Theoretische Grundlagen des Software Engineering

4: Nichtdeterminismus Teil 2

Stephan Schulz
schulz@eprover.org

Software
Systems
Engineering

Stiftung Universität Hildesheim
2003

Nichtdeterministische endliche Automaten

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

- Q ist eine endliche Menge (die Zustände)
- Σ ist ein Alphabet
- $\delta: Q \times \Sigma \rightarrow 2^Q$ ist die Übergangsfunktion
- $q_0 \in Q$ ist der Startzustand
- $F \subseteq Q$ ist die Menge der akzeptierenden Zustände

Diskussion

Definitionen von DEA und NEA unterscheiden sich nur in der Definition der Übergangsfunktion δ

- DEA bildet Zustände in Zustände ab
- NEA bildet Zustände auf Menge von Zuständen ab
  - Beachte: Menge kann leer sein!
- Erfordert entsprechende Änderungen bei erweiterter Übergangsfunktion und Sprache des Automaten

Beispiel: DEA-Konstruktion

$N:\delta_N$	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{\}$	$\{q_2\}$
$*q_2$	$\{\}$	$\{\}$

$D:\delta_D$	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

$$\delta_D(q, a) = \bigcup_{r \in q} \delta_N(r, a)$$

Anwendung: Suche nach Schlüsselwörtern

Problem: Durchsuchen von Datenströmen nach einer Menge von Schlüsselworten

- RSS feeds
- Google News Alerts
- Ebay Abgebote
- Gen-Sequenzen/Proteinkodierungen

Naive Lösung: Speichern und sequentielle Suche

- Ineffizient in CPU und Speicher

Suche nach Schlüsselwörtern (2)

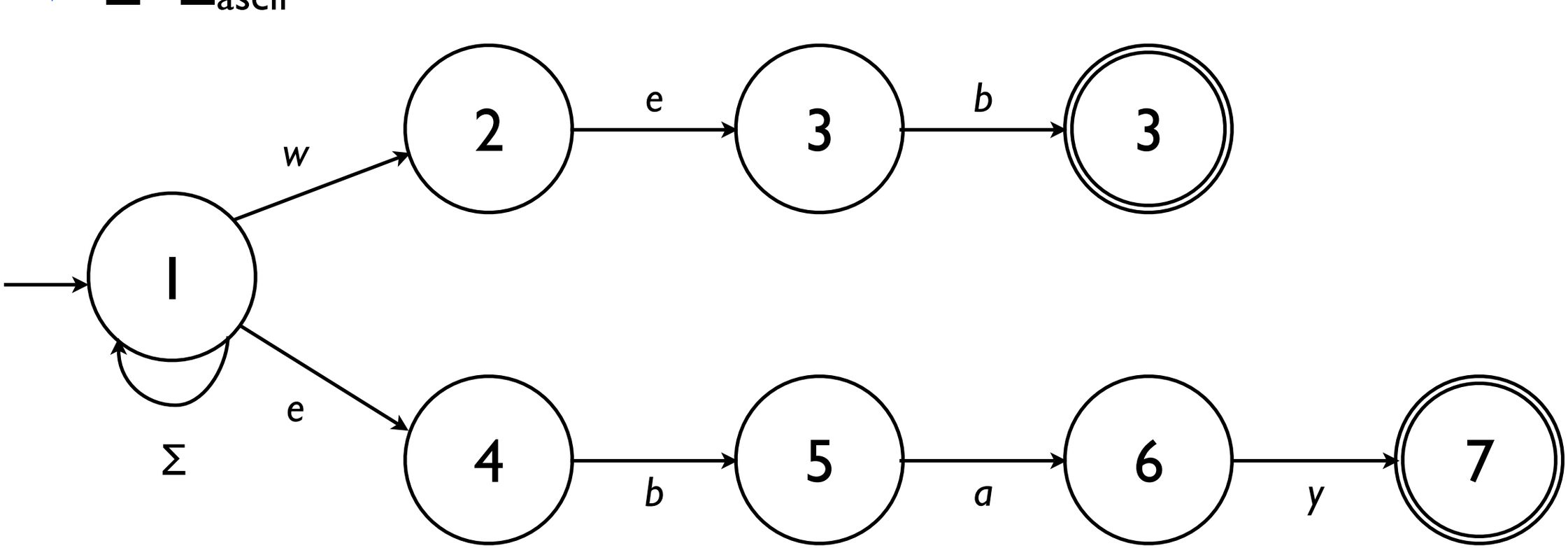
Besser:

- Spezifiziere geeigneten NEA
- Konvertiere NEA in DEA
- Implementiere effizienten DEA-Interpreter
- Terminiere mit Ergebnis, wenn jemals Endzustand erreicht wird

Suche nach Schlüsselwörtern (3)

Beispiel:

- Suche nach {*ebay*, *web*}
- $\Sigma=\Sigma_{ascii}$



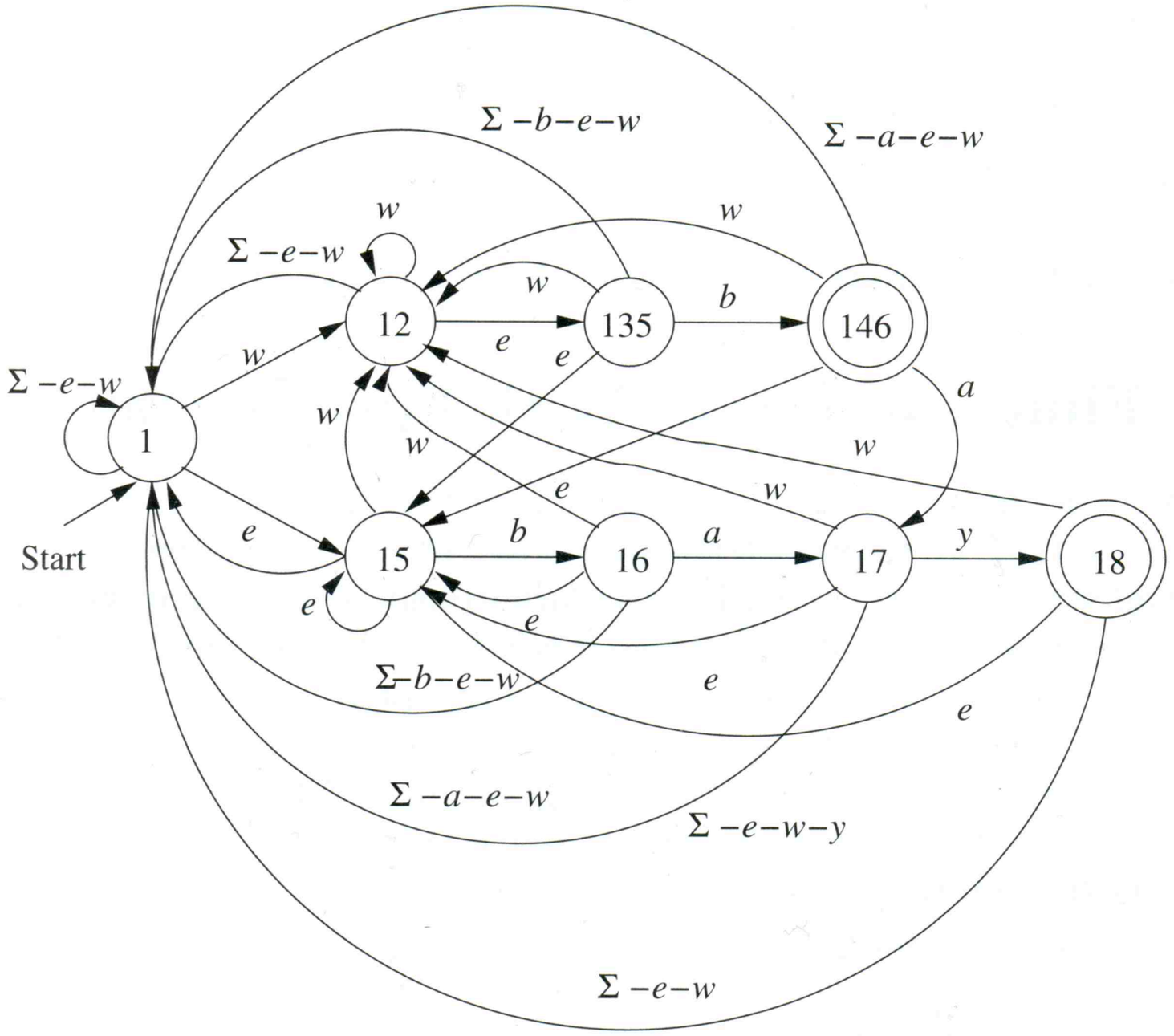
Suche nach Schlüsselwörtern (4)

Fakt:

- NEAs mit der Konstruktion wie im Beispiel können immer in DEAs mit höchstens der selben Anzahl von Zuständen umgewandelt werden.
- Diese Umwandlung kann effizient durchgeführt werden (max $O(|Q|^2)$, ~Anzahl der Buchstaben in allen Schlüsselworten)
- Deterministische endliche Automaten können effizient evaluiert werden (Übergangstabelle als Arrays: Konstante Zeit pro Zeichen)

Suche nach Schlüsselwörtern (5)

Ergebnis der DEA-Konstruktion (Hopcroft, Motwani, Ullman):

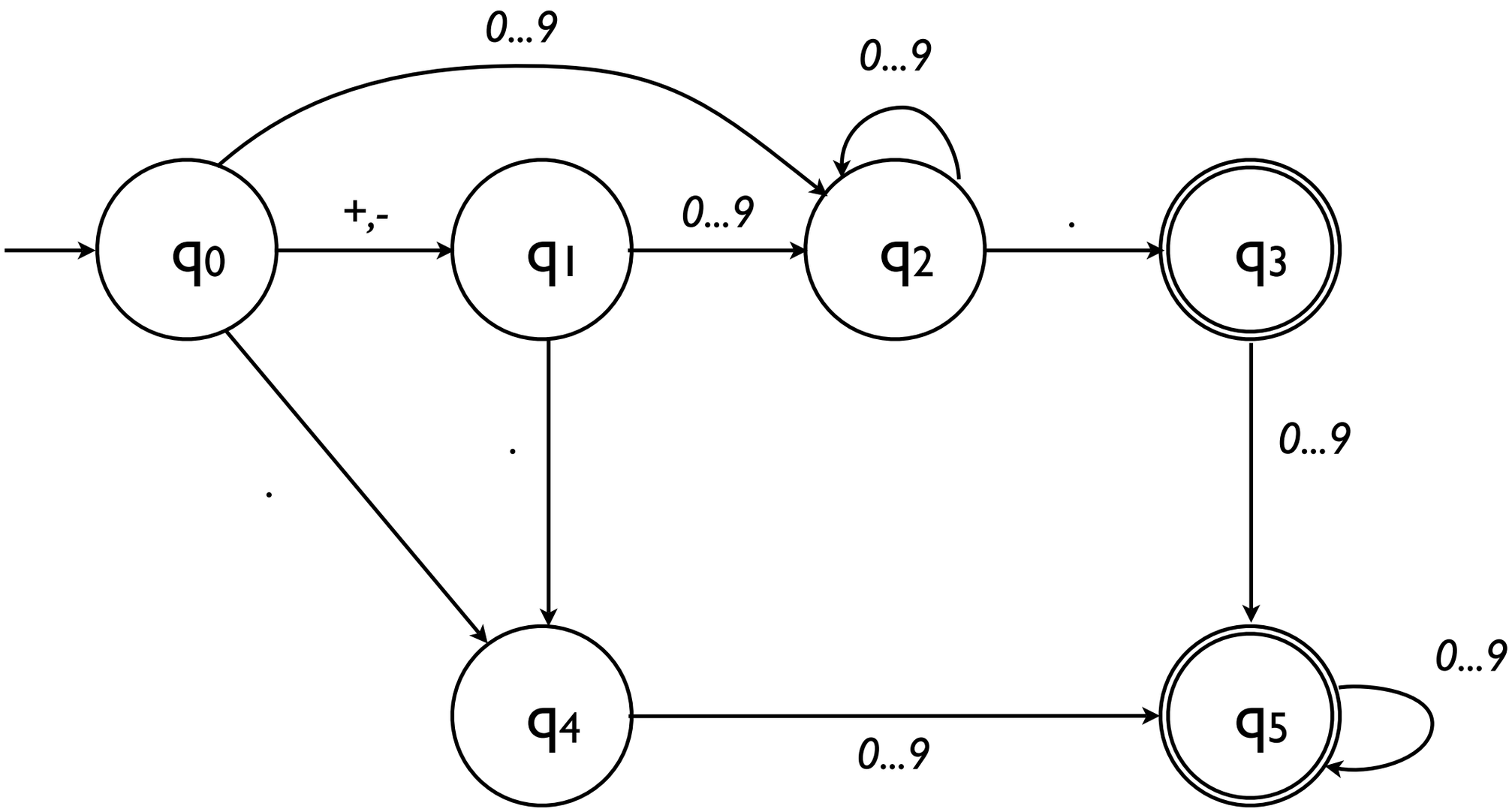


Anwendung: Erkenne Fließpunktzahlen

Fließpunkt-Zahlen:

- Optionales Vorzeichen (+/-)
- Sequenz von Ziffern
- Dezimalpunkt
- Sequenz von Ziffern
- Höchstens eine Ziffern-Sequenz darf leer sein
  - Korrekt: 1., 0.1, +.1, -3.1415, +123.
  - Falsch: ++13.1, 7, +-.3, +., 1231.89.8, .

Fließpunkt-Automat

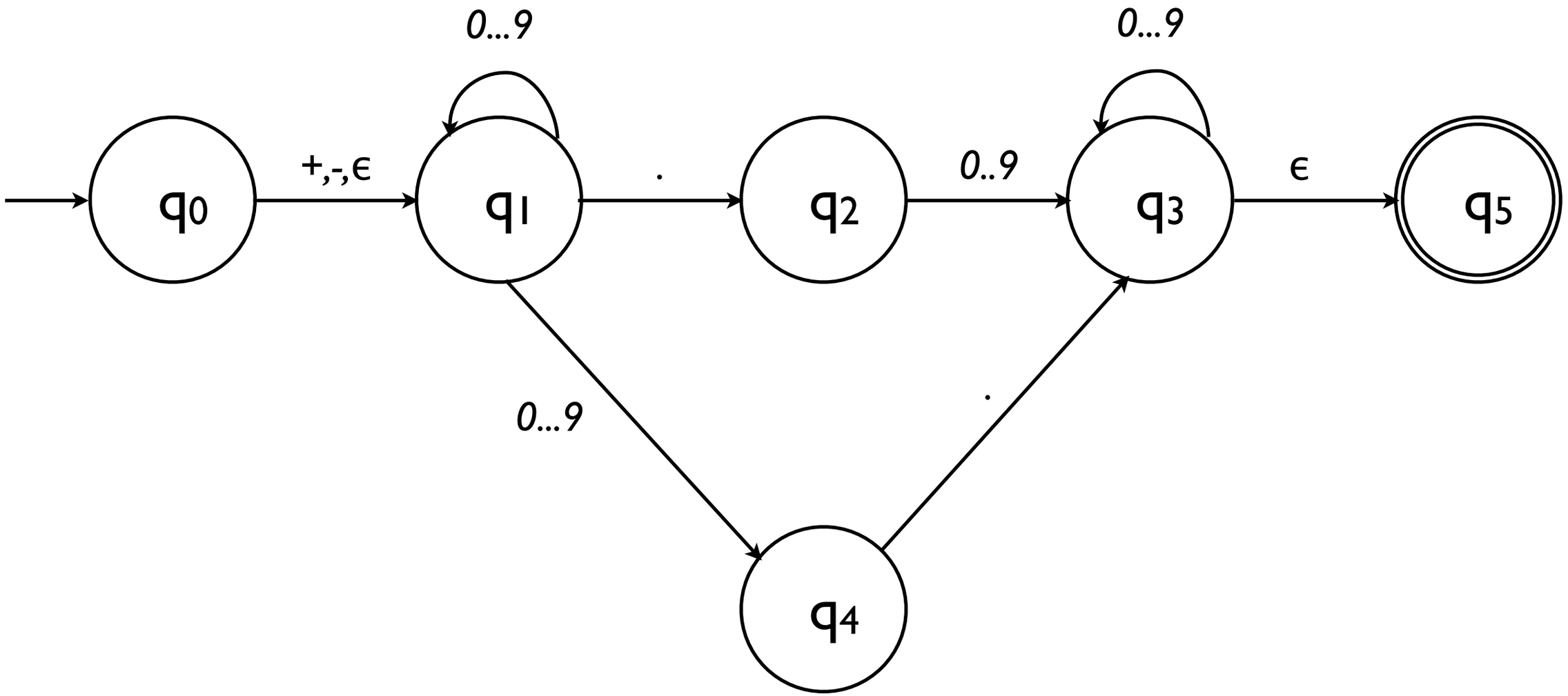


ϵ-NEAs

Idee: Erlaube Übergänge ohne ein Zeichen zu konsumieren

- Sogenannte ϵ-Übergänge
- Überspringen von optionalen Teilen
  - Auf ϵ überspringen von Teilautomaten
- Alternativen vollständig separiert
  - Gehe von Start auf ϵ zu Subautomat
- Eindeutiger akzeptierender Zustand möglich
  - Alte Endzustände $\rightarrow_\epsilon$ eindeutiger neuer Zustand

Fließkomma-Erkennung mit ϵ-NEA



ϵ-NEAs Formal

Definition: Ein nichtdeterministischer endlicher Automat mit ϵ-Übergängen (ϵ-NEA) ist ein 5-Tupel $E=(Q, \Sigma, \delta, q_0, F)$, wobei

- Q ist eine endliche Menge (die Zustände)
- Σ ist ein Alphabet
- $\delta: Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$ ist die Übergangsfunktion
- $q_0 \in Q$ ist der Startzustand
- $F \subseteq Q$ ist die Menge der akzeptierenden Zustände

Beispiel: ϵ-NEA als Tabelle

	ϵ	+/-	.	0...9
→q_0	$\{q_1\}$	$\{q_1\}$	∅	∅
q_1	∅	∅	$\{q_2\}$	$\{q_1, q_4\}$
q_2	∅	∅	∅	$\{q_3\}$
q_3	$\{q_5\}$	∅	∅	$\{q_3\}$
q_4	∅	∅	$\{q_3\}$	∅
*q_5	∅	∅	∅	∅

