Theoretische Grundlagen des Software Engineering

2: Eigenschaften von regulären Sprachen

Stephan Schulz
schulz@eprover.org

Software
Systems
Engineering

Stiftung Universität Hildesheim
2003

Alphabet

Definition: Ein **Alphabet** Σ ist eine nichtleere, endliche Menge von Symbolen.

- Die Symbole heißen auch **Buchstaben** oder **Zeichen** des Alphabets.

Beispiele:

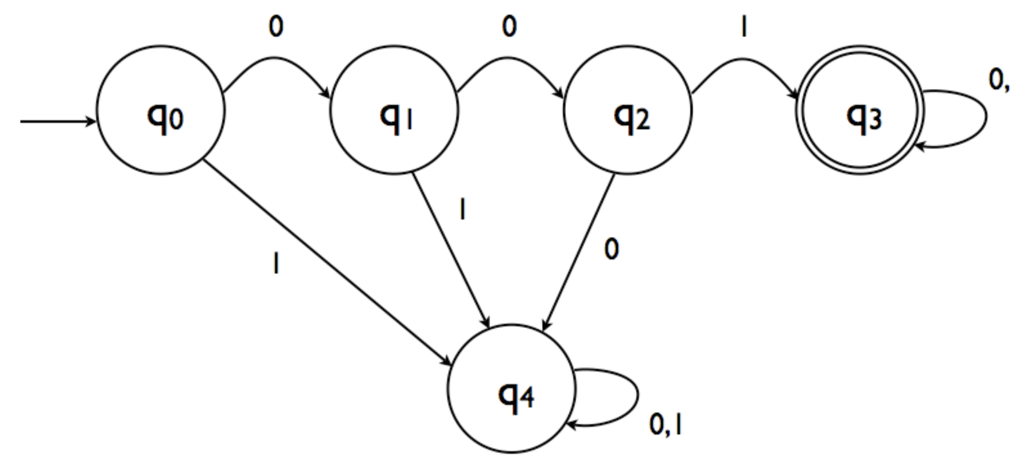
- $\Sigma=\{0,1\}$ (Binäralphabet)
- $\Sigma=\{a,...,z\}$ (Kleinbuchstaben)
- $\Sigma=\{x|x$ ist ein ASCII-Zeichen$\}$

Endliche Automaten: Formale Definition

Definition: Ein (deterministischer) endlicher Automat ist ein 5-Tupel $A=(Q, \Sigma, \delta, q_0, F)$, wobei

- Q ist eine endliche Menge (die Zustände)
- Σ ist ein Alphabet
- $\delta: Q \times \Sigma \rightarrow Q$ ist die Übergangsfunktion
- $q_0 \in Q$ ist der Startzustand
- $F \subseteq Q$ ist die Menge der akzeptierenden Zustände

Beispiel EA

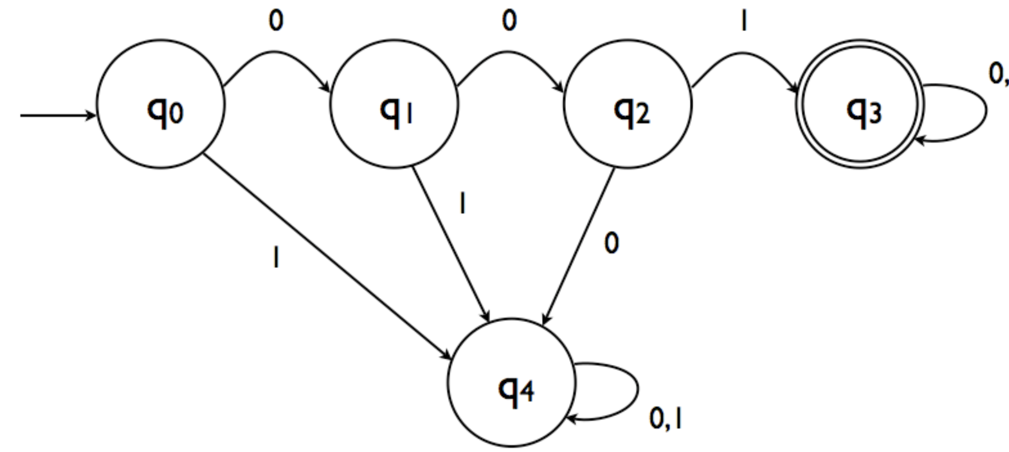


$A = (Q, \Sigma, \delta, q_0, F)$

- $Q=\{q_0, q_1, q_2, q_3, q_4\}$
- Der Startzustand ist q_0
- $\Sigma=\{0, 1\}$
- $F=\{q_3\}$

δ	0	1
q_0	q_1	q_4
q_1	q_2	q_4
q_2	q_4	q_3
q_3	q_3	q_3
q_4	q_4	q_4

Automaten als Übergangstabellen



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

- $\Sigma=\{0,1\}$
- $Q=\{q_0, q_1, q_2, q_3, q_4\}$
- Der Startzustand ist q_0
- $F=\{q_3\}$

	δ	0	1
→	q_0	q_1	q_4
	q_1	q_2	q_4
	q_2	q_4	q_3
*	q_3	q_3	q_3
	q_4	q_4	q_4

Automaten als Übergangstabellen

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

- Σ implizit aus Tabelle
- Q implizit aus Tabelle
- q_0: Markiert mit →
- F: Markiert mit *
- δ: Tabelle

Automat ist vollständig beschrieben!

