Grundlagen der Theoretischen Informatik

3. Endliche Automaten (II)

28.04.2016

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Übersicht

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

Letzte Vorlesung

- Determinierte endliche Automaten (DEAs)

  Bespiele: reguläre Sprachen
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

Endlicher Automat: Informell

Ein endlicher Automat testet, ob ein gegebenes $w \in \Sigma^*$ in einer Sprache L liegt.

- **Lesekopf** erlaubt w zu lesen.
  Bewegt sich nur von links nach rechts.
- Endlich viele mögliche **interne Zustände**,
  immer einer davon ist der aktuelle Zustand
- Automat beginnt in einem **initialen Zustand**.
- Bei jedem gelesenen Buchstaben Übergang zu neuem aktuellen Zustand,
  in Abhängigkeit vom Buchstaben und dem alten Zustand
- Wenn am Ende von w ein **finaler Zustand** erreicht ist,
  ist w **akzeptiert** als Element von L, sonst nicht.
- Automat **stoppt** (auf jeden Fall) nach $|w|$ Schritten

Endlicher Automat: Modell eines einfachen Computers

Endlicher Automat: Computer mit begrenztem Speicher

- Kann vom Band nur lesen
  $\Rightarrow$ kein externer Speicher
- Speichert nur den aktuellen Zustand (≈ Programmzähler)
  $\Rightarrow$ stark begrenzter interner Speicher

Endlicher Automat: Darstellung als Graph

Darstellung als Graph

- ein **Knoten** für jeden möglichen **Zustand**,
- **Kanten** sind mit Buchstaben beschriftet: Sie beschreiben **Zustandsänderungen**.
- **Initiale** Zustände werden mit einem **Pfeil** gekennzeichnet,
- **finale Zustände** mit einem **doppelten Kreis**.

Endlicher Automat: Darstellung als Graph

Beispiel: Sprache $\{w \mid \#_a(w) \textit{ gerade}\} \subset \{a, b\}^*$

Der folgende endliche Automat erkennt die Sprache

$$\{w \mid \#_a(w) \text{ gerade}\} \qquad \text{über} \qquad \Sigma = \{a, b\}$$

der Wörter mit gerader Anzahl von „a“s

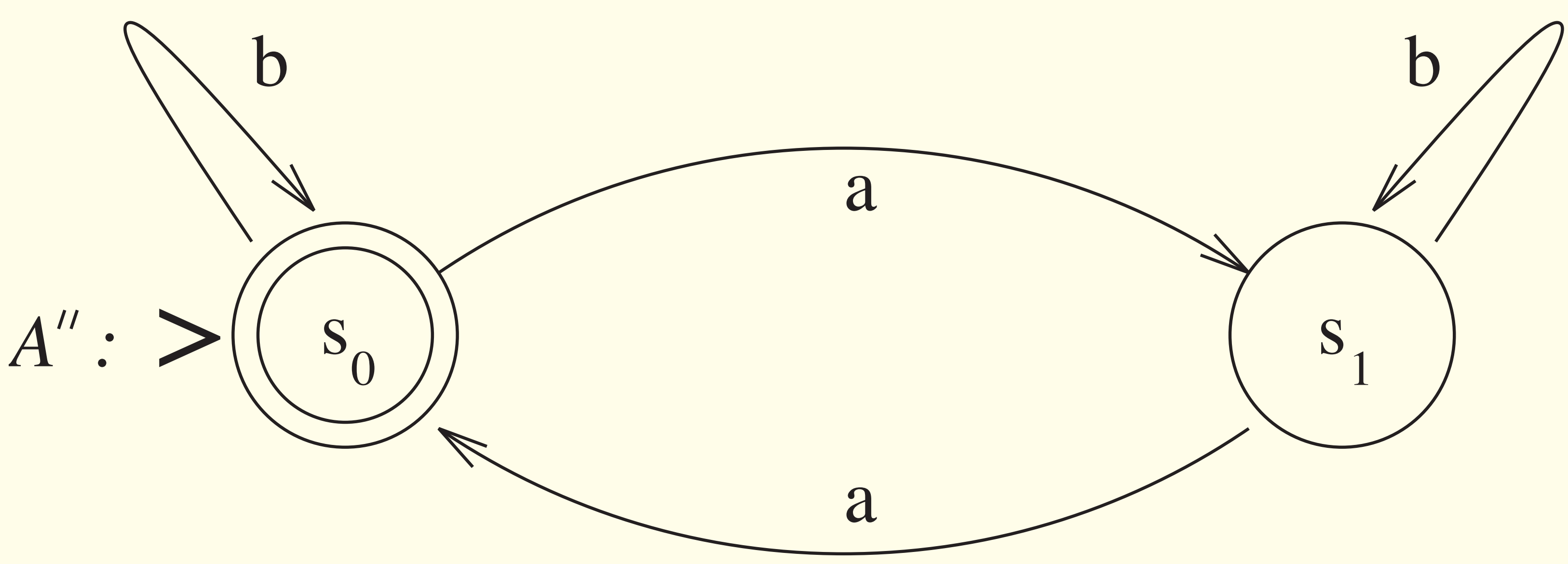
Endlicher Automat: Darstellung als Graph

