Grundbegriffe der Theoretischen Informatik

Sommersemester 2017 - Beate Bollig

Die Folien basieren auf den Materialien von Thomas Schwentick.

Teil A: Reguläre Sprachen

3: Äquivalenz der Modelle

Einleitung

- Wir kennen schon:
  - Reguläre Ausdrücke (REs)
  - NFAs (auch mit ϵ-Übergängen)
  - DFAs
- Wir können REs in ϵ-NFAs umwandeln

- In diesem Kapitel werden wir sehen:
  - ϵ-NFAs (und NFAs) lassen sich in DFAs umwandeln
  - DFAs lassen sich auch in REs umwandeln
  - $\rightarrow$ Alle vier Modelle sind gleich mächtig

- Wir werden die Größen der dabei jeweils konstruierten Objekte vergleichen
- Und mit dem Nachweis der Korrektheit von Automaten werden wir uns auch beschäftigen

Inhalt

Vom NFA zum DFA

- Wir verhalten sich NFAs zu DFAs?
- Gibt es Sprachen, die von einem NFA entschieden werden, aber von keinem DFA?
- Erstaunlicherweise und praktischerweise ist die Antwort **nein**!

Satz 3.1

- Zu jedem NFA $\mathcal{A}$ gibt es einen DFA $\mathcal{A}_D$ mit $L(\mathcal{A}_D) = L(\mathcal{A})$

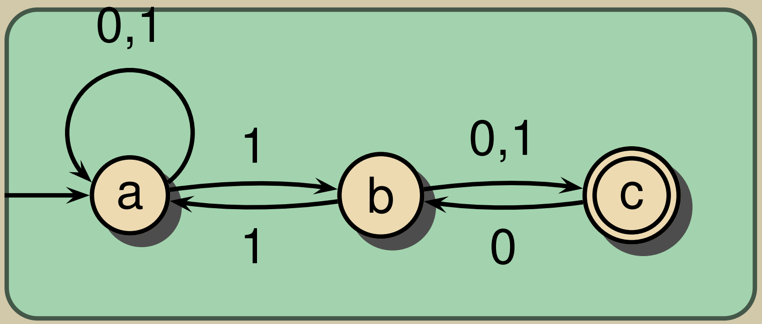
Beweisidee

- Als Zustände von $\mathcal{A}_D$ werden die Teilmengen der Zustandsmenge von $\mathcal{A}$ gewählt
- Nach Lesen eines Wortes w soll der Zustand von $\mathcal{A}_D$ die Menge aller Zustände von $\mathcal{A}$ sein, zu denen es einen Lauf vom Startzustand gibt, der w liest

  $\rightarrow$ **Potenzmengen-Automat**

Potenzmengen-Automat: Beispiel (1/3)

Beispiel

$\mathcal{A} =$ 

PINGO-Frage: `pingo.upb.de`

Was ist die Menge aller Zustände von $\mathcal{A}$, zu denen es einen Lauf vom Startzustand gibt, der 1010 liest?

(A) $\{a, b, c\}$

(B) $\emptyset$

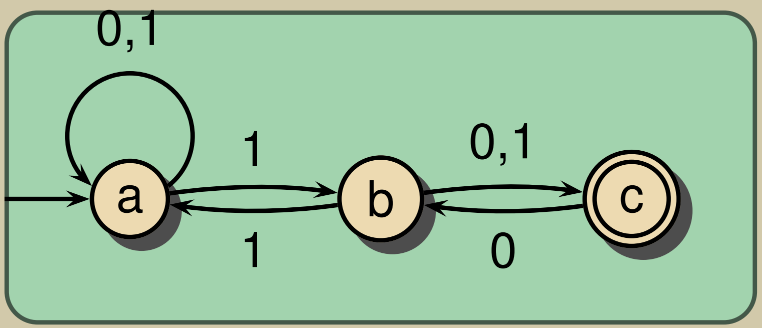
(C) $\{a, c\}$

(D) $\{a, b\}$

Potenzmengen-Automat: Beispiel (2/3)

