Grundlagen Theoretischer Informatik I

SoSe 2011 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Grundlagen Theoretischer Informatik I Gesamtübersicht

- Organisatorisches; Einführung
- Logik & Beweisverfahren
- Mengenlehre
- reguläre Sprachen

Formale Sprachen—eine Annäherung

Was ist eine Sprache ?
Eine *Sprache* ist eine Menge von Wörtern.

Das sollte jetzt auch mathematisch klar sein:

Eine Menge Σ heißt *Alphabet*, falls Σ eine endliche, nicht-leere Menge ist, i.Z.: $|\Sigma| < \infty$ und $\Sigma \neq \emptyset$.

Kurz gesagt: Eine Sprache L (über Σ) ist eine Teilmenge des von der Menge Σ frei erzeugten Monoids, i.Z.: $\mathrm{L} \subseteq \Sigma^*$.

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*

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 $a_2 = \dots = a_n = b$.

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.

Operationen auf Sprachen

Erinnerung: Eine Sprache ist eine *Menge* von Wörtern.

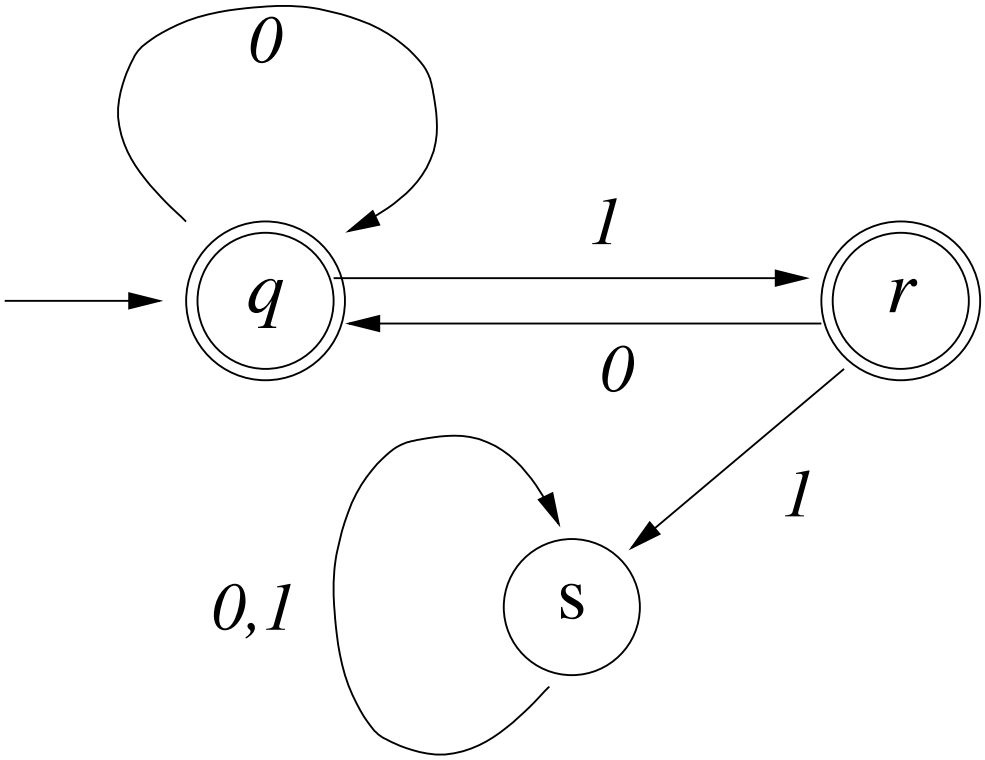
Also: Vereinigung, Durchschnitt, Komplement, ... von Sprachen liefern wieder Sprachen, sind also *Operationen auf Sprachen*.

Eine Menge von Sprachen heißt auch *Sprachfamilie*.

Wir haben bislang die Familie **DEA** der DEA-akzeptierbaren Sprachen betrachten.

Ist f eine k-stellige Operation auf Sprachen und ist $\mathcal{L}$ eine Sprachfamilie, so heißt $\mathcal{L}$ *abgeschlossen gegen* f gdw. für alle $L_1, \ldots, L_k \in \mathcal{L}$ gilt: $f(L_1, \ldots, L_k) \in \mathcal{L}$.

Ein Beispiel Was beschreibt folgender Automat ?