Beispiel: Sprache $\{w \mid \#_a(w)\ \textit{gerade}\} \subset \{a, b\}^*$

Der folgende endliche Automat erkennt die Sprache

$$\{w \mid \#_a(w)\ \text{gerade}\} \qquad \text{über} \qquad \Sigma = \{a, b\}$$

der Wörter mit gerader Anzahl von „a“s



Endlicher Automat: Definition

Definition. Ein **endlicher Automat (e.a.) (finite automaton)** ist ein Tupel

$$\mathcal{A} = (K, \Sigma, \delta, s_0, F).$$

Dabei ist

- K eine endliche Menge von **Zuständen**,
- Σ ein **endliches Alphabet** (aus dessen Buchstaben die Eingabewörter bestehen können),
- $\delta : K \times \Sigma \to K$ die totale(!) **Übergangsfunktion**,
- $s_0 \in K$ der **Startzustand**, und
- $F \subseteq K$ die Menge der **finalen Zustände**.

Endlicher Automat: Übergangsfunktion

Bedeutung der Übergangsfunktion

$\delta(q, a) = q'$ bedeutet:

- Der Automat ist im Zustand q,
- liest ein a und
- geht in den Zustand q' über.

Endlicher Automat: Übergangsfunktion

Bedeutung der Übergangsfunktion

$\delta(q, a) = q'$ bedeutet:

- Der Automat ist im Zustand q,
- liest ein a und
- geht in den Zustand q' über.

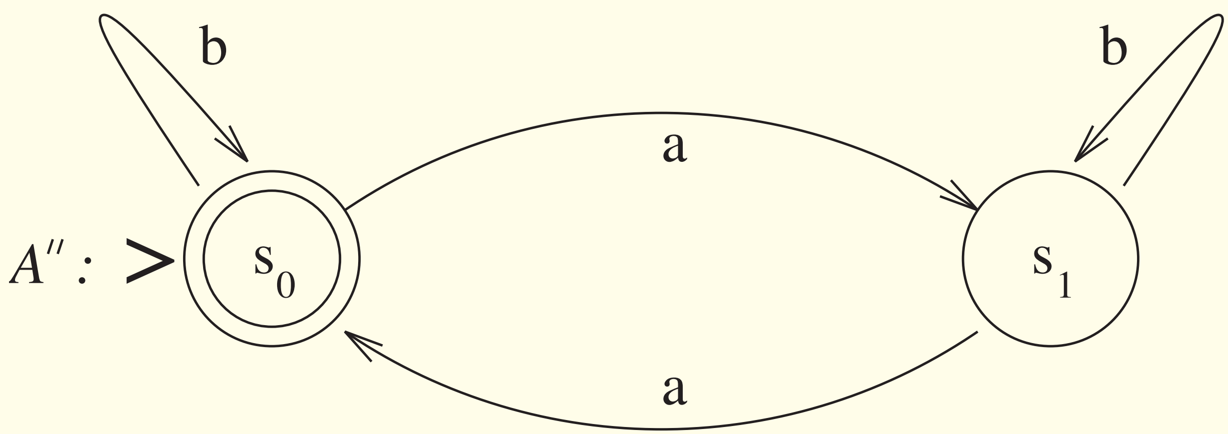
Wir erweitern δ zu δ^*

$\delta^* : K \times \Sigma^* \to K$ ist strukturell rekursiv über Σ^* definiert:

$$\begin{aligned} \delta^*(q, \varepsilon) \quad &:= \quad q \\ \delta^*(q, wa) \quad &:= \quad \delta(\delta^*(q, w), a) \end{aligned}$$

Wenn klar ist, was gemeint ist, wird δ^* auch einfach als δ geschrieben.