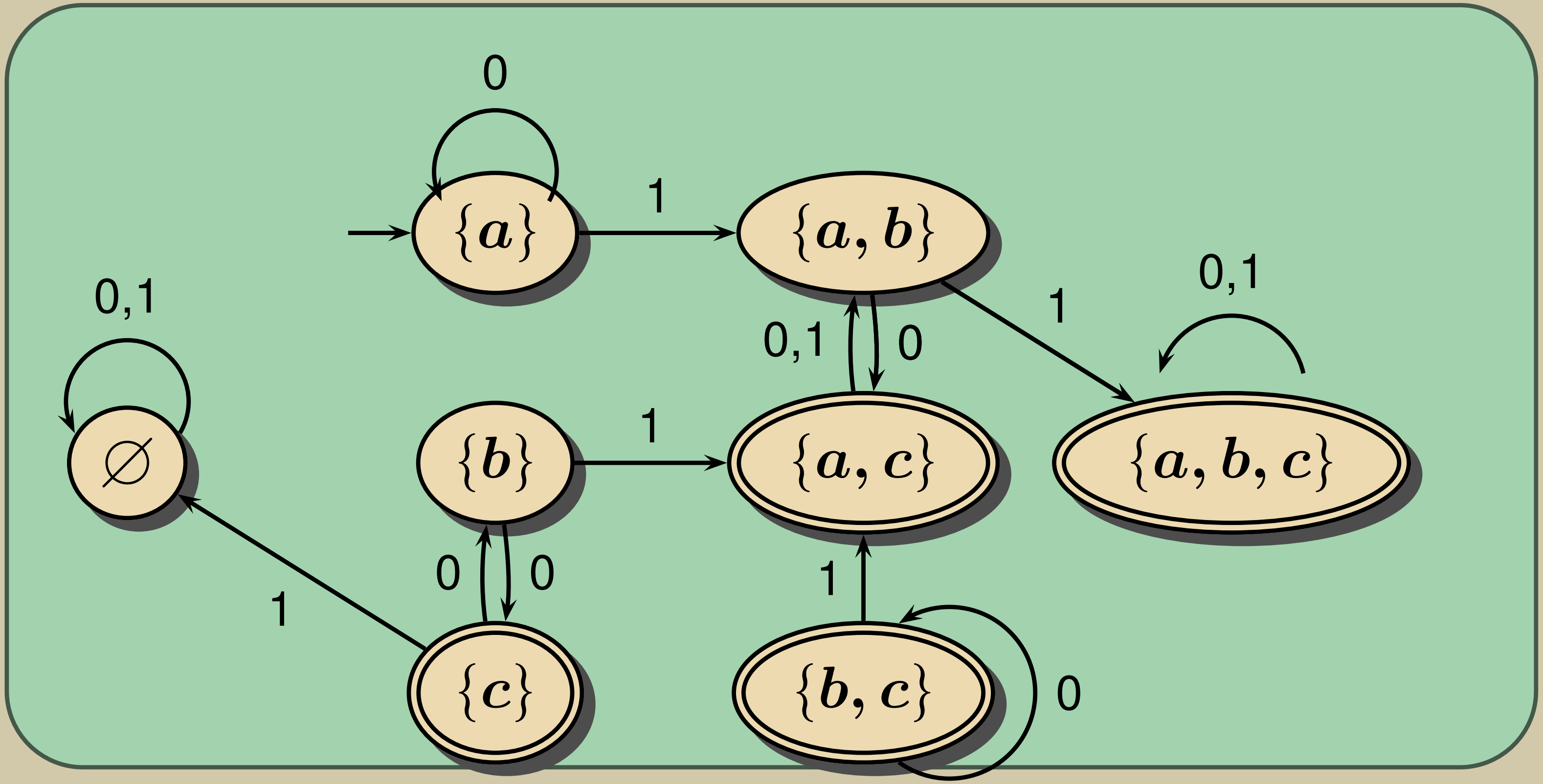
Beispiel

$\mathcal{A} =$



1010

$\mathcal{A}_D =$

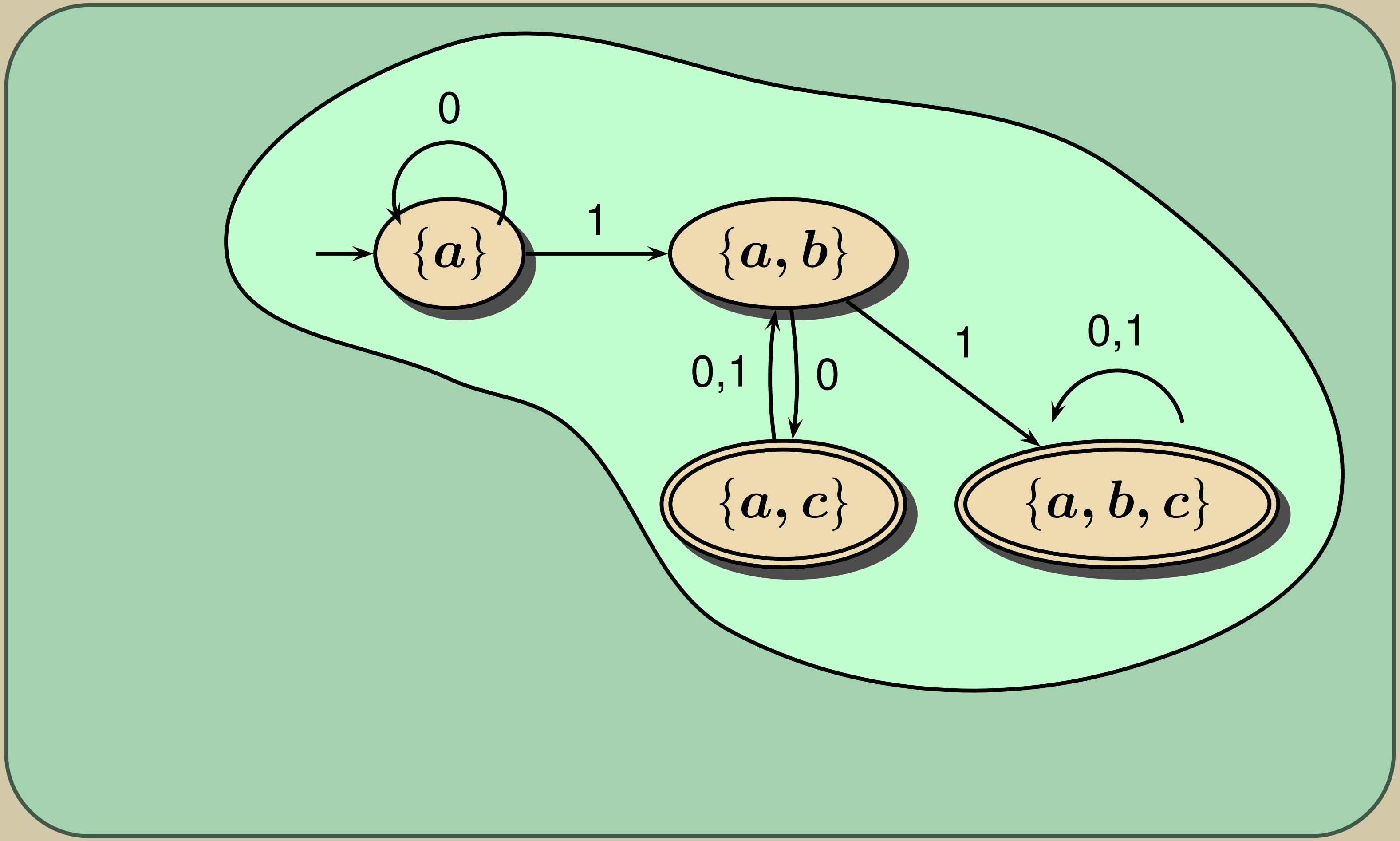


Potenzmengen-Automat: Beispiel (3/3)

- Es genügt, die von $\{a\}$ aus erreichbaren Zustände in $\mathcal{A}_D$ aufzunehmen

Beispiel

$\mathcal{A}_D =$



Einschub: Strukturelle Induktion

- Wir beweisen die Korrektheit von $\mathcal{A}_D$ mit **struktureller Induktion**

- Die Definition von regulären Ausdrücken ist ein Beispiel für eine **induktive Definition** einer Menge:
  - Zuerst werden gewisse Grundelemente der Menge definiert
    * $\varnothing$, ϵ und σ, für alle $\sigma \in \Sigma$
  - Dann wird beschrieben, wie aus gegebenen Elementen der Menge neue Elemente gewonnen werden:
    * Konkatenation, Auswahl, Wiederholung
  - Die Menge besteht dann genau aus allen so (in endlich vielen Schritten) konstruierbaren Elementen

- Ein (hoffentlich) bekanntes Beispiel einer induktiven Definition:
  - $0 \in \mathbb{N}_0$
  - $n \in \mathbb{N}_0 \quad \Rightarrow \quad n+1 \in \mathbb{N}_0$

- Induktive Definitionen ermöglichen:
  - induktive Definitionen von Funktionen auf den Elementen der Menge
  - induktive Beweise von Eigenschaften aller Elemente der Menge

- Auch die Menge Σ^* aller Strings über Σ lässt sich induktiv definieren:
  - $\epsilon \in \Sigma^*$
  - Ist $w \in \Sigma^*$ und $\sigma \in \Sigma$, so ist $w \cdot \sigma \in \Sigma^*$

- Induktive Definition einer Funktion über Σ^*:
  - $\delta^*(q, \epsilon) = q$,
  - $\delta^*(q, u\sigma) = \delta(\delta^*(q,u), \sigma)$ für $u \in \Sigma^*, \sigma \in \Sigma$

- Beweise mit **struktureller Induktion** beweisen die Aussage zuerst für die Grundelemente und dann für „zusammengesetzte Elemente"

Beweis von Satz 3.1

Satz 3.1

- Zu jedem NFA $\mathcal{A}$ gibt es einen DFA $\mathcal{A}_D$ mit $L(\mathcal{A}_D) = L(\mathcal{A})$

Beweis

- Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$
- Wir definieren $\mathcal{A}_D \stackrel{\text{def}}{=} (\mathcal{P}(Q), \Sigma, \delta_D, \{s\}, F_D)$ durch
  - $\delta_D(S, \sigma) \stackrel{\text{def}}{=}$ $\{q \mid \exists p \in S : p \xrightarrow{\sigma} q\}$, für alle $S \subseteq Q$ und $\sigma \in \Sigma$, und
  - $F_D \stackrel{\text{def}}{=}$ $\{S \subseteq Q \mid S \cap F \neq \varnothing\}$

- Für jeden String $w \in \Sigma^*$ sei $R(w) \stackrel{\text{def}}{=} \{q \mid s \xrightarrow{w} q\}$

Beweis (Forts.)

- Durch Induktion nach w zeigen wir:
$$\delta_D^*(\{s\}, w) = R(w) \qquad (*)$$

- $w = \epsilon$: $\delta_D^*(\{s\}, \epsilon) = \{s\} = R(\epsilon)$

- $w = u\sigma$:

$$\begin{aligned}
\delta_D^*(\{s\}, w) &= \delta_D(\delta_D^*(\{s\}, u), \sigma) && \text{☞ Def } \delta_D^* \\
&= \delta_D(R(u), \sigma) && \text{☞ Induktion} \\
&= \{q \mid \exists p \in R(u) : p \xrightarrow{\sigma} q\} && \text{☞ Def } \delta_D \\
&= \{q \mid \exists p : s \xrightarrow{u} p \xrightarrow{\sigma} q\} && \text{☞ Def } R \\
&= R(u\sigma) = R(w)
\end{aligned}$$

- Also:

$\mathcal{A}_D$ akzeptiert w

$\iff \delta_D^*(\{s\}, w) \in F_D$ ☞ Def „DFA akzeptiert"

$\iff \delta_D^*(\{s\}, w) \cap F \neq \varnothing$ ☞ Def F_D

$\iff R(w) \cap F \neq \varnothing$ ☞ $(*)$

$\iff \mathcal{A}$ akzeptiert w ☞ Def „NFA akzeptiert"

Vom ϵ-NFA zum DFA (1/2)

- Wir haben REs nicht in NFAs sondern in ϵ-NFAs umgewandelt

$\rightarrow$ ϵ-NFAs müssen auch noch in DFAs umgewandelt werden

Proposition 3.2

- Zu jedem ϵ-NFA $\mathcal{A}$ gibt es einen DFA $\mathcal{A}_D$ mit $L(\mathcal{A}_D) = L(\mathcal{A})$

Beweisskizze

- Sehr ähnlich zur Umwandlung von NFAs in DFAs

- Wir verwenden einen neuen Begriff:
  - **ϵ-closure**$(p) \stackrel{\text{def}}{=} \{q \mid p \xrightarrow{\epsilon} q\}$
    * (Menge aller von p aus ohne Lesen eines Symbols erreichbaren Zustände)
  - Für $S \subseteq Q$:

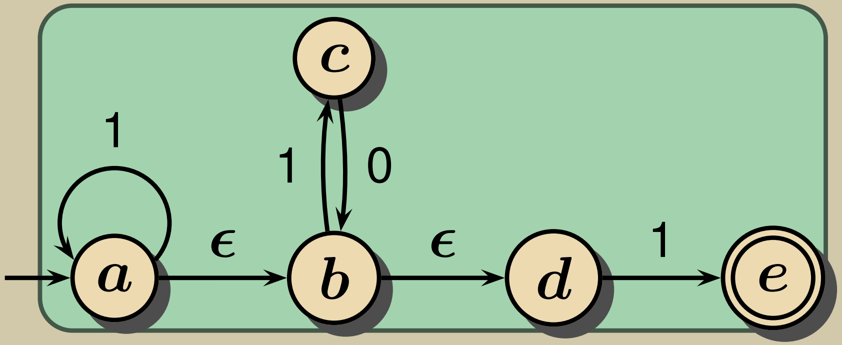
$$\textbf{ϵ-closure}(S) \stackrel{\text{def}}{=} \bigcup_{q \in S} \epsilon\text{-closure}(q)$$

- Gegenüber dem Beweis von Satz 3.1 zu ändern:
  - Startzustand: ϵ-closure(s) statt $\{s\}$
  - $\delta_D(S, \sigma) \stackrel{\text{def}}{=}$

$$\epsilon\text{-closure}(\{q \mid \exists p \in S : p \xrightarrow{\sigma} q\})$$