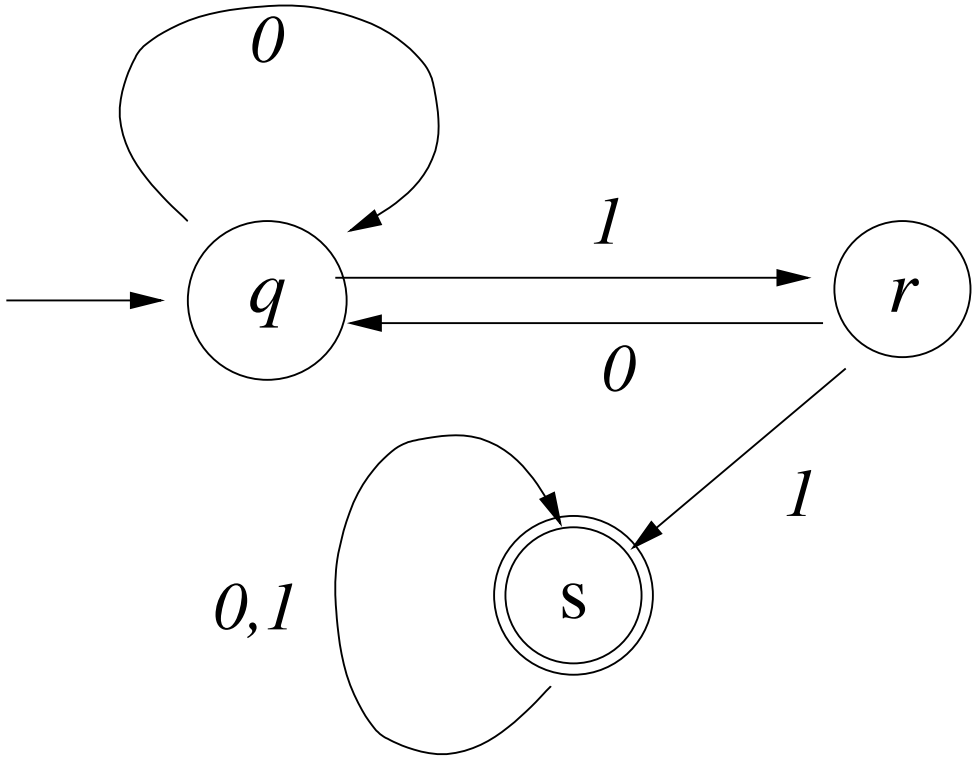
$$
\begin{aligned}
L(A) &= \{w \in \{0,1\}^* \mid w \text{ enthält nicht das Teilwort } 11\} \\
&= \{w \in \{0,1\}^* \mid \text{ Auf jedes Vorkommen von } 1 \\
&\qquad \text{vor der letzten Stelle in } w \text{ folgt } 0.\ \}
\end{aligned}
$$

Satz: **DEA**-Sprachen sind komplementabgeschlossen.

Beweis: Sprachkomplement entspricht Endzustandsmengenkomplement.

Hinweis: Dies ist streng genommen gar kein Beweis, sondern die bloß Angabe einer Konstruktion, die zu vorgelegtem DEA A einen DEA A' bastelt, für den zu zeigen wäre, dass $L(A)$ das Komplement von $L(A')$ akzeptiert. Man hat also zu zeigen: $w \in L(A) \rightarrow w \notin L(A')$ und $w \in L(A') \rightarrow w \notin L(A)$. Die formale Durchführung ist Ihnen zur Übung überlassen.