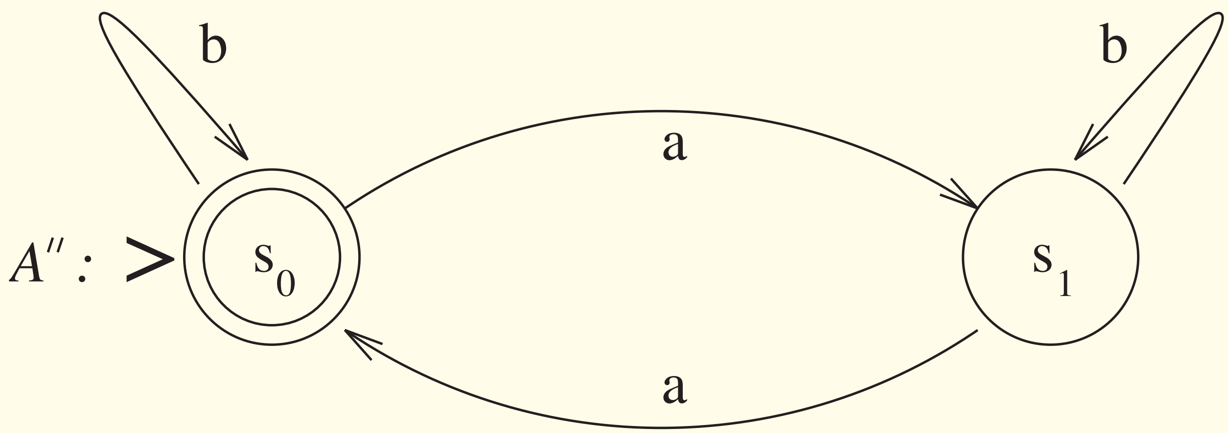
Endlicher Automat: Beispiel



Dieser Automat azeptiert die Sprache

$$\{w \mid \#_a(w) \text{ gerade}\} \subset \{a, b\}^* \text{ (Beispiel Seite 8)}$$

Endlicher Automat: Beispiel



Formal hat er die Form: $\mathcal{A} = (\{s_0, s_1\}, \{a, b\}, \delta, s_0, \{s_0\})$ mit

$$\delta(s_0, a) = s_1 \qquad \delta(s_1, a) = s_0$$

$$\delta(s_0, b) = s_0 \qquad \delta(s_1, b) = s_1$$

Endlicher Automat: Übergangsfunktion

Beispiel für δ^*

$$\begin{aligned}
\delta^*(s_0, aab) &= \delta\big(\delta^*(s_0, aa), b\big) \\
&= \delta\big(\delta\big(\delta^*(s_0, a), a\big), b\big) \\
&= \delta\big(\delta\big(\delta(\delta^*(s_0, \varepsilon), a\big), a\big), b\big) \\
&= \delta\big(\delta\big(\delta(s_0, a), a\big), b\big) \\
&= \delta\big(\delta(s_1, a), b\big) \\
&= \delta(s_0, b) \\
&= s_0
\end{aligned}$$

Endlicher Automat: Akzeptierte Sprache

Definition (Von einem endlichen Automaten akzeptierte Sprache)

Die von einem Automaten $\mathcal{A}$ **akzeptierte** Sprache, ist definiert als

$$L(\mathcal{A}) \;:=\; \{w \in \Sigma^* \;\mid\; \delta^*(s_0, w) \in F\}$$

Endlicher Automat: Akzeptierte Sprache

Definition (Von einem endlichen Automaten akzeptierte Sprache)
Die von einem Automaten $\mathcal{A}$ **akzeptierte** Sprache, ist definiert als

$$L(\mathcal{A}) \; := \; \{w \in \Sigma^* \mid \delta^*(s_0, w) \in F\}$$

Definition (Von endlichen Automaten akzeptierte Sprachen)
Die Menge

$$\textbf{RAT} \; := \; \{L \mid \text{es gibt einen endlichen Automaten } \mathcal{A} \text{ mit } L = L(\mathcal{A})\}$$

der von endlichen Automaten akzeptierten Sprachen
heißt Menge der **rationalen** Sprachen