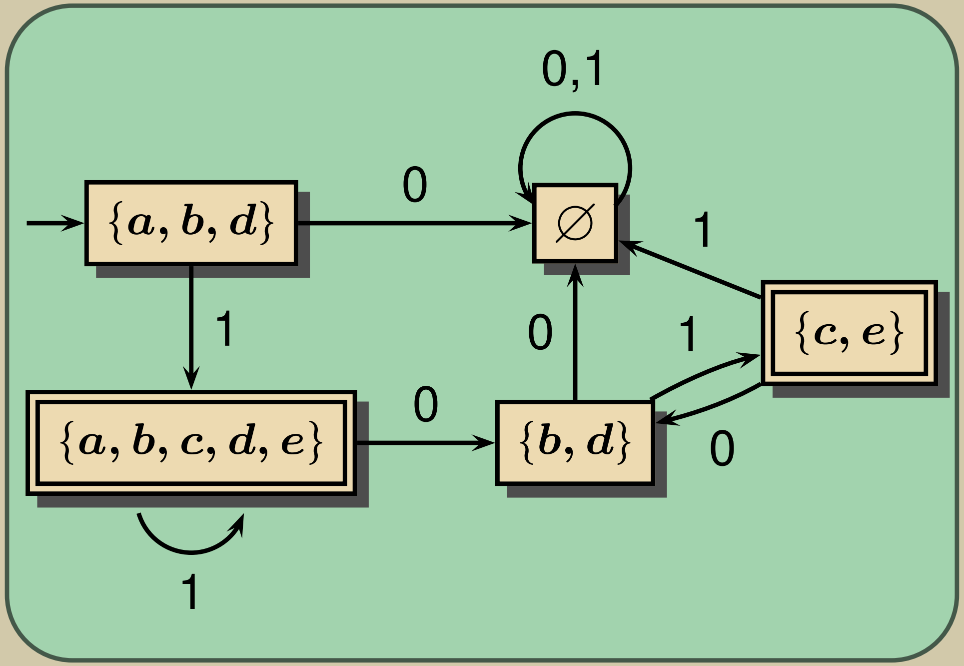
Vom ϵ-NFA zum DFA (2/2)

Beispiel: ϵ-NFA

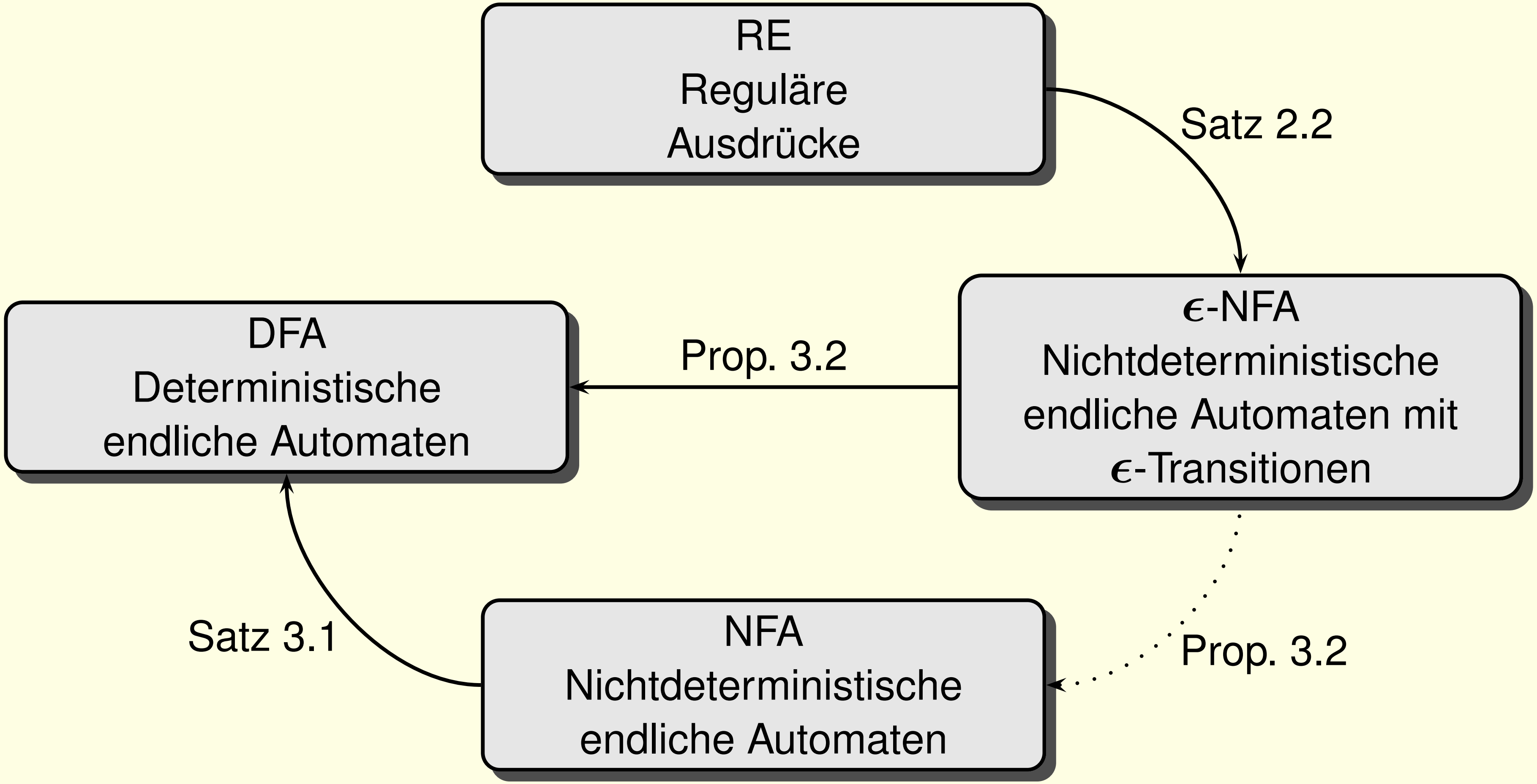


- ϵ-closure$(a) = \{a, b, d\}$
- ϵ-closure$(b) = \{b, d\}$

Beispiel: äquivalenter DFA

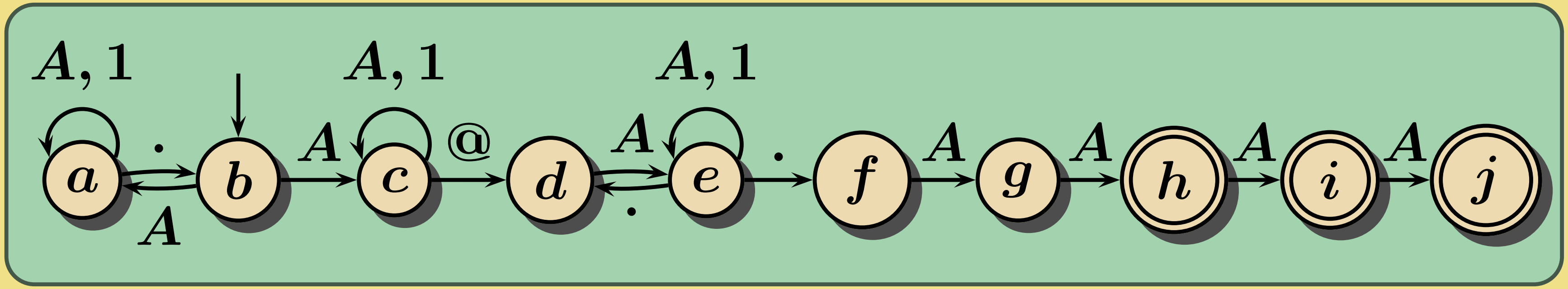


Die Äquivalenz der Modelle (Forts.)



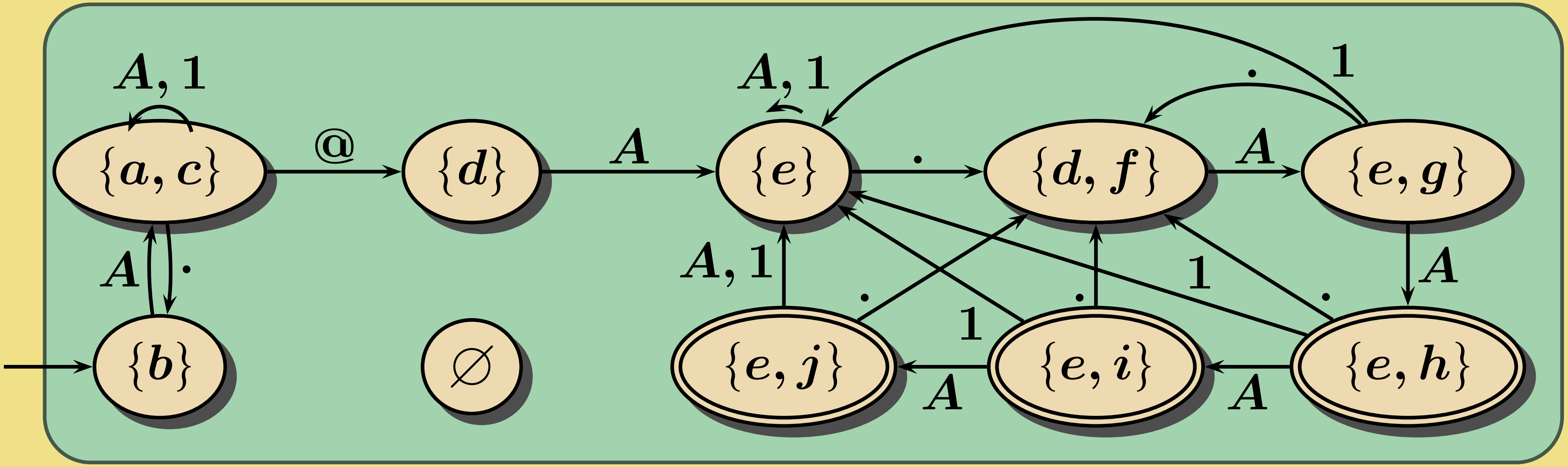
Vom RE zum DFA

- Im letzten Kapitel hatten wir aus dem erweiterten regulären Ausdruck für Mail-Adressen bereits einen NFA konstruiert:



✎ Zur Erinnerung: A steht für $a, \ldots, z, A, \ldots, Z$ und 1 für $0, \ldots, 9, -, \_$

- Dieser lässt sich in den folgenden DFA umwandeln:



✎ Alle übrigen Übergänge führen in den Zustand $\emptyset$

Inhalt

Endliche Automaten vs. reguläre Ausdrücke

- Um den Nachweis der Äquivalenz der betrachteten Modelle abzuschließen, zeigen wir folgendes Resultat

Proposition 3.3 [McNaughton, Yamada 60]

- Zu jedem DFA $\mathcal{A}$ gibt es einen RE α mit
$$L(\alpha) = L(\mathcal{A})$$

- Für den Beweis von Proposition 3.3 betrachten wir zuerst einen konstruktiven und anschaulichen Weg, um von $\mathcal{A}$ zu α zu kommen:
Zustandselimination

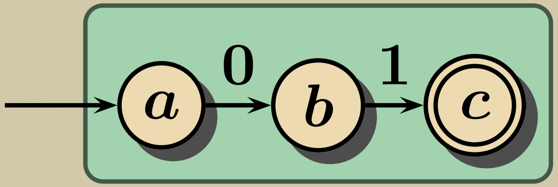
- Der Nachweis, dass diese Konstruktion korrekt ist, ist aber technisch mühsam
- Deshalb führen wir den formalen Beweis für Proposition 3.3 dann auf eine etwas weniger anschauliche, aber leicht hinzuschreibende Weise

Vom DFA zum RE: Anschauliche Vorgehensweise (1/4)

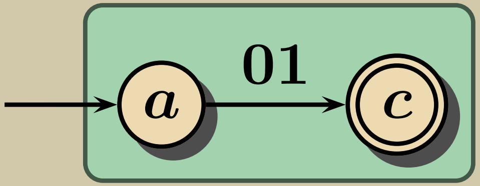
- Grundidee der anschaulichen Vorgehensweise
  - Wir verwenden ein **hybrides Automatenmodell**, dessen Transitionen mit regulären Ausdrücken (statt einzelnen Zeichen) beschriftet sind
  - Durch **sukzessives Entfernen von Zuständen** wird schließlich ein einzelner regulärer Ausdruck erreicht

Ein einfaches Beispiel

- Wandle



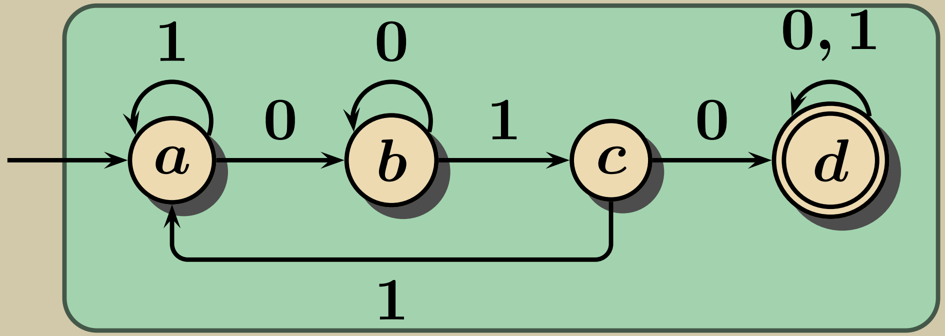
um in

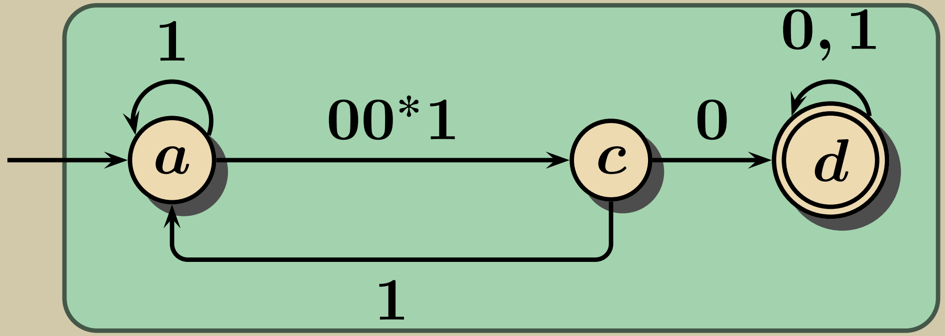


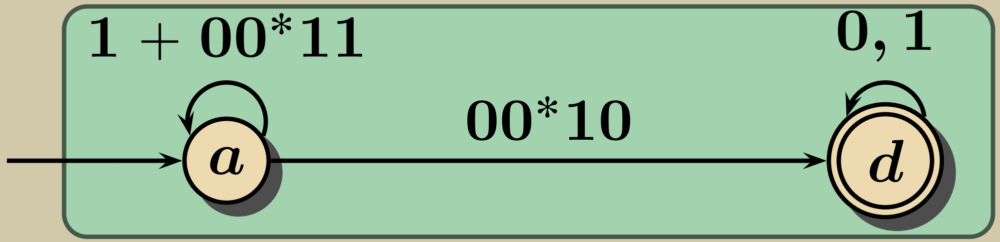
und erhalte den regulären Ausdruck $\mathbf{01}$