ϵ-Abschluss

Definition: Sei $E=(Q, \Sigma, \delta, q_0, F)$. Der ϵ-Abschluss eines Zustandes q ist rekursiv definiert wie folgt:

- $q \in ECLOSE(q)$
- Wenn $p \in ECLOSE(q)$, dann ist $\delta(p, \epsilon) \subseteq ECLOSE(q)$

Informell: $ECLOSE(q)$ ist die Menge aller Zustände, die von q mit beliebig vielen ϵ-Übergängen erreicht werden kann

- Der ϵ-Abschluss einer Menge von Zuständen $P \subseteq Q$ ist:

$$\mathrm{ECLOSE}(P) = \bigcup_{q \in P} \mathrm{ECLOSE}(q)$$

Beispiel ϵ-Abschluß

$ECLOSE(q_0)=\{q_0,q_1\}$

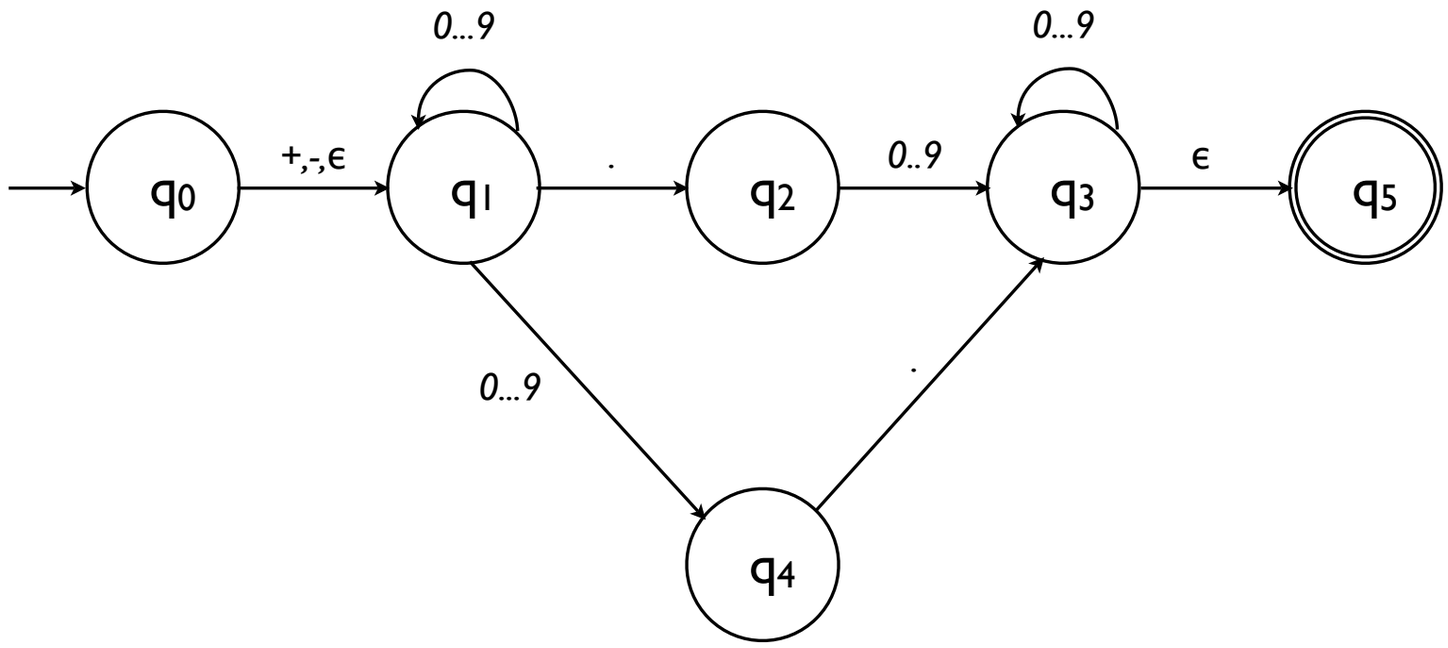
$ECLOSE(q_1)=\{q_1\}$

$ECLOSE(q_2)=\{q_2\}$

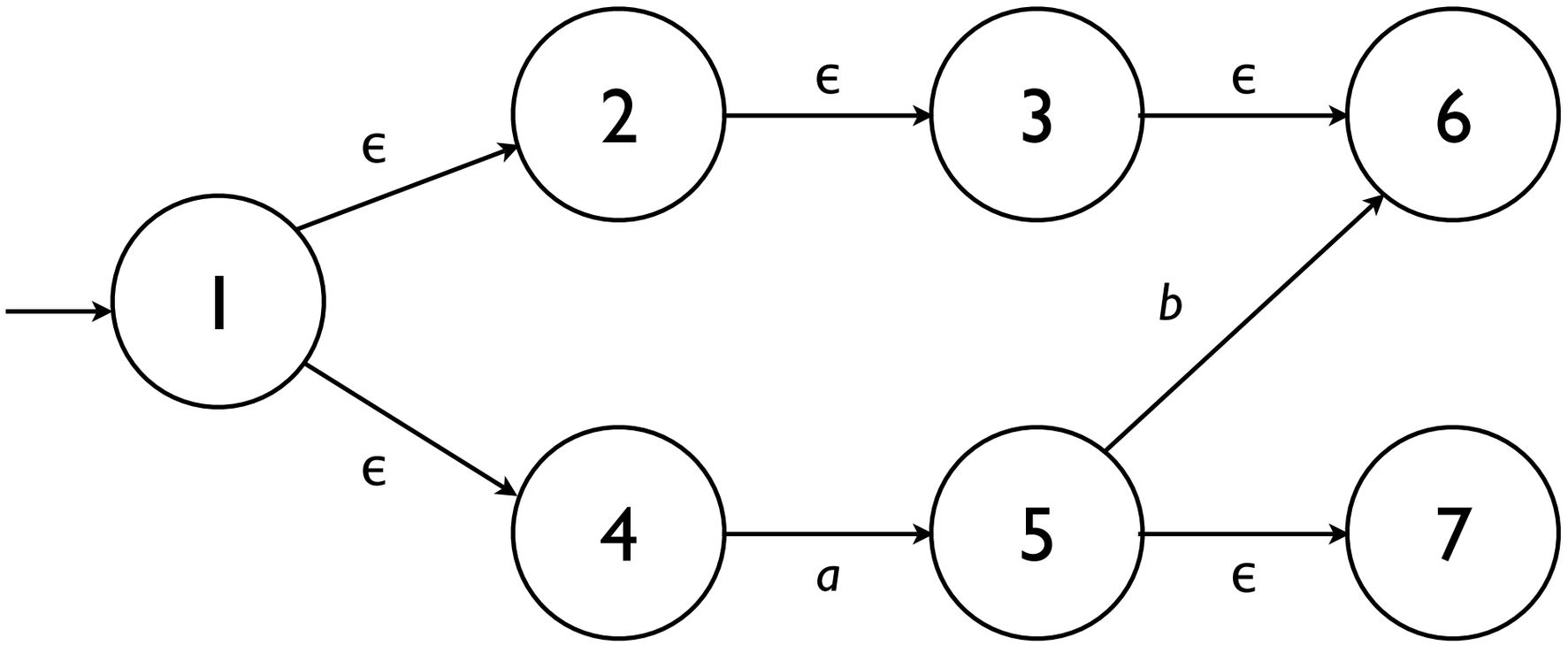
$ECLOSE(q_3)=\{q_3, q_5\}$

$ECLOSE(q_4)=\{q_4\}$

$ECLOSE(q_5)=\{q_5\}$



Komplexeres Beispiel



ECLOSE(1)= {1,2,3,4,6}

ECLOSE(2)= {2,3,6}

ECLOSE(3)= {3,6}

ECLOSE(4)= {4}

ECLOSE(5)= {5,7}

ECLOSE(6)= {6}

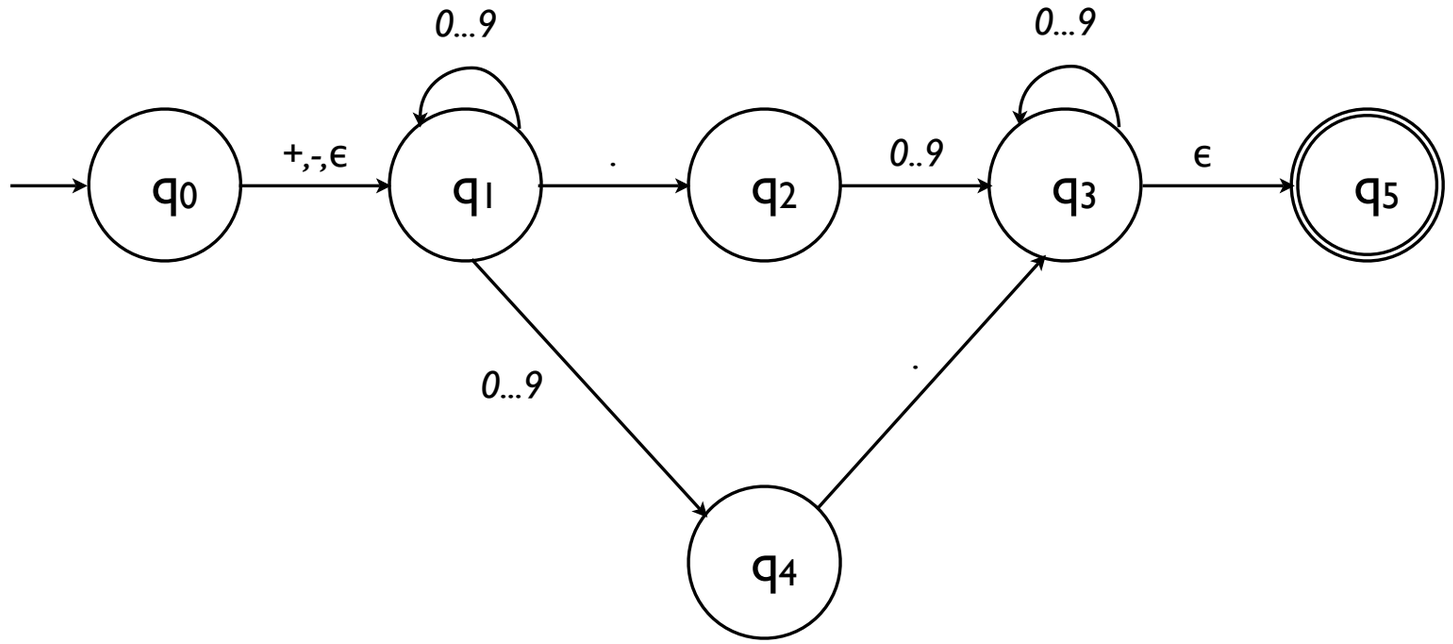
ECLOSE(7)= {7}

Erweiterte Übergangsfunktion (ϵ-NEA)