Das Beispiel



Schwieriger: Vereinigungsbildung

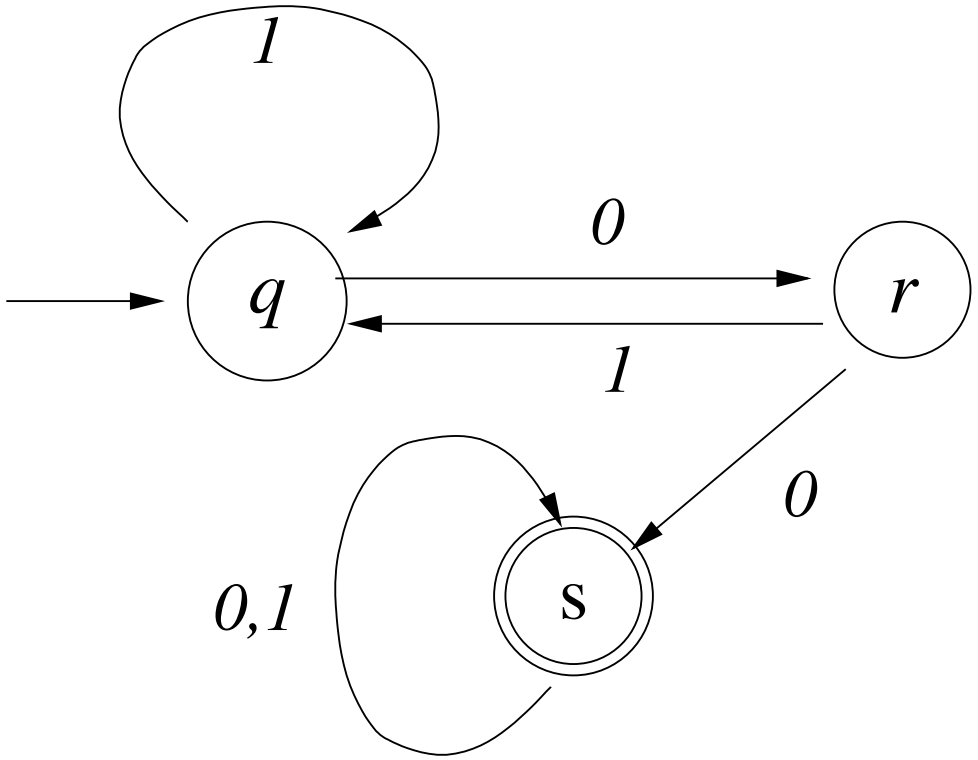
Wie kann man aus DEA-Beschreibungen für

$$L_1 = \{w \in \{0,1\}^* \mid w \text{ enthält das Teilwort } 11\},$$
$$L_2 = \{w \in \{0,1\}^* \mid w \text{ enthält das Teilwort } 00\},$$

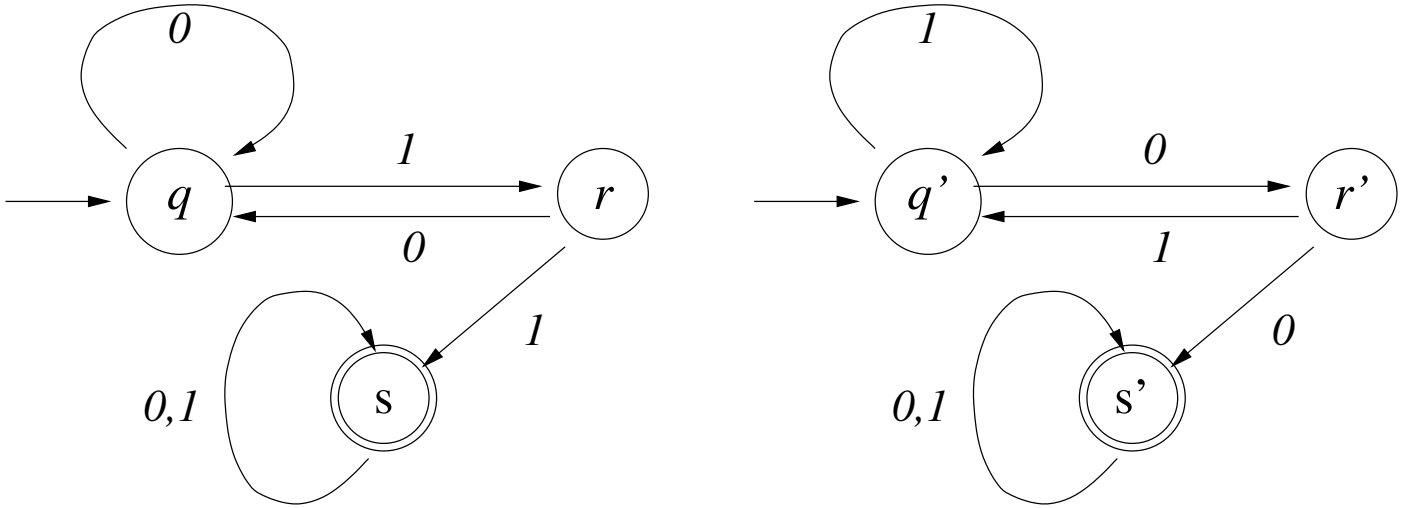
einen DEA für $L_1 \cup L_2$ basteln ?

Idee: Verwende "irgendwie" Automaten für L_1 und für L_2.
Kann man EAs aus "Teilprogrammen" zusammensetzen ?