	δ	0	1
→	q_0	q_1	q_4
	q_1	q_2	q_4
	q_2	q_4	q_3
*	q_3	q_3	q_3
	q_4	q_4	q_4

L(A)

Definition: Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat

- A akzeptiert ein Wort $w \in \Sigma^*$, wenn er beim Start in Zustand q_0 nach Verarbeitung des Wortes in einem Zustand $q \in F$ ist
- $L(A) = \{w \in \Sigma^* | A$ akzeptiert w$\}$ ist die Sprache von A
  - Auch: "*A* erkennt *L(A)*"

Erweiterte Übergangsfunktion

Idee:

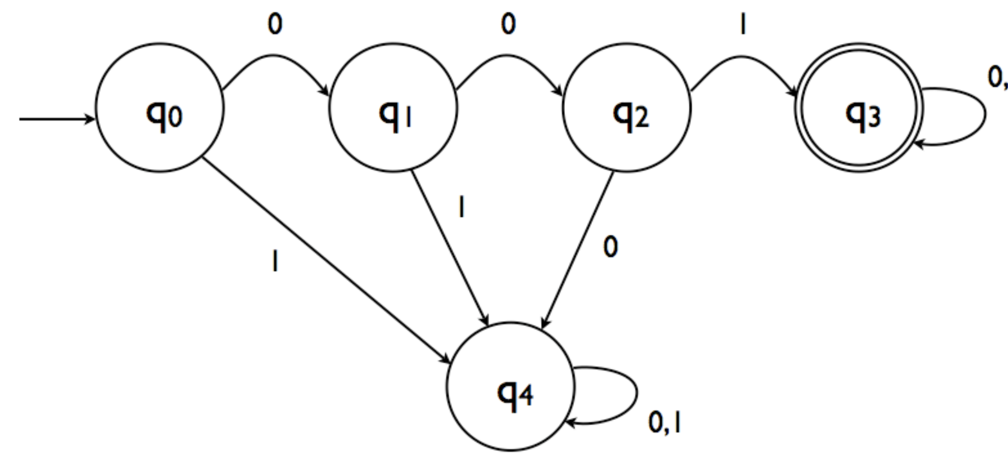
- Die Übergangsfunktion δ ist auf Tupeln $(q,a) \in Q \times \Sigma$ definiert, wobei a ein Zeichen des Alphabets ist.
- Wir wollen die Übergangsfunktion zu einer Funktion $\hat{\delta} : Q \times \Sigma^* \to Q$ auf Zuständen und Worten erweitern
- $\hat{\delta}$ berechnet zu einem Zustand q und einem Wort w den Zustand, den der Automat bei Start in q nach Abarbeitung von w hat

Erweiterte Übergangsfunktion

Definition: Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat mit Übergangsfunktion δ. Die erweiterte Übergangsfunktion $\hat{\delta}$ ist rekursiv definiert wie folgt:

- $\hat{\delta}(q,x) = q$ falls $x=\epsilon$ (Basisfall)
- Ansonsten kann x als wa mit $w \in \Sigma^*$ und $a \in \Sigma$ geschrieben werden und es gilt $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q,w),a)$

Beispiel



$$
\begin{aligned}
\hat{\delta}(q_0, 00101) &= \delta(\hat{\delta}(q_0, 0010), 1) \\
&= \delta(\delta(\hat{\delta}(q_0, 001), 0), 1) \\
&= \delta(\delta(\delta(\hat{\delta}(q_0, 00), 1), 0), 1) \\
&= \delta(\delta(\delta(\delta(\hat{\delta}(q_0, 0), 0), 1), 0), 1) \\
&= \delta(\delta(\delta(\delta(\delta(\hat{\delta}(q_0, \epsilon), 0), 0), 1), 0), 1) \\
&= \delta(\delta(\delta(\delta(\delta(q_0, 0), 0), 1), 0), 1) \\
&= \delta(\delta(\delta(\delta(q_1, 0), 1), 0), 1) \\
&= \delta(\delta(\delta(q_2, 1), 0), 1) \\
&= \delta(\delta(q_3, 0), 1)) \\
&= \delta(q_3, 1)) \\
&= q_3
\end{aligned}
$$

δ	*0*	*1*
q_0	q_1	q_4
q_1	q_2	q_4
q_2	q_4	q_3
q_3	q_3	q_3
q_4	q_4	q_4

Verträglichkeit von $\hat{\delta}$

Lemma 1: Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat. Dann gilt für alle $u,v \in \Sigma^*$, $q \in Q$: $\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q,u),v)$

Verträglichkeit von $\hat{\delta}$

Lemma 1: Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat. Dann gilt für alle $u,v \in \Sigma^*, q \in Q$: $\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q,u),v)$

Beweis: Induktion nach $|v|$

IA: $|v|=0$, also $|v|=\epsilon$.

- Dann gilt: $\hat{\delta}(q, uv) = \hat{\delta}(q, u) = \hat{\delta}(\hat{\delta}(q,u), \epsilon) = \hat{\delta}(\hat{\delta}(q,u), v)$ ✓