Endlicher Automat: Akzeptierte Sprache

Definition (Von einem endlichen Automaten akzeptierte Sprache)
Die von einem Automaten $\mathcal{A}$ **akzeptierte** Sprache, ist definiert als

$$L(\mathcal{A}) \;:=\; \{w \in \Sigma^* \mid \delta^*(s_0, w) \in F\}$$

Definition (Von endlichen Automaten akzeptierte Sprachen)
Die Menge

$$\mathbf{RAT} \;:=\; \{L \mid \text{es gibt einen endlichen Automaten } \mathcal{A} \text{ mit } L = L(\mathcal{A})\}$$

der von endlichen Automaten akzeptierten Sprachen
heißt Menge der **rationalen** Sprachen

Wir zeigen demnächst: **RAT** $=$ Menge der regulären Sprachen

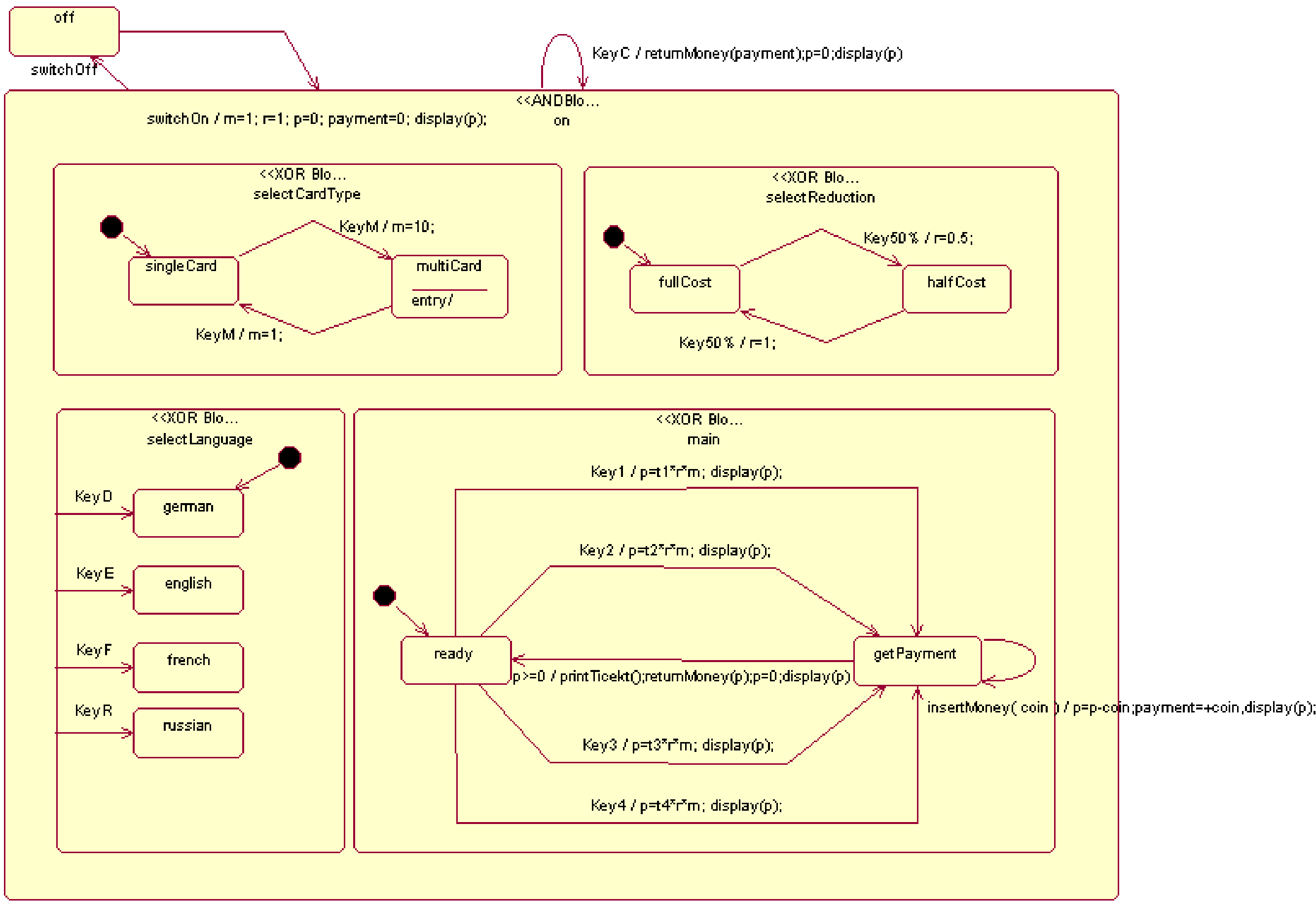
Endlicher Automat: UML State Chart

UML State Charts sind eine (erweiterte) Form endlicher Automaten

Endlicher Automat: UML State Chart

UML State Charts sind eine (erweiterte) Form endlicher Automaten

Beispiel:



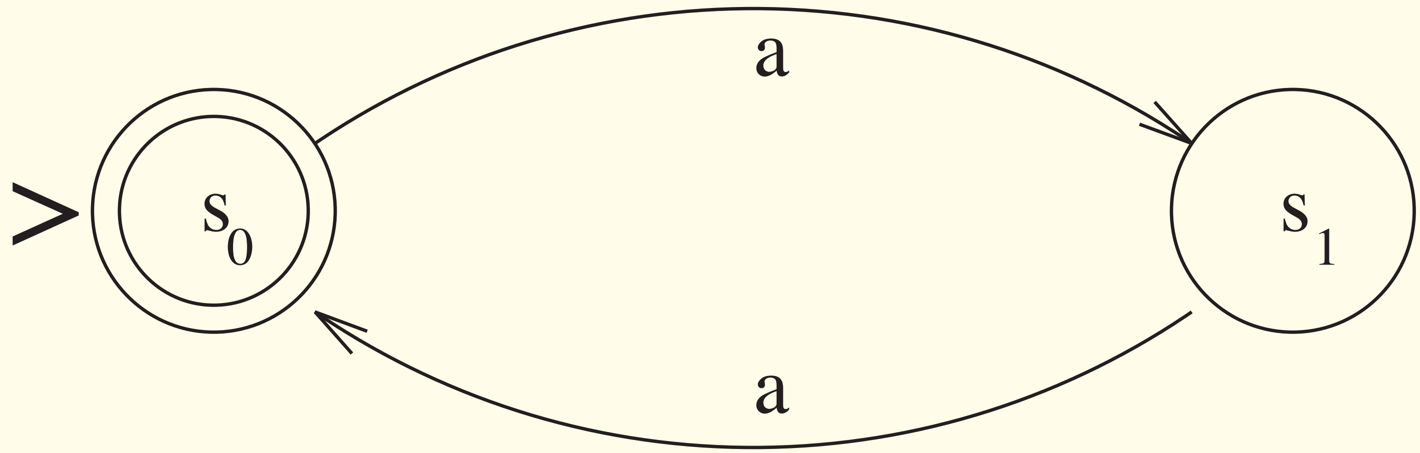
Endliche Automate: Weitere Beispiele

Beispiel. Die Sprache aller Wörter mit gerader Anzahl von a über dem (kleineren) Alphabet $\Sigma = \{a\}$ wird akzeptiert von

Endliche Automate: Weitere Beispiele

Beispiel. Die Sprache aller Wörter mit gerader Anzahl von a über dem (kleineren) Alphabet $\Sigma = \{a\}$ wird akzeptiert von:

A':