Ein DEA für L_2 sähe wie folgt aus:



Vereinigung durch mehrere Anfangszustände ??



Ein **nichtdeterministischer endlicher Automat** oder NEA 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)

$\delta \subseteq Q \times \Sigma \times Q$: *Überführungsrelation*

$Q_0 \subseteq Q$: *Anfangszustände*

$F \subseteq Q$: *Endzustände*

Sprachfamilie: **NEA**.

Überführungstafel

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

Beispiel: Betrachte:

δ	0	1
$\rightarrow q$	q	r
$r \rightarrow$	q	s
s	s	s
$\rightarrow q'$	r'	q'
$r' \rightarrow$	s'	q'
s'	s'	s'

Die **von einem NEA $A = (Q, \Sigma, \delta, Q_0, F)$ 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 $\delta \ni (q, a, q')$.
Die zweite Komponente einer Konfiguration ist die Resteingabe.
$\vdash_A$ beschreibt einen möglichen Konfigurationsübergang in einem Schritt.
Entsprechend beschreibt $\vdash_A^n$ n Schritte von A.
Daher können wir definieren:

$$L(A) = \{w \in \Sigma^* \mid \exists q_0 \in Q_0, q \in F : (q_0, w) \vdash_A^* (q, \lambda)\}.$$

NEAs zur Spezifikation I

Beispiel: Gesucht: NEA, der in der Reihenfolge (aber nicht unbedingt nebeneindander) die Zeichen 0, 1, 1 enthält.

Lösung:

δ	0	1
$\rightarrow q$	q, r	q
r	r	r, s
s	s	s, t
$t \rightarrow$	t	t

Dazu DEA ?!

Idee: “erstes Vorkommen” zu prüfen genügt !

δ	0	1
$\rightarrow$ q	r	q
r	r	s
s	s	t
t $\rightarrow$	t	t

Hinweis: Pattern Matching !

NEAs zur Spezifikation II

Satz: Jede endliche Sprache ist **NEA**-Sprache.

Beweis: Sei $L = \{w_1, \ldots, w_M\} \subseteq \Sigma^*$ mit

$$w_i = a_{i,1} \cdots a_{i,\ell(w_i)}, \ 1 \leq i \leq M.$$

Betrachte den folgendenmaßen spezifizierten *Skelettautomaten*:

$$Q = \{(i,j) \mid 1 \leq i \leq M, 0 \leq j \leq \ell(w_i)\},$$

$$Q_0 = \{(i,0) \mid \leq i \leq M\} \quad \text{und} \quad F = \{(i,\ell(w_i)) \mid \leq i \leq M\}$$

$$\delta = \{((i,j), a, (i,j+1)) | 1 \leq i \leq M, 0 \leq j < \ell(w_i), a_{i,j+1} = a\}.$$

Hinweis: Master-Vorlesung über Lernalgorithmen

Unterschiede DEA / NEA Spezifikation

anhand der Überführungstafel; wie DEA, ABER:

- Einträge dürfen leer sein, d.h. der Automat ist *unvollständig*.
  Manchmal auch bei DEAs zugelassen. . . (partielle DEAs)

- Es gibt mehr als einen Eintrag (das heißt ja Nichtdeterminismus!).

- Es gibt eine Anfangszustandsmenge.

- Manchmal werden neben Buchstaben auch Wörter als Spaltenüberschrift zugelassen, insbesondere das leere Wort: *NEA mit λ-Übergängen*.

Vereinigung leicht gemacht

Satz: **NEA** ist unter Vereinigung abgeschlossen.

Beweis: Es seien $A_i = (Q_i, \Sigma, \delta_i, Q_{i,0}, F_i)$ für $i = 1, 2$ zwei NEAs.

Durch Umbenennen können wir $Q_1 \cap Q_2 = \emptyset$ voraussetzen.

Dann wird $L(A_1) \cup L(A_2)$ beschrieben durch:

$$(Q_1 \cup Q_2, \Sigma, \delta_1 \cup \delta_2, Q_{1,0} \cup Q_{2,0}, F_1 \cup F_2).$$

NEA=DEA $\rightsquigarrow$ Sprachfamilie **REG** der *regulären Sprachen*.