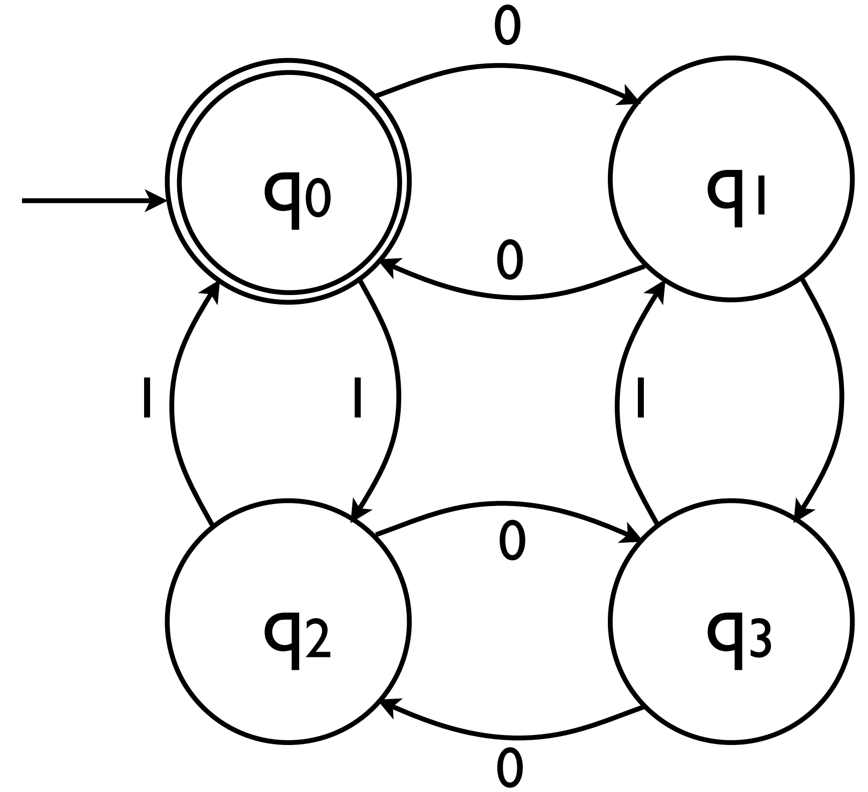
IV: Behauptung gelte für v mit $|v|=k$

IS: $k \rightarrow k+1$. Sei also o.B.d.A. $v=wa$, $w \in \Sigma^*$, $a\Sigma$.

- $\hat{\delta}(q, uv) = \hat{\delta}(q,uwa) = \delta(\hat{\delta}(q, uw), a)$

  $=_{(IV)} \delta(\hat{\delta}(\hat{\delta}(q,u),w),a) = \hat{\delta}(\hat{\delta}(q,u),wa) = \hat{\delta}(\hat{\delta}(q,u),v)$ ✓ q.e.d.

Beispiel



$\hat{\delta}(q_0, 00101101) = q_0$

$= \hat{\delta}(\hat{\delta}(0010), 1101)$

$= \hat{\delta}(q_3, 1101)$

δ	0	1
→*q0	q1	q4
q1	q2	q4
q2	q4	q3
q3	q3	q3

$\hat{\delta}(q_0, 01111100)$

$= \hat{\delta}(\hat{\delta}(q_0, 011), 11100)$

$= \hat{\delta}(q_1, 11100) = q_3$

L(A) und reguläre Sprachen

Definition: Sei $A= (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat und $\hat{\delta}$ die erweiterte Übergangsfunktion zu δ.

- A akzeptiert ein Wort $w \in \Sigma^*$, wenn $\hat{\delta}(q_0,w) \in F$
- $L(A)=\{w \in \Sigma^* \mid \hat{\delta}(q_0,w) \in F\}$ ist die Sprache von A
- Eine Sprache L heißt reguläre Sprache, wenn es einen endlichen Automaten A gibt, so dass $L=L(A)$

Grenzen

Theorem 1: Es gibt Sprachen, die nicht regulär sind.

Beweis: Allgemein später ("Pumping Lemma"). En Detail:

- Idee:
  - Endliche Automaten haben nur endlich viel Speicher
  - Manche Sprachen erfordern unbeschränktes Gedächtnis
  - Einfaches Beispiel: $\{a^n b^n | n \in \mathbf{N}\} = \{ab, aabb, aaabbb, ...\}$ ist nicht regulär

Beweis (1)

Behauptung: Sei $\Sigma=\{a,b\}$ und L= $\{a^n b^n | n \in \mathbf{N}\}$. Dann existiert kein endlicher Automat A mit $L(A)=L$.

Beweis per Widerspruch.

Annahme: Es existiert EA $A=(Q, \Sigma, \delta, q_0, F)$ mit $L(A)=L$.

- Sei $k=|Q|$
- Betrachte $w=a^{k+1}$. Es existieren $i,j \in \mathbf{N}$ mit $1 \leq i < j \leq k+1$ so dass $\hat{\delta}(q_0, a^i)=\hat{\delta}(q_0, a^j)$, denn A muss beim Verarbeiten von a^{k+1} mindestens einen Zustand mehrfach durchlaufen.
- Sei $x=a^i$, $u=a^{j-i}$, $v=a^{k+1-j}$. Dann ist $w=xuv$ und $xv=a^{k+1-(j-i)}$

Beweis (2)

- Nach Konstruktion gilt $\hat{\delta}(q_0, x)=\hat{\delta}(q_0, xu)$.
- Nach Lemma 1 gilt:
  - $\hat{\delta}(q_0, xuv)=\hat{\delta}(\hat{\delta}(q_0, xu),v)=\hat{\delta}(\hat{\delta}(q_0,x),v)=\hat{\delta}(q_0, xv)$
- Sei $\hat{\delta}(q_0, a^{k+1}b^{k+1})=q$. Nach Lemma 1 gilt weiter:

$$
\begin{aligned}
q = \hat{\delta}(q_0, a^{k+1}b^{k+1}) &= \hat{\delta}(\hat{\delta}(q_0, a^{k+1}), b^{k+1}) \\
&= \hat{\delta}(\hat{\delta}(q_0, xuv), b^{k+1}) \\
&= \hat{\delta}(\hat{\delta}(q_0, xv), b^{k+1}) \\
&= \hat{\delta}(q_0, xvb^{k+1}) \\
&= \hat{\delta}(q_0, a^{k+1-(j-i)}b^{k+1})
\end{aligned}
$$

Beweis (3)

Fallunterscheidung

- Fall 1: $q \in F$
  - Dann $a^{k+l-(j-i)} b^{k+l} \in L(A)$, also $L(A) \neq \{a^n b^n | n \in \mathbf{N}\}$
    - Widerspruch zur Annahme!
- Fall 2: $q \notin F$
  - Dann $a^{k+l} b^{k+l} \notin L(A)$, also $L(A) \neq \{a^n b^n | n \in \mathbf{N}\}$
    - Widerspruch zur Annahme!