Endliche Automate: Weitere Beispiele

Beispiel. Die Sprache

$$L = \{w \in \{0,1\}^* \mid w \text{ enthält genau zwei Einsen}\}$$

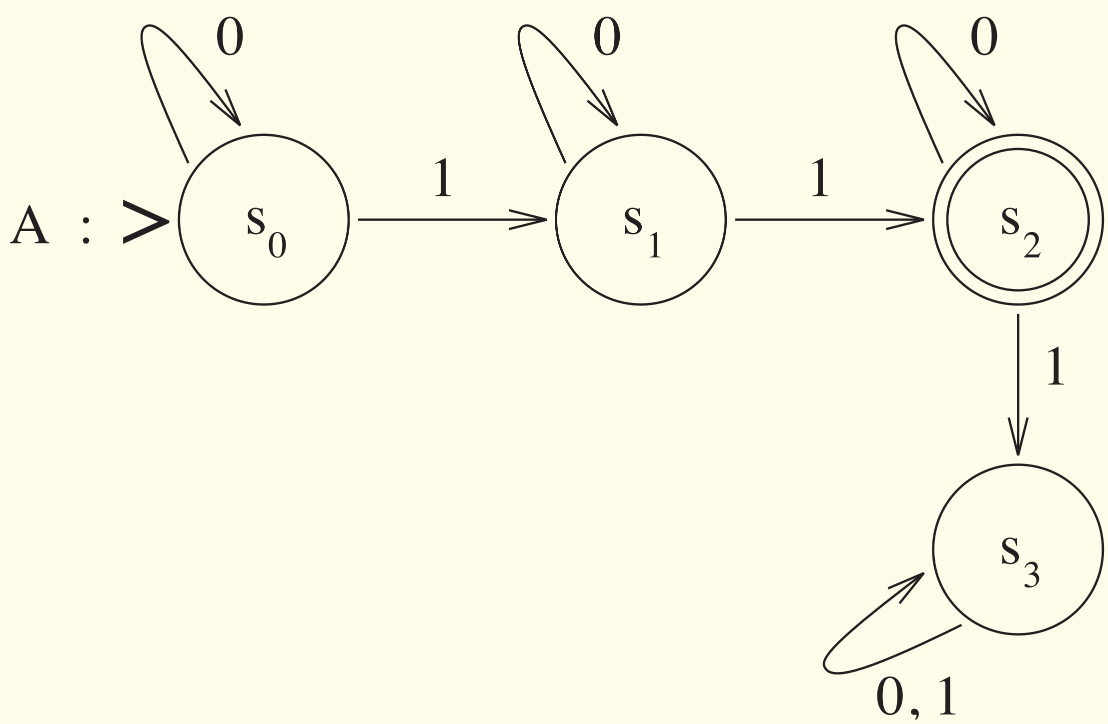
wird akzeptiert von dem folgenden endlichen Automaten:

Endliche Automate: Weitere Beispiele

Beispiel. Die Sprache

$$L = \{w \in \{0,1\}^* \mid w \text{ enthält genau zwei Einsen}\}$$

wird akzeptiert von dem folgenden endlichen Automaten:



Endliche Automate: Weitere Beispiele

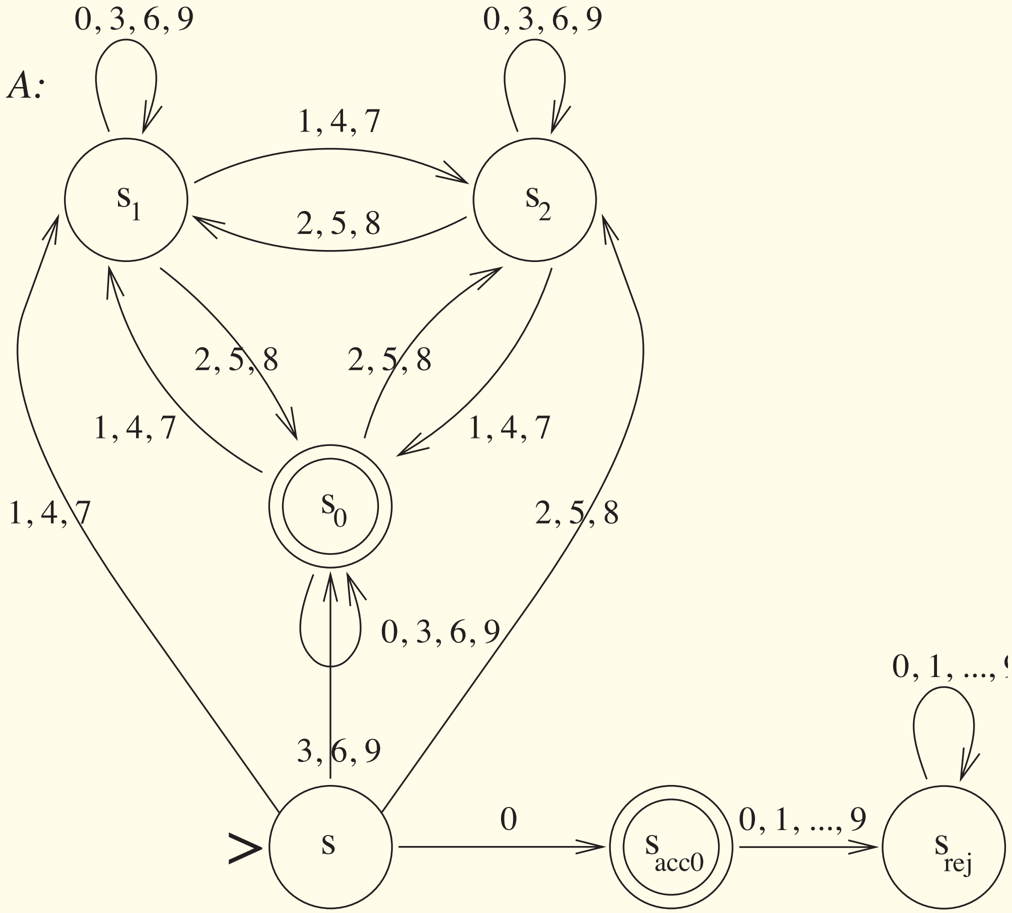
Beispiel

Die Sprache aller durch 3 teilbaren Dezimalzahlen wird akzeptiert durch:

Endliche Automate: Weitere Beispiele

Beispiel

Die Sprache aller durch 3 teilbaren Dezimalzahlen wird akzeptiert durch:



Bis jetzt

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational $=$ Reguläre Ausdrücke

Jetzt

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational $=$ Reguläre Ausdrücke

Indeterminierte endliche Automaten (NDEAs)

Determiniert / indeterminiert

Determinierter endlicher Automat

- Für einen Zustand q und eine Eingabe a
  genau ein einziger Nachfolgezustand
- festgelegt durch Übergangsfunktion δ

Indeterminierter endlicher Automat

- Für einen Zustand q und eine Eingabe a
  evtl. **mehrere Nachfolgezustände** – oder **gar keiner**
- festgelegt durch Übergangsrelation Δ

Indeterminierter endlicher Automat

Definition (Indeterminierter endlicher Automat)
Ein **indeterminierter** endlicher Automat (NDEA) ist ein Tupel

$$\mathcal{A} = (K, \Sigma, \Delta, I, F)$$

Dabei ist

- K eine endliche Menge von Zuständen,
- Σ ein endliches Alphabet,
- $\Delta \subseteq (K \times \Sigma) \times K$ eine Übergangsrelation,
- $I \subseteq K$ eine **Menge von** Startzuständen,
- $F \subseteq K$ eine Menge von finalen Zuständen.

NDEA: Übergangsrelation

Definition (Erweiterung von Δ zu Δ^*)

$$\Delta^* \subseteq (K \times \Sigma^*) \times K$$

ist definiert durch:

$$\Delta^*(\ (q, \varepsilon),\ q'\ ) \quad \underline{\text{gdw}} \quad q' = q$$

$$\Delta^*(\ (q, wa),\ q'\ ) \quad \underline{\text{gdw}} \quad \exists q'' \in K\ \big(\Delta^*(\ (q, w),\ q''\ )\ \wedge\ \Delta(\ (q'', a),\ q'\ )\big)$$

NDEA: Akzeptierte Sprache

Wann akzeptiert ein indeterminierter Automat ein Wort?

Ein indeterminierter endlicher Automat $\mathcal{A}$ akzeptiert ein Wort w, wenn

- es **mindestens einen** Weg mit der Beschriftung w durch $\mathcal{A}$ gibt,
- der in einem finalen Zustand endet.

NDEA: Akzeptierte Sprache

Wann akzeptiert ein indeterminierter Automat ein Wort?

Ein indeterminierter endlicher Automat $\mathcal{A}$ akzeptiert ein Wort w, wenn

- es **mindestens einen** Weg mit der Beschriftung w durch $\mathcal{A}$ gibt,
- der in einem finalen Zustand endet.