Ein komplizierteres Beispiel

- Umwandlung von





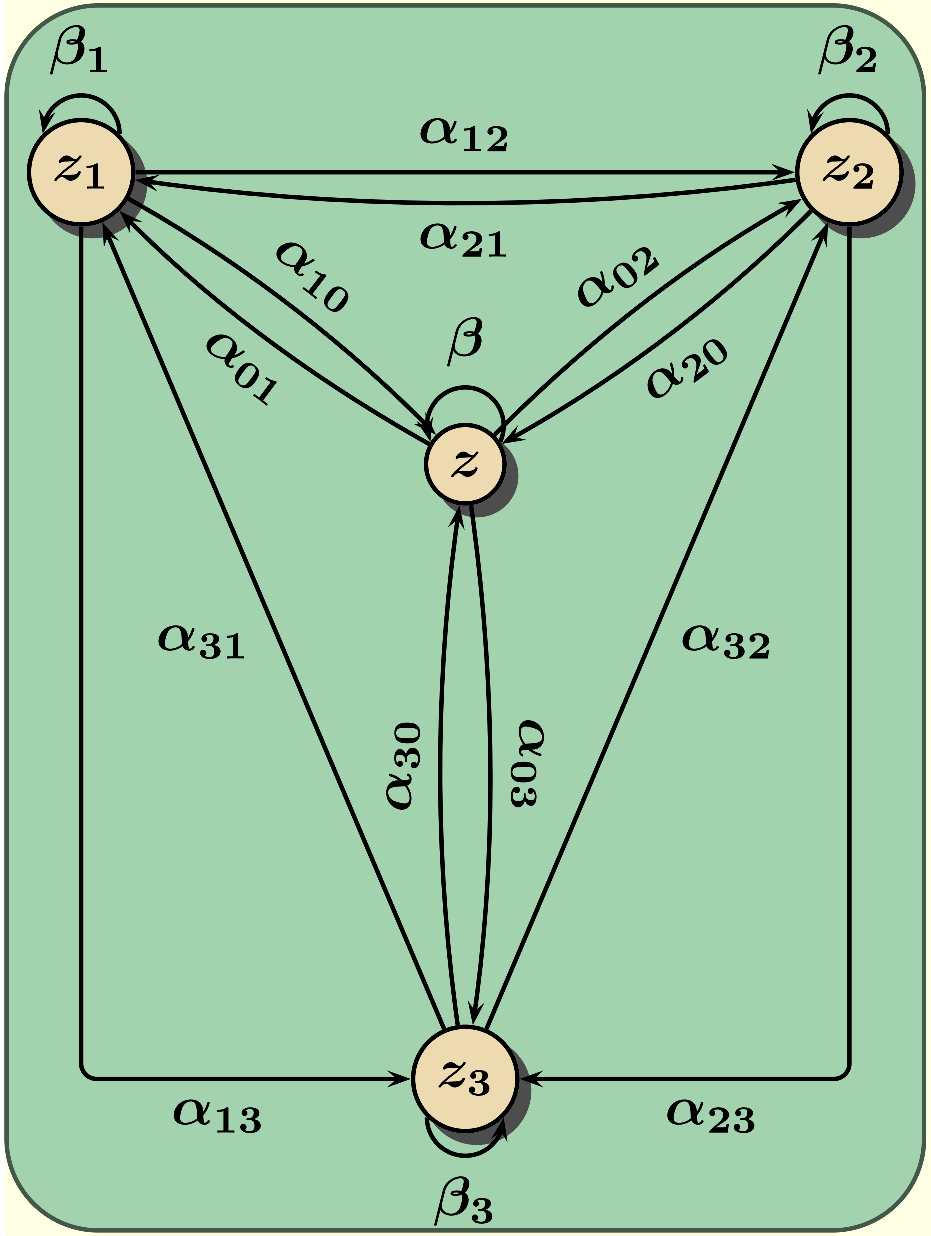


ergibt den regulären Ausdruck

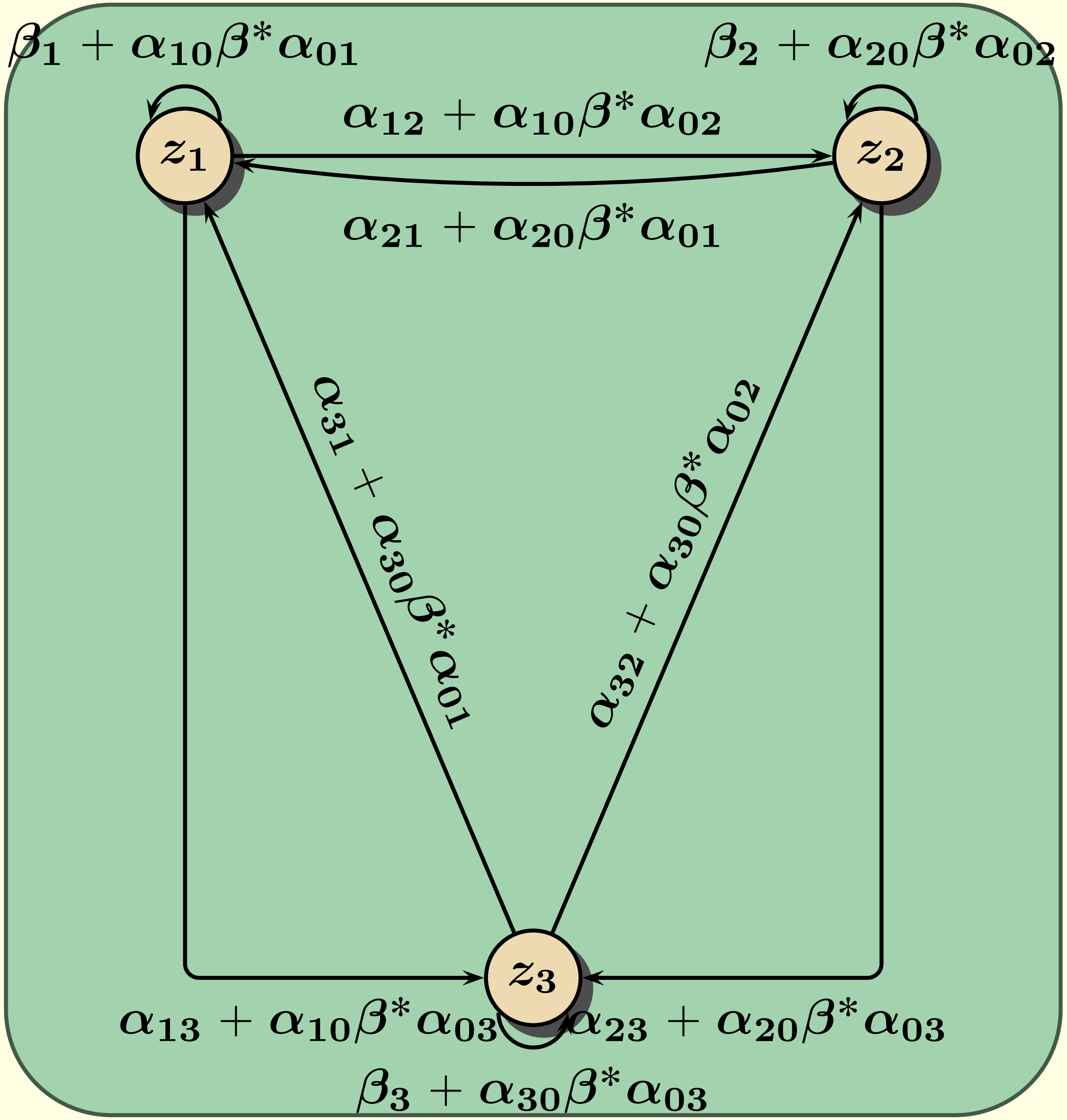
$$(\mathbf{1}+\mathbf{00}^*\mathbf{11})^*\mathbf{00}^*\mathbf{10}(\mathbf{0}+\mathbf{1})^*$$

Vom DFA zum RE: Anschauliche Vorgehensweise (2/4)

- Im Allgemeinen wird ein **nicht akzeptierender Zustand** z wie folgt entfernt:

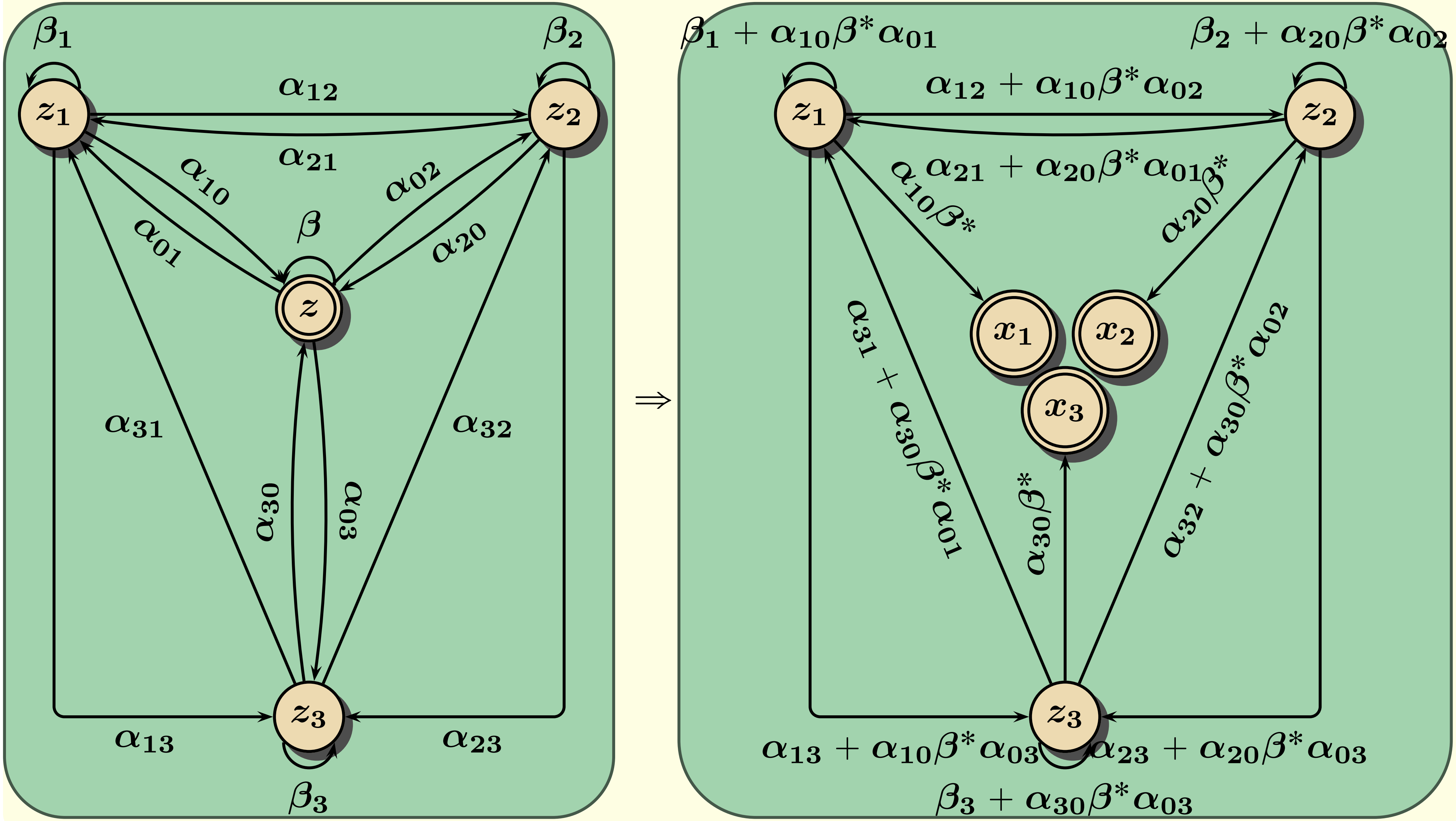


$\Rightarrow$



Vom DFA zum RE: Anschauliche Vorgehensweise (3/4)

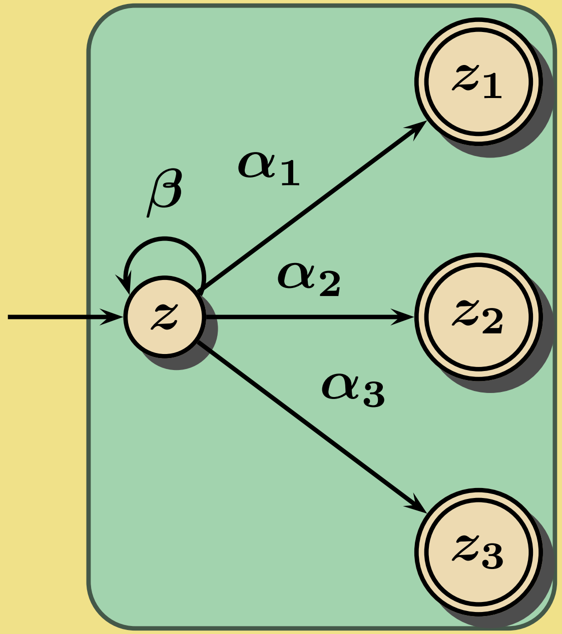
- Im Allgemeinen wird ein **akzeptierender Zustand** z wie folgt behandelt:



Vom DFA zum RE: Anschauliche Vorgehensweise (4/4)

- Am Ende erhalten wir einen hybriden Automaten in einer von zwei Formen:

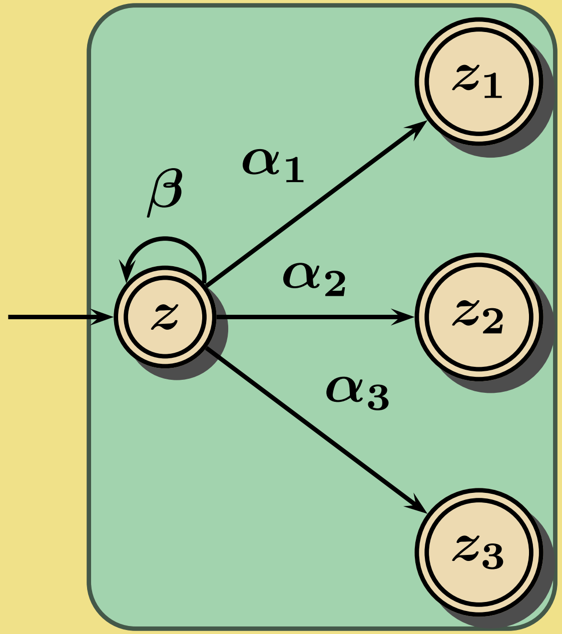
- 1. Fall:
  Startzustand ist **nicht akzeptierend**:



Der zugehörige reguläre Ausdruck ist dann $\beta^*(\alpha_1 + \alpha_2 + \alpha_3)$

✎ Der Fall, dass es in z keine Schleife gibt, enspricht $\beta = \emptyset$ und liefert $\emptyset^*(\alpha_1 + \alpha_2 + \alpha_3) \equiv (\alpha_1 + \alpha_2 + \alpha_3)$

- 2. Fall: Startzustand ist **akzeptierend**:



Der zugehörige reguläre Ausdruck ist dann $\beta^*(\alpha_1 + \alpha_2 + \alpha_3 + \epsilon)$

- Wie gesagt: der Beweis der Korrektheit dieser Vorgehensweise ist etwas mühsam
- Deshalb betrachten wir jetzt einen Beweis, der weniger anschaulich ist, sich aber leichter aufschreiben lässt

Vom DFA zum RE: Beweis (1/4)

Proposition 3.3 [McNaughton, Yamada 60]

- Zu jedem DFA $\mathcal{A}$ gibt es einen regulären Ausdruck α mit $L(\alpha) = L(\mathcal{A})$

Beweisskizze

- Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$
- Seien oBdA $Q = \{1, \dots, n\}$ und $s = 1$

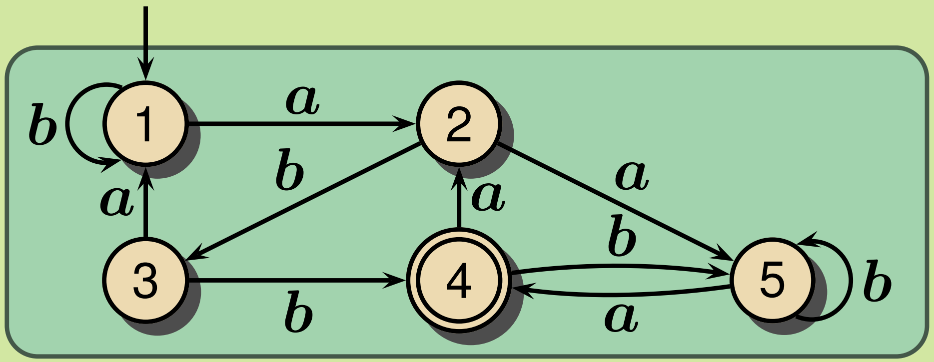
- Für jedes $i, j \in \{1, \dots, n\}$ und $k \in \{0, \dots, n\}$ soll $L_{i,j}^k$ die Menge aller Strings $w \in \Sigma^*$ sein, für die der Automat
  - vom Zustand i in den Zustand j übergeht,
  - und zwischendurch nur Zustände aus $\{1, \dots, k\}$ annimmt

- $L_{i,j}^k \stackrel{\text{def}}{=}$ Menge aller Strings w mit:
  - $\delta^*(i, w) = j$ und
  - für alle echten Präfixe $v \neq \epsilon$ von w ist $\delta^*(i, v) \leqslant k$

3.1

PINGO-Frage: `pingo.upb.de`

Ein regulärer Ausdruck für die Menge $L_{1,5}^3$ zum Automaten



ist:

(A) aa

(B) aab^*

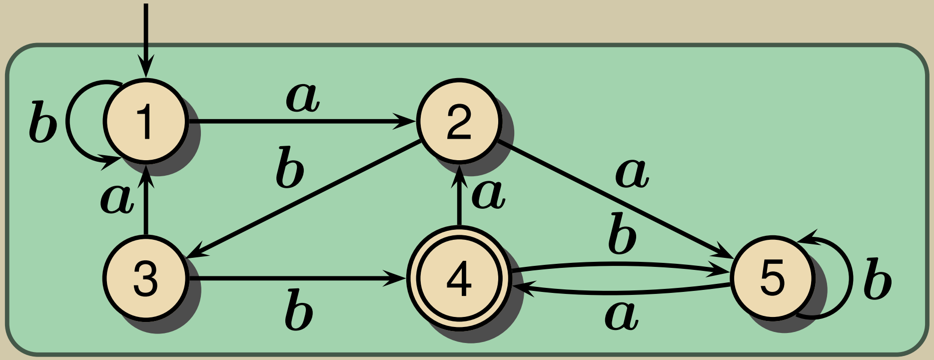
(C) $b^*a(bab^*a)^*a$

(D) $b^*a(bab^*a)^*ab^*$

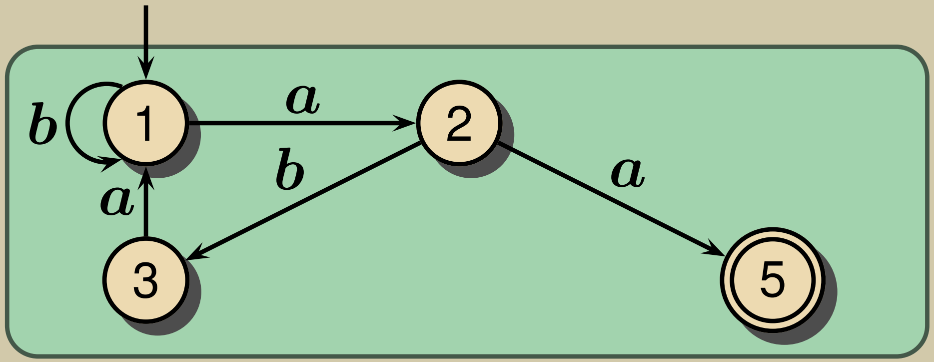
Vom DFA zum RE: Beweis (2/4)

Beispiel

- Für den Automaten



entspricht die Menge $L^3_{1,5}$
dem (partiellen) Teil-DFA



Vom DFA zum RE: Beweis (3/4)

Beweisskizze (Forts.)

- Behauptung: für jede Menge $L^k_{i,j}$ gibt es einen regulären Ausdruck $\alpha^k_{i,j}$ mit $L(\alpha^k_{i,j}) = L^k_{i,j}$

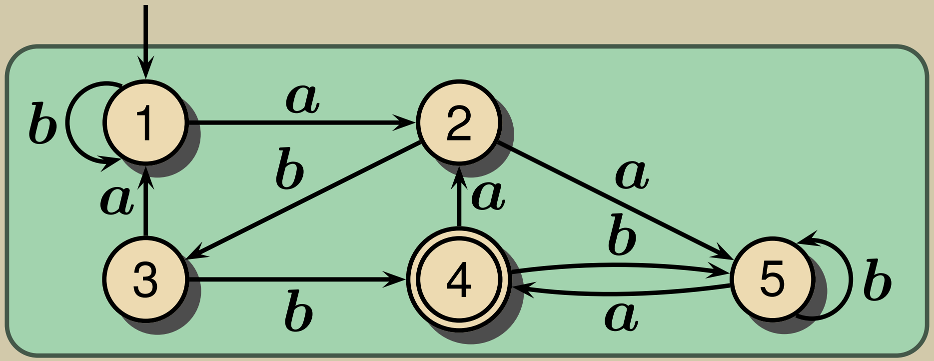
- Beweis durch Induktion nach k

- $k = 0$:
  - Für $i \neq j$ ist
  $$L^0_{i,j} = \{\sigma \in \Sigma \mid \delta(i,\sigma) = j\}$$
    ✎ Hier sind nur direkte Übergänge erlaubt! Keine Zwischenschritte!
  - Analog:
  $$L^0_{i,i} = \{\epsilon\} \cup \{\sigma \in \Sigma \mid \delta(i,\sigma) = i\}$$

- $L^0_{i,j}$ und $L^0_{i,i}$ sind endlich und können deshalb durch reguläre Ausdrücke beschrieben werden

Vom DFA zum RE: Beweis (4/4)

Beispiel



$$\underbrace{ab}_{\in L^2_{1,3}}\underbrace{abbbab}_{\in L^2_{3,3}}\underbrace{abab}_{\in L^2_{3,3}}\underbrace{abaa}_{\in L^2_{3,5}} \in L^3_{1,5} \text{ (!)}$$

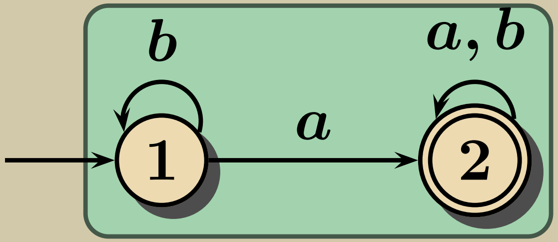
Beweisskizze (Forts.)