Beweis: $\supseteq$ ✓. Für $\subseteq$ betrachte NEA $A = (Q, \Sigma, \delta, Q_0, F)$.
Die Relation δ lässt sich als Abbildung

$$\delta_f : Q \times \Sigma \to 2^Q \text{ auffassen mit } (q, a) \mapsto \{r \in Q \mid (q, a, r) \in \delta\}.$$

Dies liefert auch eine Abbildung $\delta_f : 2^Q \times \Sigma \to 2^Q$, $(P, a) \mapsto \bigcup_{p \in P} \delta_f(p, a)$.
Setze $F' = \{P \in 2^Q \mid P \cap F \neq \emptyset\}$ und betrachte DEA

$$A' = (2^Q, \Sigma, \delta_f, Q_0, F').$$

Behauptung: $L(A) = L(A')$.

Hinweis: Zustandsexplosion !

Zustandsexplosion: ein böses Beispiel

$$L_k = \{x \in \{0,1\}^* \mid \ell(x) \geq k, \text{ das } k\text{-letzte Zeichen von } x \text{ ist } 0\}$$

Lemma: L_k kann durch einen NEA mit $k+1$ Zuständen erkannt werden, aber durch keinen DEA mit weniger als 2^k Zuständen.

Beweis: Der NEA ist durch folgende Tafel beschrieben:

	0	1	
$\rightarrow q_0$	q_0, q_1	q_0	
q_i	q_{i+1}	q_{i+1}	für $0 < i < k$
$q_k \rightarrow$			

Warum ist das Beispiel “böse” ?

$$L_k = \{x \in \{0,1\}^* \mid \ell(x) \geq k, \text{ das } k\text{-letzte Zeichen von } x \text{ ist 0 } \}$$

Gäbe es einen DEA A mit $< 2^k$ Zuständen für L_k, so gibt es (Schubfachprinzip) Wörter $y_1, y_2 \in \{0,1\}^k$ und einen Zustand q mit $(q_0, y_i) \vdash_A^k (q, \lambda)$ für $i = 1, 2$.
Wähle j, sodass es ein Wort $u \in \{0,1\}^{j-1}$ gibt mit $y_1 = ua_1v_1$ und $y_2 = ua_2v_2$ mit $\{a_1, a_2\} = \{0,1\}$ (längstes gemeinsames Anfangswort von y_1, y_2).
Da $|v_1| = |v_2| = k - j$, liegt genau eines der Wörter y_1u bzw. y_2u in L_k.
Andererseits gibt es genau ein q', sodass:

$$(q_0, y_iu) \vdash_A^k (q, u) \vdash_A^{j-1} (q', \lambda) \quad \text{für } i = 1, 2,$$

d.h., y_1u und y_2u werden gleichermaßen akzeptiert oder verworfen, denn A arbeitet deterministisch.

Zustandsexplosion vermeidbar ?

Manchmal ja: durch “lazy evaluation” (Bereitstellen erst wenn nötig!)

Beispiel: Betrachte den Skelettautomaten zu $L = \{a, aa, ab, abb\}$ mit Zustandsmenge $\{(1,0),(1,1);(2,0),(2,1),(2,2);(3,0),(3,1),(3,2);(4,0),(4,1),(4,2),(4,3)\}$.

Beginnen wir mit $Q_0 = \{(1,0),(2,0),(3,0),(4,0)\}$, dem Startzustand des DEA.

$\delta_f(Q_0, a) = \{(1,1),(2,1),(3,1),(4,1)\} =: Q_1$

$\delta_f(Q_0, b) = \emptyset$

$\delta_f(Q_1, a) = \{(2,2)\} =: Q_2$

$\delta_f(Q_1, b) = \{(3,2),(4,2)\} =: Q_3$

$\delta_f(Q_3, a) = \emptyset$

$\delta_f(Q_3, b) = \{(4,3)\} =: Q_4$

$\delta_f(\emptyset, x) = \delta_f(Q_2, x) = \delta_f(Q_4, x) = \emptyset$ für $x = a, b$.

Endzustände sind Q_1, Q_2, Q_3, Q_4.

Anstelle von 2^{12} hat unser DEA nur 5 Zustände, weniger als Ausgangs-NEA!

Der so aus Skelettautomaten gewonnene DEA heißt *Präfixbaumakzeptor*.

Boolesche Abschlusseigenschaften—zusammengefasst

Aus den abgeleiteten Abschlusseigenschaften von **DEA** und von **NEA** sowie dem Gesetz von de Morgan und der soeben bewiesenen Sprachfamiliengleichheit folgt:

Satz: **REG** ist abgeschlossen gegenüber Vereinigung, Durchschnitt und Komplementbildung.