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

- $\hat{\delta}(q, \epsilon) = \mathrm{ECLOSE}(q)$ falls $x=\epsilon$ (Basisfall)
- Ansonsten kann x als wa mit $w \in \Sigma^*$ und $a \in \Sigma$ geschrieben werden. Dann gilt: $\hat{\delta}(q, wa) = \mathrm{ECLOSE}(\bigcup_{r \in \hat{\delta}(q,w)} \delta(r, a))$

Erweiterte Übergangsfunktion - Beispiel



$$
\begin{aligned}
\hat{\delta}(q_0, \epsilon) &= \text{ECLOSE}(q_0) = \{q_0, q_1\} \\
\hat{\delta}(q_0, 1) &= \text{ECLOSE}(\delta(q_0, 1) \cup \delta(q_1, 1)) \\
&= \text{ECLOSE}(\{q_1, q_4\}) = \{q_1, q_4\} \\
\hat{\delta}(q_0, 1.) &= \text{ECLOSE}(\{q_2, q_3\}) = \{q_2, q_3, q_5\} \\
\hat{\delta}(q_0, 1.2) &= \text{ECLOSE}(\{q_3\}) = \{q_3, q_5\} \\
\hat{\delta}(q_0, 1.23) &= \text{ECLOSE}(\{q_3\}) = \{q_3, q_5\}
\end{aligned}
$$

Sprache eines ϵ-NEA

Definition: Sei $E= (Q, \Sigma, \delta, q_0, F)$ ein ϵ-NEA und $\hat{\delta}$ die erweiterte Übergangsfunktion zu δ.