Beweis (4) (für ganz genaue)

Fall 1 und Fall 2 decken alle Möglichkeiten ab. Die Annahme es existiert EA $A=(Q, \Sigma, \delta, q_0, F)$ mit $L(A)=L$ ist also falsch.

Ergo: Es existiert kein endlicher Automat mit $L(A)=L$

$\Rightarrow$ $L(A)$ ist keine reguläre Sprache

q.e.d.

Operationen auf Sprachen

Definition:

- Sei A eine Menge, $B \subseteq A$. Das Komplement von B (in Bezug auf A) ist die Menge $\overline{B} = B \setminus A$
- Seien $A, B \subseteq \Sigma^*$ Mengen von Worten über einem Alphabet Σ.
  - $A \circ B = \{ab \mid a \in A, b \in B\}$ ist die Konkatenation von A und B
  - $A^* = \{x_1 x_2 \ldots x_n \mid n \in \mathbf{N}_0\}$ ist der (Kleene-)Abschluss von A, der *-Operator heisst auch der "Kleene-Star".

Anmerkung

Der *-Operator hat auf Alphabeten und Sprachen ähnliche, aber verschiedene Wirkung.

- Σ^* ist die Menge der Worte, die aus Buchstaben von Σ gebildet werden kann.
- L^* ist die Menge der Worte, die durch Konkatenation von Worten aus L gebildet werden kann.

Stephen Cole Kleene (1909-1994), Schüler von Alonzo Church, wichtige Beträge zu Sprachen (reguläre Ausdrücke!) und Berechenbarkeitstheorie.

Beispiele

Sei $\Sigma=\{h,o,i,e\}$ und $L_1=\{ho, hi\}$, $L_2=\{he, di\}$

$\Sigma^* = \{\epsilon, h, o, i, e, hh, ho, hi, he, oh, oo, oi, oe, ih, ih, ii, ie, eh, eo, ei, ee, hhh, hoh, hih, heh, \ldots\}$

$L_1 \cup L_2 = \{ho, hi, he, di\}$

$L_1 \circ L_2 = \{hohe, hode, hihe, hidi\}$

$L_2 \circ L_1 = \{heho, hehi, diho, dihi\}$

$L_1^* = \{\epsilon, ho, hi, hohi, hiho, hoho, hiho, \ldots\}$

$L_1^* \cup L_2^* = \{\epsilon, ho, hi, he, di, hoho, hohi, hihi, hiho, hehe, hedi, didi, dihe, \ldots\}$

$(L_1 \cup L_2)^* =$ *Minnie the Moocher*

Beispiele (2)

Sei $\Sigma=\{0,1\}$, $L_1=\{0\}^*$, $L_2=\{1\}^*$

$$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, ...\}$$

$$L_1 \cup L_2 = \{\epsilon, 0, 1, 00, 11, 000, 111, 0000, 1111, ...\}$$

$$L_1 \circ L_2 = \{\epsilon, 0, 1, 00, 11, 001, 011, 000, 111, ...\} = \{0^i 1^j\}$$

$$\overline{L_1} = \{w \in \Sigma^* \mid w \text{ enthält mindestens eine } 1\}$$

$$L_1^* = L_1$$

$$L_1^* \cup L_2^* = L_1 \cup L_2$$

$$(L_1 \cup L_2)^* = \Sigma^*$$

Abgeschlossenheit

Theorem 2: Seien $L_1, L_2 \in \Sigma^*$ reguläre Sprachen.

(a) $\overline{L_1}$ ist eine reguläre Sprache.

(b) $L_1 \cup L_2$ ist eine reguläre Sprache.

(c) $L_1 \cap L_2$ ist eine reguläre Sprache.

(d) $L_1 \circ L_2$ ist eine reguläre Sprache.

(e) L_1^* ist eine reguläre Sprache.

(f) Jede endliche Sprache ist regulär

Beweis (Theorem 2a)

Behauptung: Sei L_1 eine reguläre Sprache. Dann ist $\overline{L_1}$ eine reguläre Sprache.

Beweis: L_1 ist reguläre Sprache.

- Dann existiert EA $A=(Q, \Sigma, \delta, q_0, F)$ mit $L(A)=L_1$
- Betrachte $A' = (Q, \Sigma, \delta, q_0, F')$ mit $F'=Q \setminus F (=\overline{F})$
- $L(A') = \overline{L_1}$, denn:
  - Fall 1: $w \in L(A) \Rightarrow \hat{\delta}(q_0,w) \in F \Rightarrow \hat{\delta}(q_0,w) \notin F' \Rightarrow w \notin L(A')$
  - Fall 2: $w \notin L(A) \Rightarrow \hat{\delta}(q_0,w) \notin F \Rightarrow \hat{\delta}(q_0,w) \in F' \Rightarrow w \in L(A')$
- Also: Es existiert EA A' mit $L(A')=\overline{L_1}$, also ist $\overline{L_1}$ regulär

Beweis (Theorem 2b)

Behauptung: Seien $L_1, L_2 \in \Sigma^*$ reguläre Sprachen. Dann gilt:

$L_1 \cup L_2$ ist eine reguläre Sprache.

Idee?

Beispiel

Definition: Sei $w \in \Sigma^*$, $a \in \Sigma$. $|w|_a$ bezeichnet die Anzahl von Vorkommen von a in w.