- $k > 0$:
  $$L^k_{i,j} = L^{k-1}_{i,j} \cup L^{k-1}_{i,k}(L^{k-1}_{k,k})^* L^{k-1}_{k,j}$$
- Nach Induktion gibt es für jede auf der rechten Seite vorkommende Sprache einen regulären Ausdruck, also auch für $L^k_{i,j}$
- Der RE α ist dann $\sum_{i\in F} \alpha^n_{1,i}$

- Die Konstruktion funktioniert natürlich genauso auch für NFAs

Vom DFA zum RE: Beispiel

Beispiel



- $k = 0$:
  - $\alpha^0_{1,1} = b + \epsilon, \quad \alpha^0_{1,2} = a, \quad \alpha^0_{2,1} = \emptyset, \quad \alpha^0_{2,2} = a + b + \epsilon$
- $k = 1$:
  - $\alpha^1_{1,1} = b + \epsilon + (b + \epsilon)(b + \epsilon)^*(b + \epsilon) \qquad \equiv b^*$
  - $\alpha^1_{1,2} = a + (b + \epsilon)(b + \epsilon)^* a \qquad \equiv b^* a$
  - $\alpha^1_{2,1} = \emptyset + \emptyset(b + \epsilon)^*(b + \epsilon) \qquad \equiv \emptyset$
  - $\alpha^1_{2,2} = (a + b + \epsilon) + \emptyset(b + \epsilon)^* a \qquad \equiv a + b + \epsilon$
- $k = 2$:
  - $\alpha^2_{1,1} = b^* + b^* a(a + b + \epsilon)^* \emptyset \qquad \equiv b^*$
  - $\alpha^2_{1,2} = b^* a + b^* a(a + b + \epsilon)^*(a + b + \epsilon) \qquad \equiv b^* a(a + b)^*$
  - $\alpha^2_{2,1} = \emptyset + (a + b + \epsilon)(a + b + \epsilon)^* \emptyset \qquad \equiv \emptyset$
  - $\alpha^2_{2,2} = (a + b + \epsilon) + (a + b + \epsilon)(a + b + \epsilon)^*(a + b + \epsilon) \qquad \equiv (a + b)^*$
- $\alpha = b^* a(a + b)^*$ (mit $k = 2$ hier nur $\alpha^2_{1,2}$ nötig)

Reguläre Sprachen: Äquivalenz

- Insgesamt haben wir den folgenden Satz bewiesen:

Satz 3.4

Für eine Sprache $L \subseteq \Sigma^*$ sind äquivalent:

(a) $L = L(\alpha)$ für einen RE α

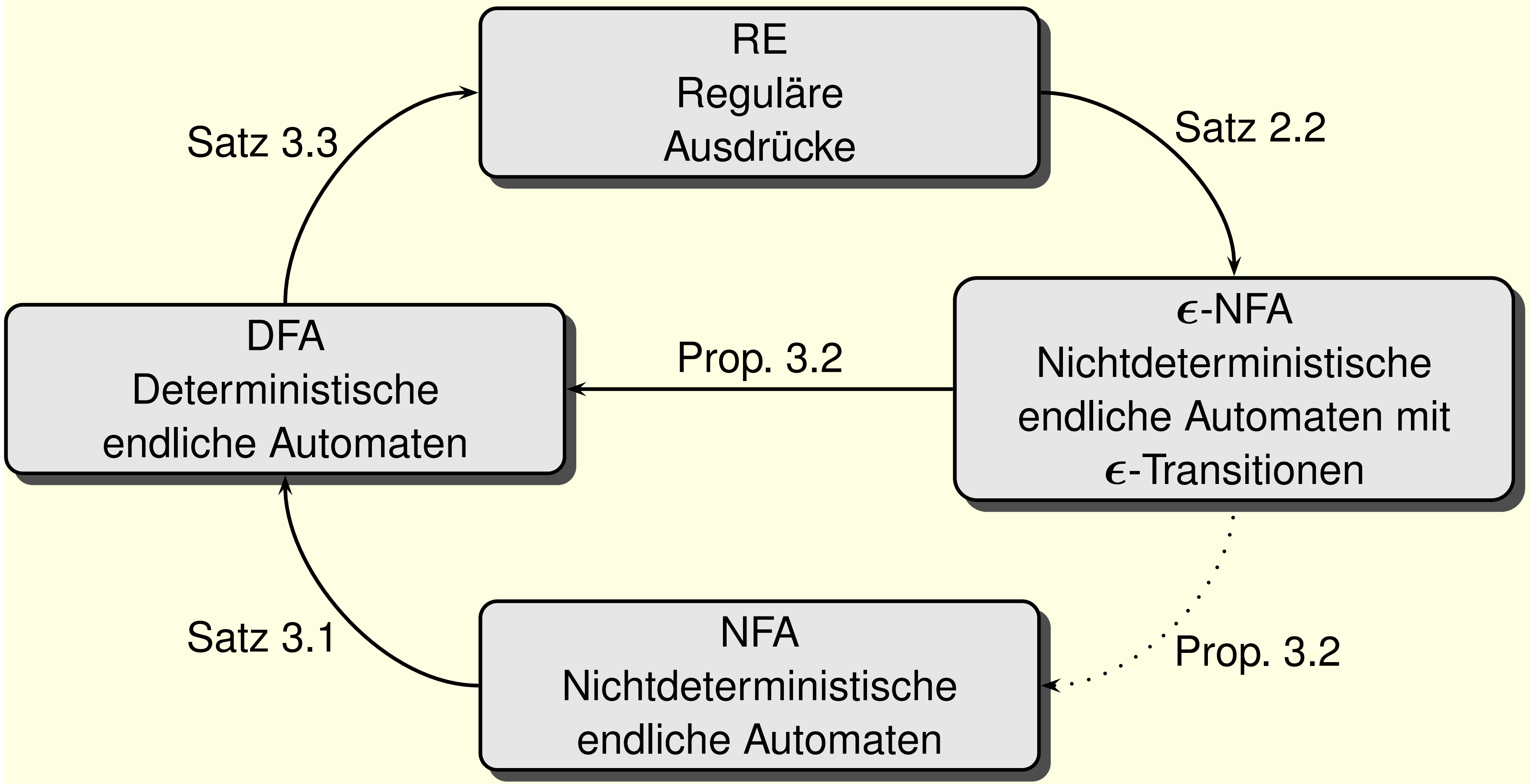
(b) $L = L(\mathcal{A})$ für einen DFA $\mathcal{A}$

(c) $L = L(\mathcal{A})$ für einen NFA $\mathcal{A}$

(d) $L = L(\mathcal{A})$ für einen ϵ-NFA $\mathcal{A}$

- Die regulären Sprachen bilden also eine sehr robuste Klasse von Sprachen

Die Äquivalenz der Modelle (Forts.)



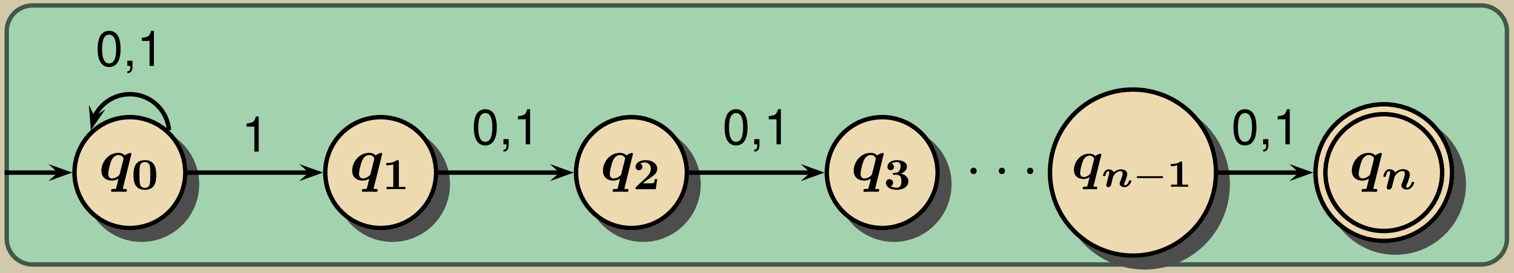
Inhalt

Vom NFA zum DFA: Größe des Potenzmengenautomaten

- Wir groß kann der Potenzmengenautomat $\mathcal{A}_D$ im Verhältnis zu $\mathcal{A}$ im Beweis von Satz 3.1 werden?

- Klar: maximal $2^{|Q|}$ Zustände ($=$ Anzahl der Teilmengen von Q)
- Aber: kann es wirklich passieren, dass alle möglichen Teilmengen von Q erreichbar sind?