- E akzeptiert ein Wort $w \in \Sigma^*$, wenn $\hat{\delta}(q_0,w) \cap F \neq \emptyset$.
- $L(E)=\{w \in \Sigma^* | \hat{\delta}(q_0,w) \cap F \neq \emptyset\}$ ist die Sprache von E.

Frage: Sind ϵ-NEAs mächtiger als normale (nichtdeterministische) endliche Automaten?

Äquivalenz von DEAs, NEAs und ϵ-NEAs

Theorem 4:

(a) Für jeden deterministischen endlichen Automaten *A* gibt es einen ϵ-NEA *B* mit *L(A)=L(B)*.

(b) Für jeden ϵ-NEA *B* gibt es einen deterministischen endlichen Automaten *A* mit *L(B)=L(A)*.

Korollar:

(a) Eine Sprache ist genau dann regulär, wenn es einen ϵ-NEAgibt, der sie erkennt.

(b) ϵ-NEAs und NEAs sind äquivalent.

Beweis (Theorem 4a)

Behauptung: Für jeden deterministischen endlichen Automaten *A* gibt es einen ϵ-NEA *B* mit *L(A)=L(B)*.

Beweis:

- Laut Theorem 3 existiert ein NEA *C= (Q, Σ, δ , q0, F)* mit *L (A)=L(C)*
- Sei $\delta' : (Q \times \Sigma \cup \{\epsilon\}) \mapsto 2^Q, \delta'(q,x) = \begin{cases} \{\} & \text{wenn } x = \epsilon \\ \delta(q,x) & \text{sonst} \end{cases}$
- Dann ist *B= (Q, Σ, δ′, q0, F)* ein ϵ-NEA mit *L(B)=L(C)=L(A.)*.

q.e.d.

Beweis (Theorem 4b)

Behauptung: Für jeden ϵ-NEA *B* gibt es einen deterministischen endlichen Automaten *A* mit *L(B)=L(A)*.

Beweis (Skizze): Analog Theorem 3b, aber mit modifizierter Teilmengen-Konstruktion.

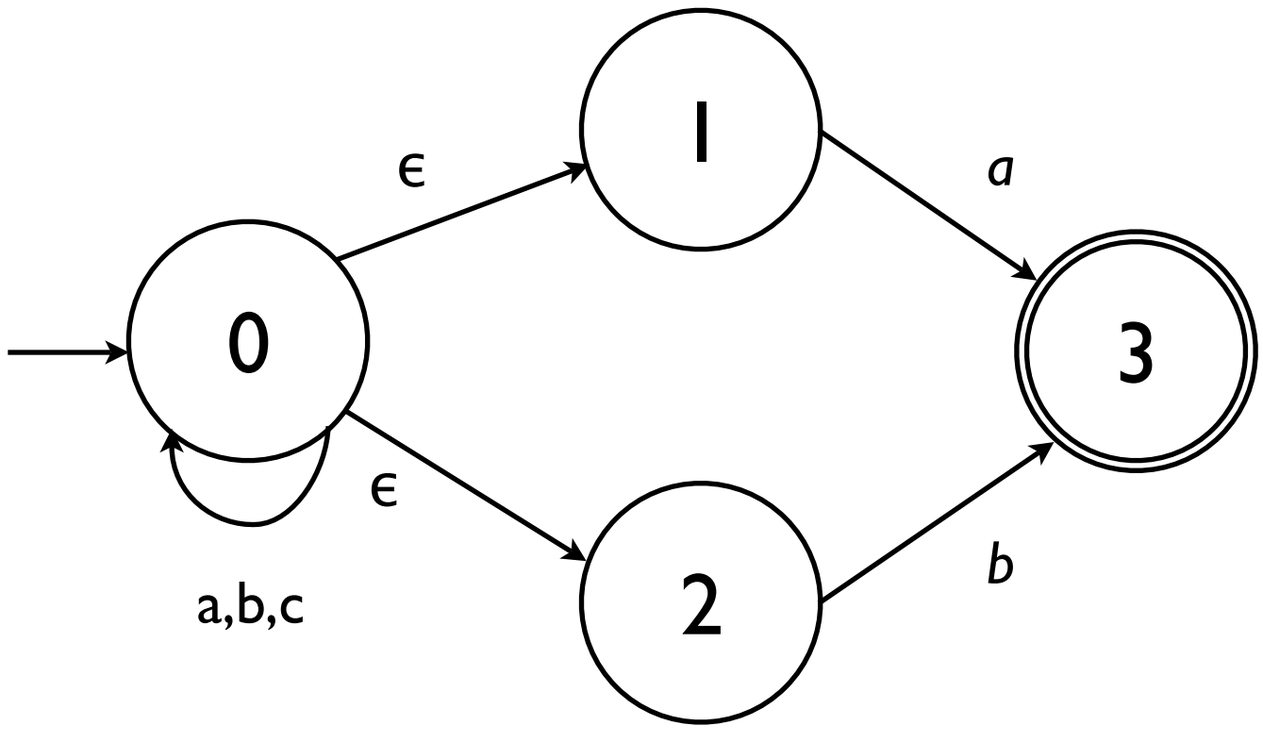
DEA-Konstruktion für ϵ-NEAs

Definition: Sei $E=(Q_E, \Sigma, \delta_E, q_0, F_E)$ ein ϵ-NEA.