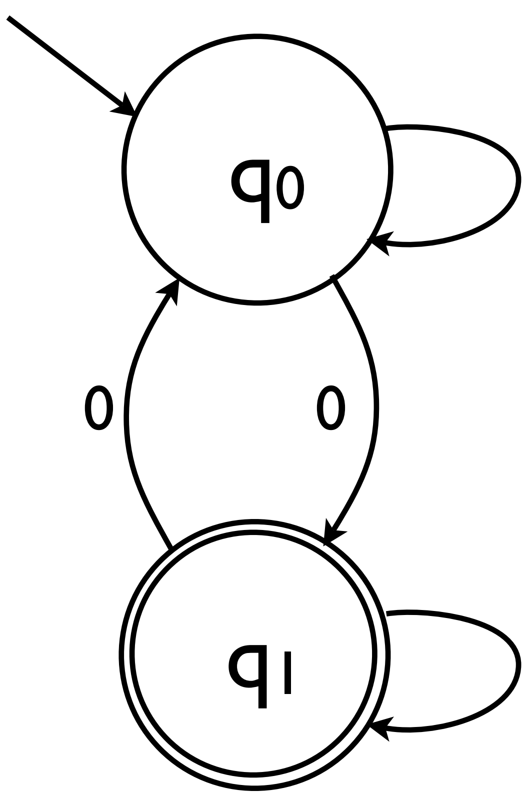
Beispiel: Sei $\Sigma = \{0, 1\}$

- $L_1 = \{w \mid |w|_0 \bmod 2 = 1\}$, $L_2 = \{w \mid |w|_1 \bmod 3 = 0\}$
  - L_1 ist die Menge aller Worte mit einer ungeraden Zahl von Vorkommen von 0. Z.B. $\{0, 110, 00101\} \subseteq L_1$
  - L_2 ist die Menge aller Worte, bei denen die Anzahl der Vorkommen von 1 durch 3 teilbar ist.
    Z.B. $\{\epsilon, 111, 10101\} \subseteq L_2$

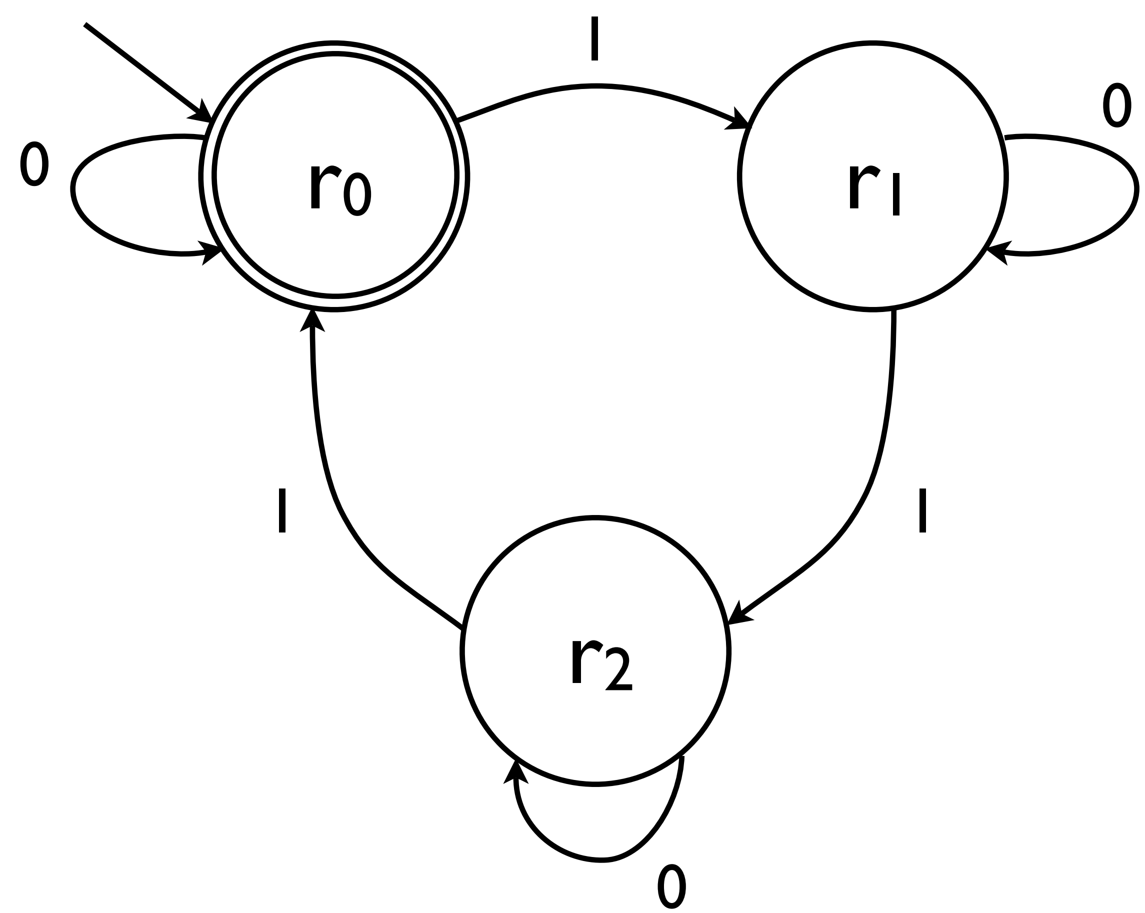
Beispiel (fortgesetzt)

$L_1=\{w \mid |w|_0 \text{ mod } 2=1\}$, $L_2=\{w \mid |w|_1 \text{ mod } 3=0\}$ sind regulär.

