzeptieren, die keinen oder genau einen Punkt enthalten.

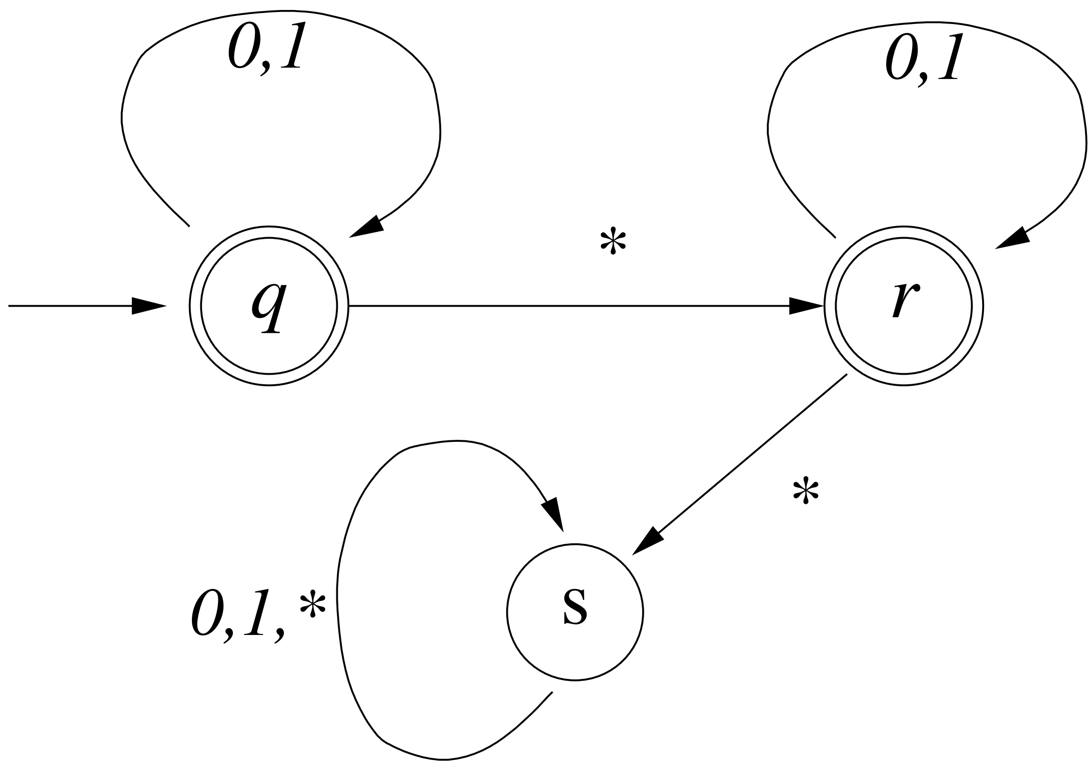
$$A = (\{q, r, s\}, \{0, 1, *\}, \delta, q, \{q, r\})$$

$$
\begin{aligned}
\delta(q,0) &= q, & \delta(q,1) &= q \\
\delta(q,*) &= r, & \delta(r,0) &= r \\
\delta(r,1) &= r, & \delta(r,*) &= s \\
\delta(s,0) &= s, & \delta(s,1) &= s \\
\delta(s,*) &= s
\end{aligned}
$$

Die Beschreibung mittels einer Überführungstafel hat das folgende Aussehen:

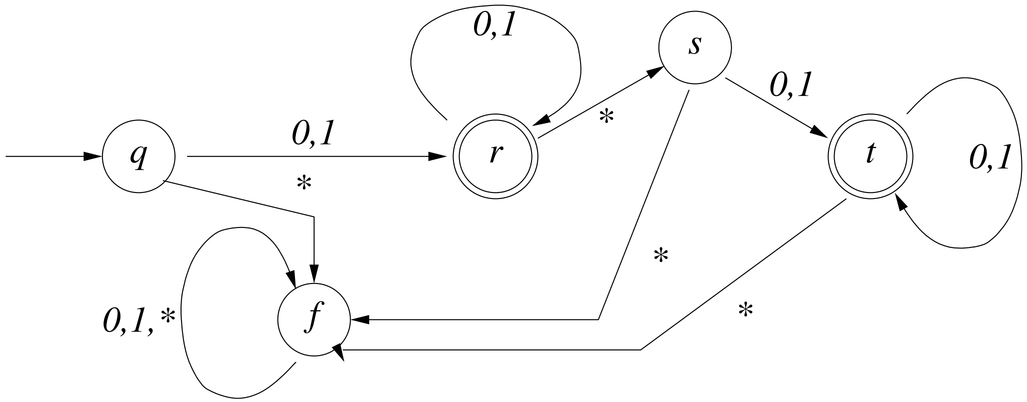
$\delta(\ ,\ )$	0	1	$*$
$\rightarrow q \rightarrow$	q	q	r
$r \rightarrow$	r	r	s
s	s	s	s

Völlig äquivalent ist die Beschreibung als gerichteter kantenbeschrifteter Graph:



Hierbei haben wir uns erlaubt, Kanten zwischen den selben Knoten zusammenzufassen zu einer Kante mit einer Liste von Eingabezeichen.