Die Teilmengen-DEA-Konstruktion für ϵ-NEAs erzeugt einen DEA $D=(Q_D, \Sigma, \delta_D, q_D, F_D)$ wie folgt:

- $Q_D=2^{Q_N}$
- $q_D=ECLOSE(\{q_0\})$
- $\delta_D(S,a) = \mathrm{ECLOSE}(\bigcup_{r\in S} \delta_E(r,a))$ für alle $S\in Q_D$, $a\in\Sigma$
- $F_D=\{S\in Q_D \mid S\cap F_E\neq\emptyset\}$

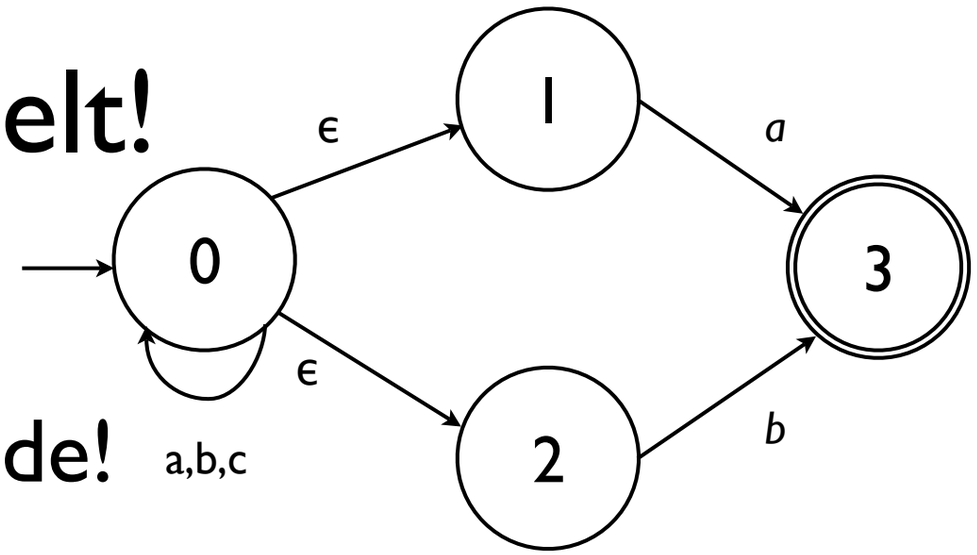
Beispiel



δ_E	ϵ	*a*	*b*	*c*
→*0*	{*1*,*2*}	{*0*}	{*0*}	{}
1	{}	{*3*}	{}	{}
2	{}	{}	{*3*}	{}
**3*	{}	{}	{}	{}

- ϵ-NEA $E=(\{0,1,2,3\}, \{a,b,c\}, \delta_E, 0, \{3\})$
- Sprache?
  - Alle Worte, die auf a oder b enden

Teilmengenkonstruktion gezielt!

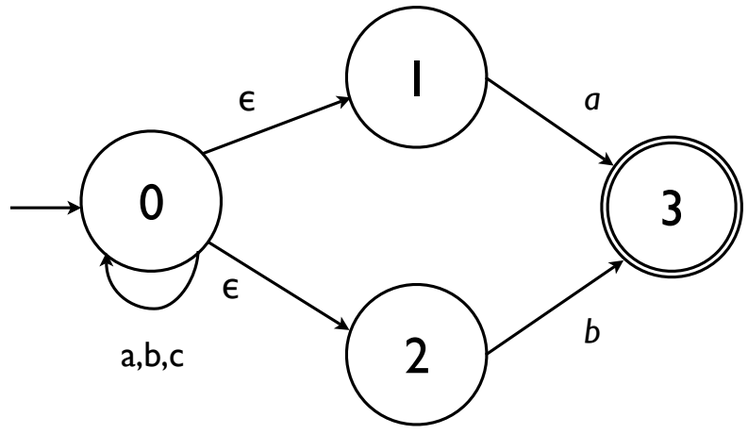


Gesucht: $D=(Q_D, \Sigma, \delta_D, q_D, F_D)$

Idee: Erzeuge nur erreichbare Zustände!

- $q_D=ECLOSE(\{0\})=\{0,1,2\} := A$
  - $\delta_D(q_D, a) = ECLOSE(\delta_E(0, a) \cup \delta_E(1, a) \cup \delta_E(2,a)) = ECLOSE(\{0, 3\}) = \{0,1,2,3\} := B$
  - $\delta_D(q_D, b) = ECLOSE(\delta_E(0, b) \cup \delta_E(1, b) \cup \delta_E(2,b)) = ECLOSE(\{0, 3\}) = \{0,1,2,3\}\ (=B)$
  - $\delta_D(q_D, c) = ECLOSE(\delta_E(0, c) \cup \delta_E(1, c) \cup \delta_E(2,c)) = ECLOSE(\{0\}) = \{0,1,2\}\ (= A)$

Teilmengenkonstruktion (2)



- $B = \{0,1,2,3\}$
  - $\delta_D(B, a) = ECLOSE(\delta_E(0, a) \cup \delta_E(1, a) \cup \delta_E(2,a) \cup \delta_E(3,a))$
    $= ECLOSE(\{0, 3\}) = \{0,1,2,3\}\ (=B)$
  - $\delta_D(B, b) = ECLOSE(\delta_E(0, b) \cup \delta_E(1, b) \cup \delta_E(2,b) \cup \delta_E(3,b))$
    $= ECLOSE(\{0, 3\}) = \{0,1,2,3\}\ (=B)$
  - $\delta_D(B, c) = ECLOSE(\delta_E(0, c) \cup \delta_E(1, c) \cup \delta_E(2,c) \cup \delta_E(3,c)) =$
    $ECLOSE(\{0\}) = \{0,1,2\}\ (=A)$

Also: *D* hat nur zwei erreichbare Zustände!