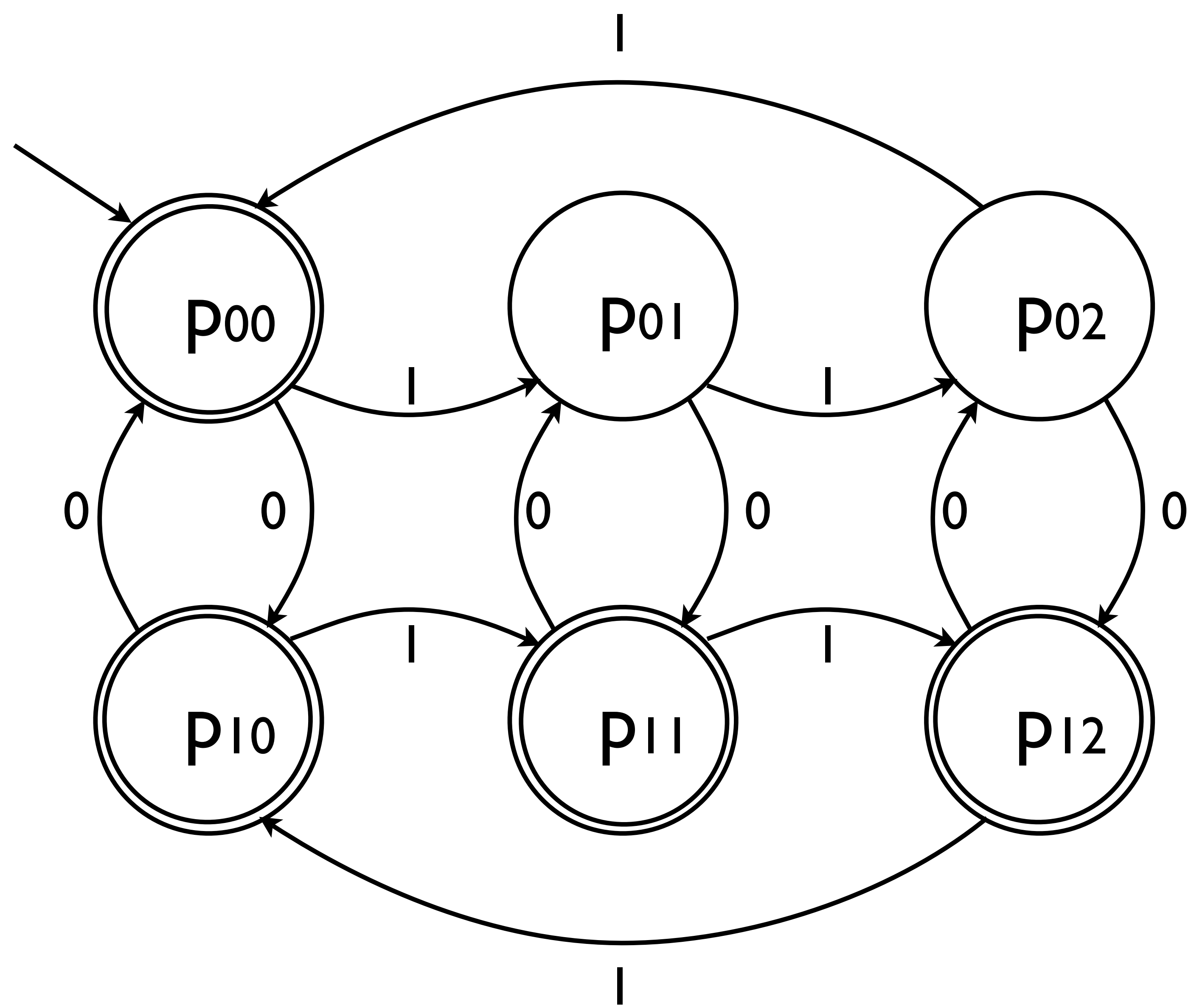
L_1



L_2



Beispiel - EA für $L_1 \cup L_2$?



Beweis (Theorem 2b)

Behauptung: Seien $L_1, L_2 \in \Sigma^*$ reguläre Sprachen. Dann gilt:

$L_1 \cup L_2$ ist eine reguläre Sprache.

Beweisidee:

- Per Definition existieren endliche Automaten A_1 und A_2 mit $L_1=L(A_1)$ und $L_2=L(A_2)$. Wir konstruieren einen endlichen Automaten A aus A_1 und A_2 so, dass $L(A)= L_1 \cup L_2$.
- Zustände von A sind Paare von Zuständen aus A_1 bzw A_2.
- Akzeptierende Zustände sind Zustände, in denen ein Element des Paares in seinem Automaten akzeptiert.

Beweis (Theorem 2b, fortgesetzt)

Also:

- L_1 ist regulär $\Rightarrow$ $\exists A_1=(Q_1, \Sigma, \delta_1, q_1, F_1)$ mit $L_1=L(A_1)$
- L_2 ist regulär $\Rightarrow$ $\exists A_2=(Q_2, \Sigma, \delta_2, q_2, F_2)$ mit $L_2=L(A_2)$
- Konstruiere $A=(Q, \Sigma, \delta, q_0, F)$ wie folgt....

Beweis (Theorem 2b, fortgesetzt)

- Konstruiere $A=(Q, \Sigma, \delta, q_0, F)$ wie folgt:
  - $Q=Q_1 \times Q_2$
  - Σ bleibt unverändert
  - $q_0=(q_1, q_2)$
  - $F=\{(r,s) | r \in F_1$ oder $s \in F_2\}$
  - $\delta((r,s),a)=(\delta_1(r,a),\delta_2(s,a))$ für alle $r \in Q_1$, $s \in Q_2$, $a \in \Sigma$
- Dann gilt: $L(A) = L_1 \cup L_2$. Formal: Induktion!

Beachte: $|Q| = |Q_1||Q_2|$ - Also: A kann deutlich größer werden.