Man erkennt, dass der Automat nur dann in einen Endzustand übergeht, wenn er eine ganze oder eine ‘real’-Dualzahl liest. Wollen wir dagegen leere Zahlen und Zahlen der Formen $xxxxx*$ und $*xxxx$ nicht akzeptieren, so müssen wir den Automaten folgendermaßen modifizieren:



Wollen wir auch zulassen, dass Zahlen der Form $*xxxx$ möglich sind, oder führende Nullen vor einer Eins vor dem Dualzeichen unterdrücken, müssen wir den Automaten weiter modifizieren. Dies sei zur Übung überlassen.

Was “tut” der folgende Automat ?

δ	a	b
$\rightarrow s$	s	q
$q \rightarrow$	r	q
r	r	r

Wie sieht der Automatengraph aus ?

Wie kann man $L(A)$ beschreiben

- in Worten oder
- in Mengennotation?

Lemma: $L(A) = L := \{a^n b^m \mid n \geq 0, m \geq 1\}$
mit $a^0 := \lambda$ and $a^{n+1} = a^n \cdot a$ (*Wortpotenzen*).

Der Beweis von $L(A) = L$ hat in der Regel zwei Richtungen: (a) $L(A) \subseteq L$ und (b) $L \subseteq L(A)$.

Beweistechnik: Induktion.

- für (a) betrachtet das Induktionsargument i.d.R. n-Schritt-Konfigurationsübergänge $c \vdash^n c'$

- für (b) erfolgt das Induktionsargument hingegen über die Wortlänge $n = |w|$ mit $w \in L$ (oder aus Σ^*).

Lemma 1: $\forall n \geq 0 : (s, a^n) \vdash^* (s, \lambda)$.
Beweis: ✓ für $n = 0$. Für $n > 0$:

$$(s, a^n) = (s, a \cdot a^{n-1}) \vdash (s, a^{n-1}) \vdash^* (s, \lambda)$$

Beweis von $L(A) \supseteq L$: Offensichtlich ist $b \in L$ das einzige Wort der Länge Eins oder kürzer in L und $b \in L(A)$, denn $\delta(s, b) = q$.
Betrachte $a^n b^m \in L$ mit $n + m > 1$. Gilt $m = 1$, so folgt mit Lemma 1:

$$(s, a^n b) \vdash^* (s, b) \vdash (q, \lambda)$$

Für $m > 1$ folgt mit der IH:

$$(s, a^n b^{m-1} b) \vdash^* (q, b) \vdash (q, \lambda)\,.$$

In beiden Fällen ist $a^n b^m \in L(A)$.

Zustand r ist eine “Falle” in folgendem Sinne:

Lemma 2: $\forall v, w \in \Sigma^* : ((r, v) \vdash^* (x, w)) \Rightarrow x \in \{r\}$.

Lemma 3: $\forall w \in \Sigma^* : ((q, w) \vdash^* (q, \lambda)) \Rightarrow w \in \{b\}^*$.

Beweis durch Induktion über $|w|$. ✓ für $n = 0$.

Für $n > 0$ betrachte $w = a_1 a_2 \dots a_n$ mit $\forall 1 \leq i \leq n : a_i \in \Sigma$.

Diskutiere $(q, a_1 a_2 \dots a_n) \vdash (x, a_2 \dots a_n) \vdash^* (q, \lambda)$.

Falls $a_1 = a$, so gilt $x = r$, und wegen Lemma 2 ist $(x, a_2 \dots a_n) \vdash^* (q, \lambda)$ unmöglich.

Ist $a_1 = b$, so gilt $x = q$, und die IH liefert $a_2 = \cdots = a_n = b$.

Beweis von $L(A) \subseteq L$: Betrachte $(s, w) \vdash^n (q, \lambda)$.

Für $n = 1$, $w = b \in L$.

Für $n > 1$ erörtern wir $(s, w) \vdash (x, w') \vdash^{n-1} (q, \lambda)$.

Ist $w = aw'$, so $x = s$; die IH liefert die Behauptung.

Ist $w = bw'$, so $x = q$; dann folgt mit Lemma 3 die Behauptung.