Beispiel



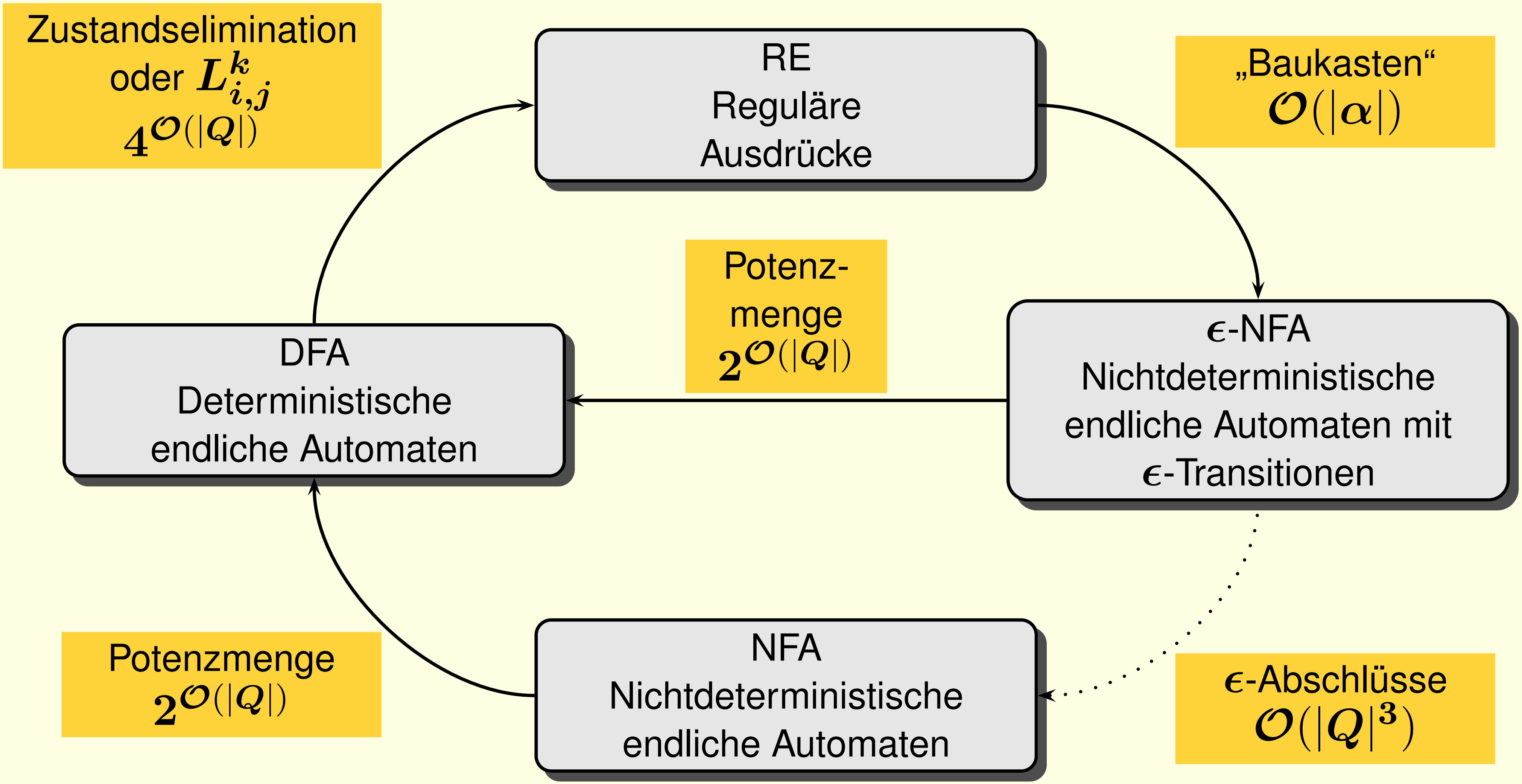
- Dieser Automat akzeptiert, falls das n-te Zeichen von rechts eine 1 ist
- Wir werden in Kapitel 4 zeigen: es gibt keinen DFA mit $< 2^{|Q|-1}$ Zuständen für diese Sprache

Vom DFA zum RE: Größe des REs

- Wie groß wird der RE α, der bei der Umwandlung vom DFA $\mathcal{A}$ nach dem Beweis von Proposition 3.3 entsteht?

- Sei $g(k)$ die maximale Länge eines Ausdrucks für $L_{i,j}^{k}$
- Dann gilt:
  - $g(0) = \mathcal{O}(|\Sigma|)$
  - $g(k) \leqslant 4g(k-1) + \mathcal{O}(1)$
- Also: $g(n) = 4^{\mathcal{O}(n)}|\Sigma|$

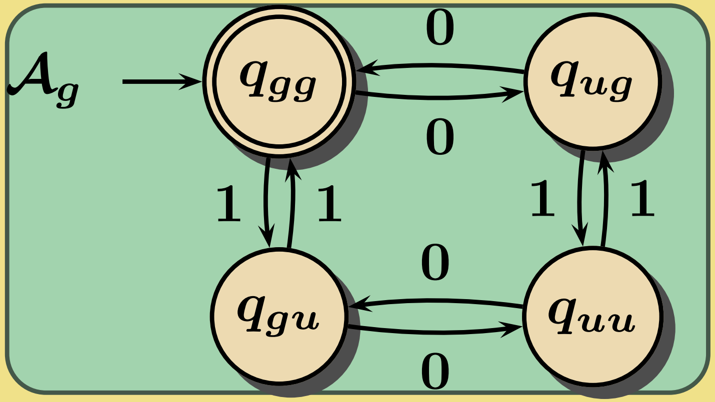
Die Äquivalenz der Modelle: Größenverhältnisse



Inhalt

Korrektheitsbeweis für Automaten (1/2)

- Zur Erinnerung: L_g ist die Menge aller Strings über $\{0, 1\}$ mit gerade vielen Einsen und Nullen



- Es erscheint offensichtlich, dass $\mathcal{A}_g$ die Sprache L_g entscheidet
- Können wir das auch **beweisen**?

- Wir benötigen für den Beweis ein wenig Notation:
  - $\#_\tau(w) \stackrel{\text{def}}{=}$ Häufigkeit des Vorkommens des Zeichens τ im String w
    * z.B.: $\#_a(baaba) = 3$
  - $n \equiv_k m \stackrel{\text{def}}{\Leftrightarrow}$ n und m haben bei Division durch k denselben Rest (für $k \in \mathbb{N}$)
    * z.B.: $5 \equiv_3 2$

- Mit dieser Notation definieren wir formal:
  - $L_g \stackrel{\text{def}}{=} \{w \in \{0,1\}^* \mid \#_0(w) \equiv_2 0, \#_1(w) \equiv_2 0\}$

Proposition 3.5

- $L(\mathcal{A}_g) = L_g$

- Korrektheitsbeweise für Automaten zeigen meisten durch Induktion nach w, dass die intuitive Bedeutung der Zustände mit der tatsächlichen Bedeutung übereinstimmt

Korrektheitsbeweis für Automaten (2/2)

Proposition 3.5

- $L(\mathcal{A}_g) = L_g$

Beweisskizze

- Wir zeigen durch Induktion nach $|w|$, dass für alle $w \in \Sigma^*$ gilt:
  - $\delta^*(q_{gg}, w) = q_{gg} \iff$
    $\#_0(w) \equiv_2 0, \#_1(w) \equiv_2 0$
  - $\delta^*(q_{gg}, w) = q_{ug} \iff$
    $\#_0(w) \equiv_2 1, \#_1(w) \equiv_2 0$
  - $\delta^*(q_{gg}, w) = q_{gu} \iff$
    $\#_0(w) \equiv_2 0, \#_1(w) \equiv_2 1$
  - $\delta^*(q_{gg}, w) = q_{uu} \iff$
    $\#_0(w) \equiv_2 1, \#_1(w) \equiv_2 1$

- Daraus folgt dann die Proposition wegen $F = \{q_{gg}\}$

Beweisskizze (Forts.)

- $w = \epsilon$: ✓

- $w = v\sigma$ für ein $v \in \Sigma^*$ und ein $\sigma \in \Sigma$:
  - Nach Induktion gilt für v die Induktionsbehauptung

  - Wir unterscheiden 8 Fälle, je nach σ und $\delta^*(q_{gg}, v)$:
    - Beispielfall:
      $\sigma = 1, \delta^*(q_{gg}, v) = q_{ug}$
      ➡ $\#_0(v) \equiv_2 1, \#_1(v) \equiv_2 0$
      ☞ Induktion
      ➡ $\#_0(w) \equiv_2 1, \#_1(w) \equiv_2 1$
      ☞ da $\sigma = 1$
      ➡ Behauptung ☞ da $\delta(q_{ug}, 1) = q_{uu}$

  - Die anderen sieben Fälle sind analog

Zusammenfassung

Themen dieser Vorlesung

- Umwandlung von NFAs und ϵ-NFAs in DFAs (Potenzmengenkonstruktion)
- Umwandlung von DFAs in REs
- Korrektheitsbeweise für endliche Automaten

Kapitelfazit

- Alle betrachteten Modelle beschreiben reguläre Sprachen
- Einige Umwandlungen zwischen den Modellen können exponentiell große Objekte erzeugen

Erläuterungen: Präfixe, Suffixe und Teilstrings

Bemerkung 3.1

- Sind x, y, z Wörter und ist $w = xyz$, so heißt
  - x ein **Präfix** von w,
  - y ein **Teilstring** von w und
  - z ein **Suffix** von w

- Dabei können x, y oder z auch leer sein.
- Ein **echtes Präfix** x von w ist ein Präfix mit $x \neq w$

Beispiel

- Der String $abab$ hat die
  - Präfixe ϵ, a, ab, aba, $abab$
  - Suffixe ϵ, b, ab, bab, $abab$
  - Teilstrings, ϵ, a, b, ab, ba, aba, bab, $abab$

Literaturhinweise

Umwandlung DFA $\longrightarrow$ RE: R. McNaughton and H. Yamada. Regular Expressions and State Graphs for Automata. *IEEE Transactions on Electronic Computers*, EC-9:39–47, 1960