Beweis (Theorem 2c)

Behauptung: Seien L_1, $L_2 \in \Sigma^*$ reguläre Sprachen. Dann gilt:

$L_1 \cap L_2$ ist eine reguläre Sprache.

Übung!

Beweis (Theorem 2d+2e)

Behauptung: Seien $L_1, L_2 \in \Sigma^*$ reguläre Sprachen.

(d) $L_1 \circ L_2$ ist eine reguläre Sprache.

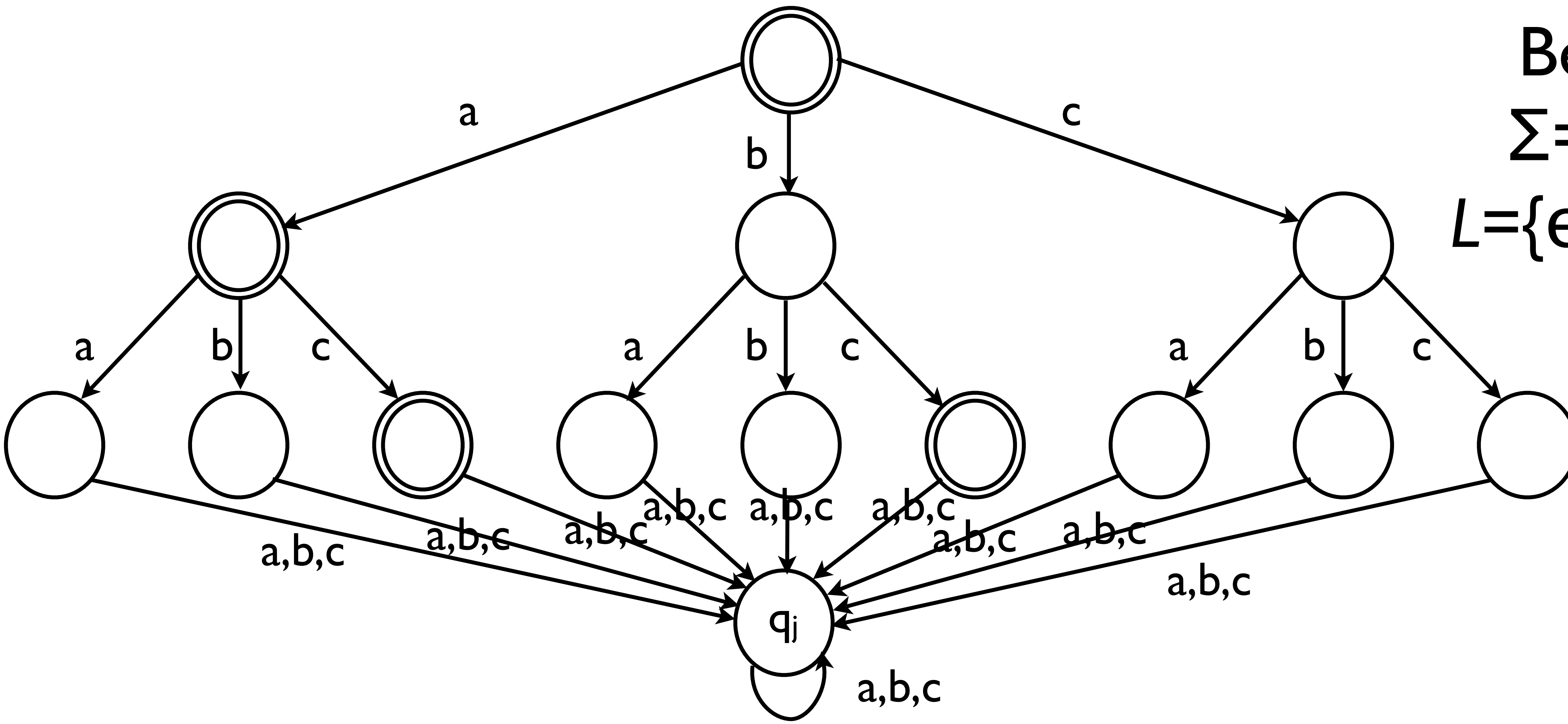
(e) L_1^* ist eine reguläre Sprache

Machen wir mit Nichtdeterminismus später

Beweisidee (Theorem 2f)

Behauptung: Jede endliche Sprache *L* ist regulär.

Idee: Konstruiere endlichen Automaten als geeigneten Trie mit Tiefe = Länge des längsten Wortes in L+1