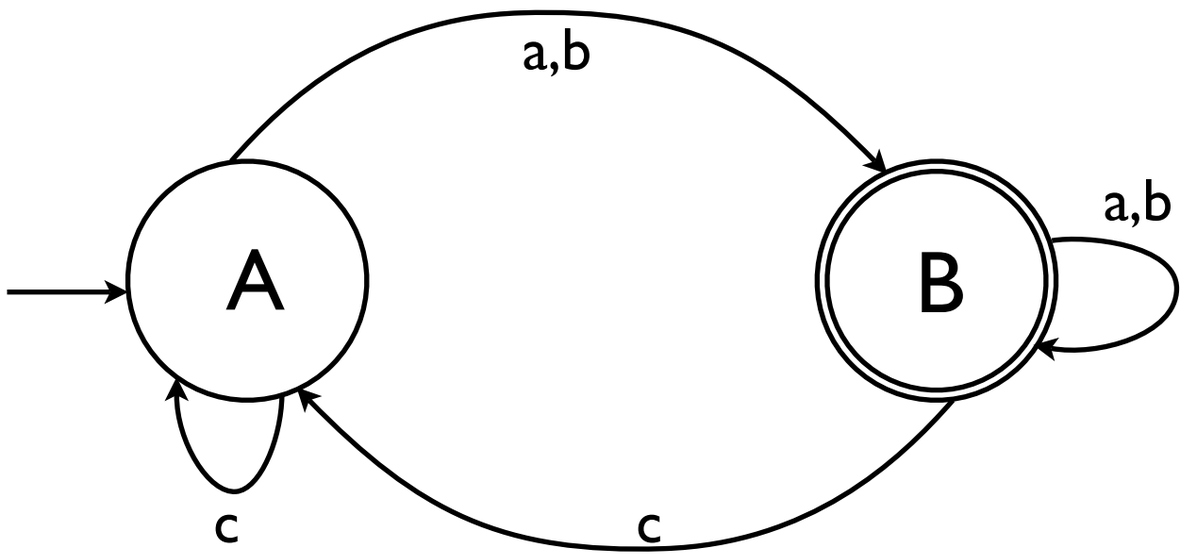
- $F_D = \{\{0,1,2,3\}\} = \{B\}$

Ergebnis der Teilmengenkonstruktion

$D=(Q_D, \Sigma, \delta_D, q_D, F_D)$ mit

- $Q_D=\{A,B\}$
- Σ wie gehabt
- δ_D nach Tabelle
- $q_D=A$
- $F_D=\{B\}$

δ_D	a	b	c
→A={0,1,2}	B	B	A
*B={0,1,2,3}	B	B	A



Erinnerung: Theorem 2

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 2(d))

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

Beweis: Nach Theorem 3 existieren ϵ-NEAs

$A=(Q_A, \Sigma, \delta_A, q_A, F_A)$ mit $L_1=L(A)$ und

$B=(Q_B, \Sigma, \delta_B, q_B, F_B)$ mit $L_2=L(B)$.

Sei o.B.A. $Q_A \cap Q_B=\emptyset$, $|F_A|= \{f_a\}$ und $\delta_A(f_a,a)=\{\}$ für alle $a \in \Sigma$.

Beweis fortgesetzt

Betrachte $E=(Q_A \cup Q_B, \Sigma, \delta, q_A, F_B)$ mit

$$\delta(q,x) = \begin{cases} \delta_A(q,x) \cup \{q_B\} & \text{wenn } q \in F_A, x = \epsilon \\ \delta_B(q,x) & \text{wenn } q \in Q_B \\ \delta_A(q,x) & \text{sonst} \end{cases}$$

Dann gilt $L(E) = L_1 \circ L_2$.

q.e.d.

Beweis (Theorem 2(e))

Behauptung: Sei $L_1 \in \Sigma^*$ eine reguläre Sprache. Dann gilt: L_1^* ist eine reguläre Sprache.

Beweis: Nach Theorem 3 existiert ϵ-NEA $A=(Q, \Sigma, \delta, q_0, F)$ mit $L_1=L(A)$. Sei o.B.A. $|F|= \{f\}$ und $\delta(f,a)=\{\}$ für alle $a \in \Sigma$.

Betrachte $E=(Q \cup \{q_0'\}, \Sigma, \delta', q_0', F \cup \{q_0'\})$ mit

$$\delta'(q,x) = \begin{cases} \{\} & \text{wenn } q = q_0', x \neq \epsilon \\ q_0 & \text{wenn } q = q_0', x = \epsilon \\ \delta(q,x) \cup \{q_0\} & \text{wenn } q \in F, x = \epsilon \\ \delta(q,x) & \text{sonst} \end{cases}$$

Dann gilt L(E)= L_1^* q.e.d.

Reguläre und andere Sprachen

Verschiedene äquivalente Klassen von Automaten

- DEAs, NEAs, ϵ-NEAs

Reguläre Sprachen:

- Zeige: Es existiert ein endlicher Automat (Klasse egal)
- Oft einfacher: "Komfortablere" Automaten

Nichtreguläre Sprachen:

- Zeige: Es gibt keinen Automaten in einer der Klassen
- Oft einfacher: Es gibt keinen DEA

Das Pumping Lemma für reguläre Sprachen

Theorem 5: Sei L eine reguläre Sprache über Σ^*. Dann existiert ein $n \in \mathbf{N}$ so dass für alle $w \in L$ mit $|w| \geq n$ gilt:

- w kann in $x,y,z \in \Sigma^*$ zerlegt werden, also $w=xyz$
- $y \neq \epsilon$ (y ist nicht das leere Wort)
- $|xy| \leq n$
- $xy^kz \in L$ für alle $k \in \mathbf{N_0}$ (w kann "gepumpt" werden)

n hängt von L ab und heißt die "pumping-Konstante" von L.

Pumping-Lemma: Beweisidee

Vergleiche Theorem 1.

- Wenn *L* regulär, dann existiert (minimaler) DEA *A* mit Zustandsmenge *Q*, der *L* akzeptiert.
- Betrachte $n=|Q|+1$
- Dann muss *A* beim Abarbeiten eines Wortes *w* mit Länge *n* einen Zustand zweimal durchlaufen
- Diese Schleife entspricht dem Wort *y* in der Zerlegung
- Der Endzustand des Automaten ist unabhängig davon, wie oft diese Schleife durchlaufen wird

Pumping-Lemma: Anmerkungen

Das Lemma gilt trivial für endliche Sprachen:

- Wähle n als die Länge des längsten Worts in L

Das Lemma macht eine Aussage über reguläre Sprachen

- Wenn L regulär, dann kann L gepumpt werden
- Umkehrschluss: Wenn L nicht gepumpt werden kann, dann ist L nicht regulär
- Falsch: Wenn L gepumpt werden kann, dann ist L regulär!
  - Es gibt nichtreguläre Sprachen, die gepumpt werden können!

Ende

Aufgaben

Sei Σ={*a*,*b*}. Welche der folgenden Sprachen sind regulär? Beweisen Sie Ihre Behauptungen.

(a) $L_1 = \{a^{3n}b^n | n \in \mathbf{N}\}$

(b) $L_2 = \{a^{3n}a^n | n \in \mathbf{N}\}$

(c) $L_3 = \{a^{2^n} | n \in \mathbf{N}\}$

(d) $L_4 = \{a^n b^m | n, m \in \mathbf{N}, n > m\}$

(e) $L_5 = \{waaaxabaybbbz | w, x, y, z \in \Sigma^*\}$