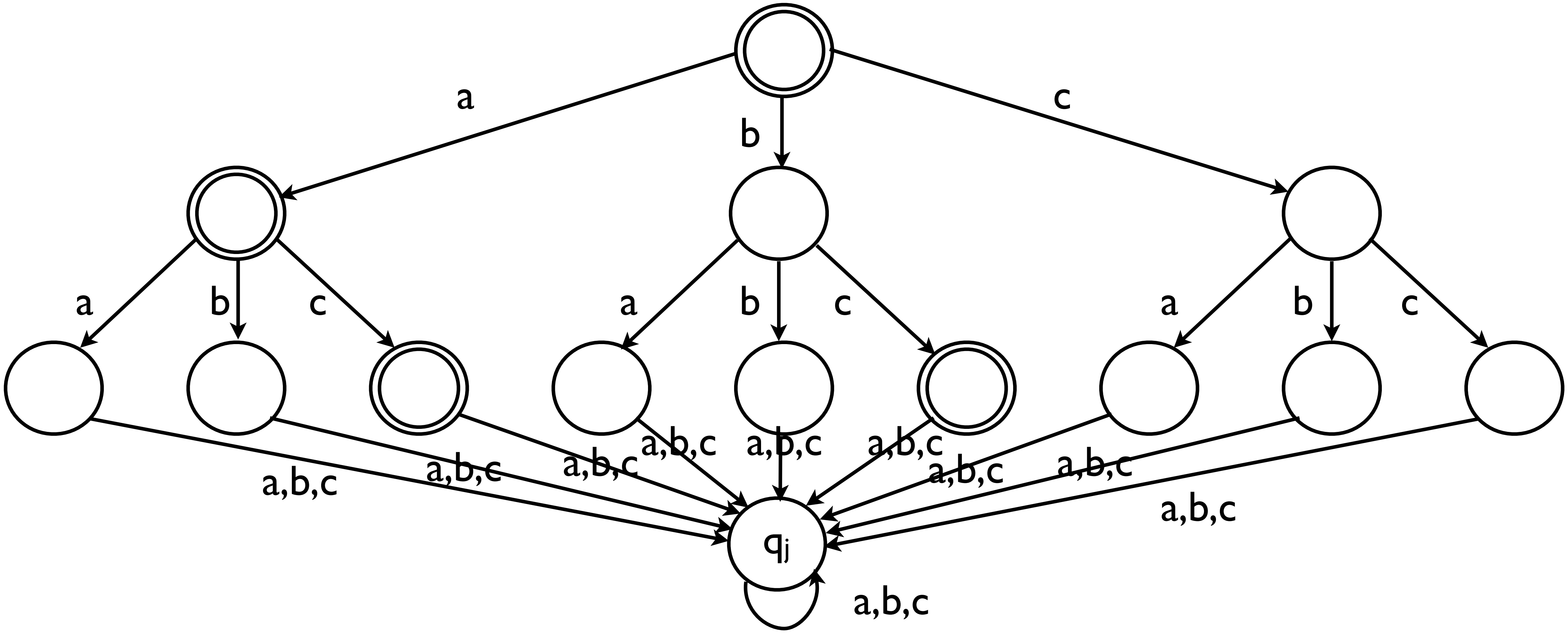
Beispiel:
$\Sigma=\{a,b,c\}$
$L=\{\epsilon, a,ac,bc\}$

Konvention

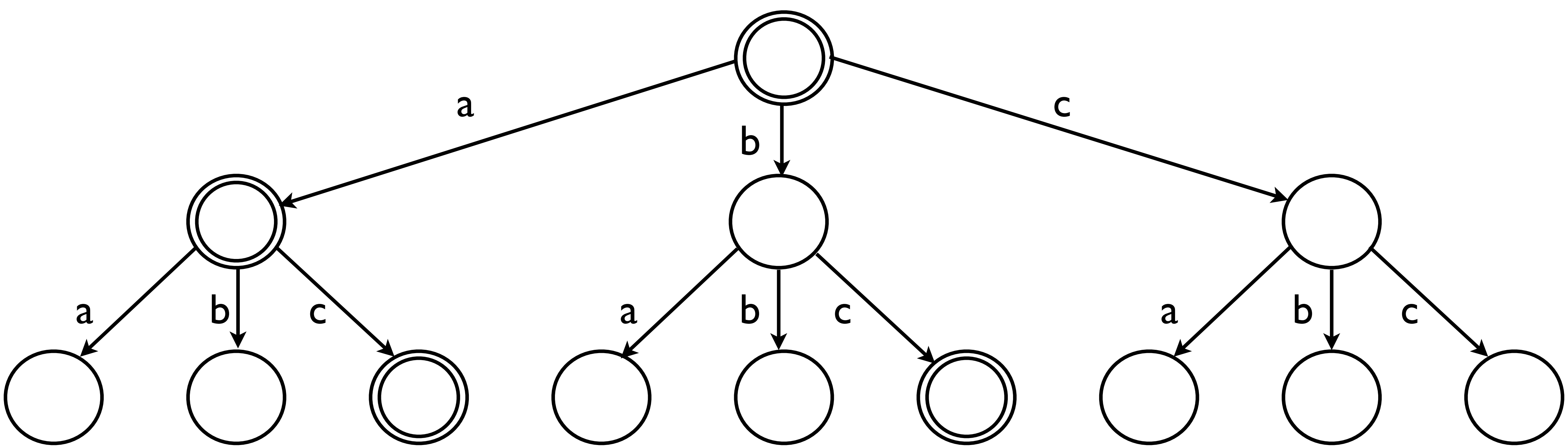
Der Automat aus dem letzten Beispiel hat einen “junk”-Zustand q_j mit den Eigenschaften

- $q_j \notin F$
- $\delta(q_j, a) = q_j$ für alle $a \in \Sigma$

Wenn in der Spezifikation eines (deterministischen) Automaten ein Übergang $\delta(q, a)$ fehlt, dann ergänzen wir automatisch einen solchen Junk-Zustand q_j und wählen $\delta(q, a)=q_j$.


a
b
c
a
b
c
a
b
c
a
b
c
a,b,c
a,b,c
a,b,c
a,b,c
a,b,c
a,b,c
a,b,c
a,b,c
a,b,c
q_j
a,b,c

=


a
b
c
a
b
c
a
b
c
a
b
c

Ende

Aufgaben

- Zeigen Sie: Seien $A,B \subseteq M$ Mengen. Dann gilt: $A \setminus B = A \cap \overline{B}$, wobei $\overline{B}$ das Komplement von B in Bezug auf M ist.
- Sei $\Sigma=\{a,b,c\}$. Sei $L_1=\{w \mid |w|_a \bmod 2=0\}$, $L_2= \{ubcbv \mid u,v\in\Sigma^*\}$
  - Konstruieren sie endliche Automaten für L_1, L_2, $L_1 \cap L_2$.
- Zeigen Sie Theorem 2c: Seien $L_1, L_2 \in \Sigma^*$ reguläre Sprachen. Dann gilt: $L_1 \cap L_2$ ist eine reguläre Sprache.