Definition (Von einem NDEA akzeptierte Sprache)
Die von einem indeterminierten endlichen Automaten $\mathcal{A}$ **akzeptierte** Sprache ist

$$L(\mathcal{A}) \quad := \quad \{w \in \Sigma^* \mid \exists s_0 \in I \; \exists q \in F \; : \Delta^*(\,(s_0, w),\; q\,)\}$$

NDEA: Beispiel

Der Automat

$$\mathcal{A} = (\{S_0, S_1, S_2\}, \{a, b\}, \Delta, \{S_0\}, \{S_0\})$$

mit

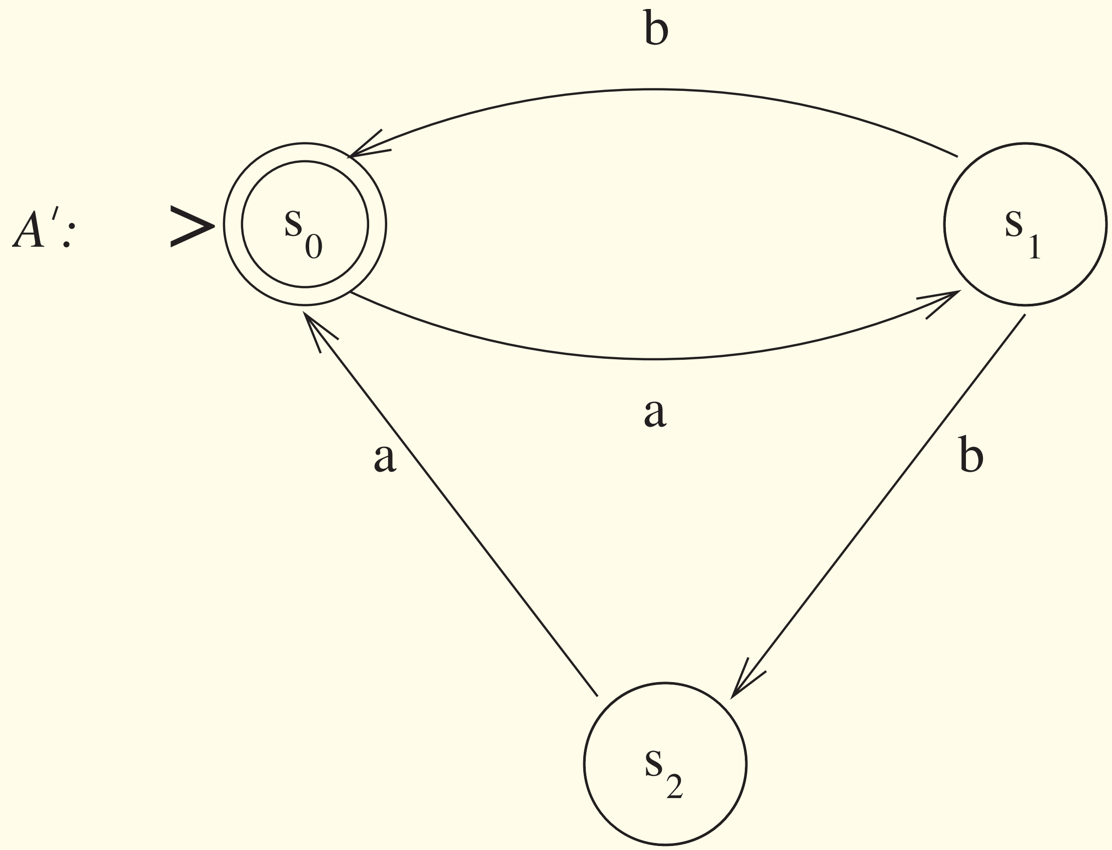
$$\begin{aligned}
\Delta(S_0, a) &= \{S_1\} \\
\Delta(S_1, b) &= \{S_0, S_2\} \\
\Delta(S_2, a) &= \{S_0\}
\end{aligned}$$

akzeptiert die Sprache

$$L = \{ab, aba\}^*$$

NDEA: Graphische Darstellung

Der indeterminierte Automat für Beispiel auf Seite 34



Akzeptiert: $\{ab, aba\}^*$

Indeterminierter endlicher Automat

Vom indeterminierten Automaten zum Algorithmus?

Vom Automaten zum Algorithmus (für das Wortproblem):

- **DEA** $=$ **Algorithmus**
- **NDEA** $+$ **Suchstrategie** $=$ **Algorithmus**

Indeterminierter endlicher Automat

Vom indeterminierten Automaten zum Algorithmus?

Vom Automaten zum Algorithmus (für das Wortproblem):

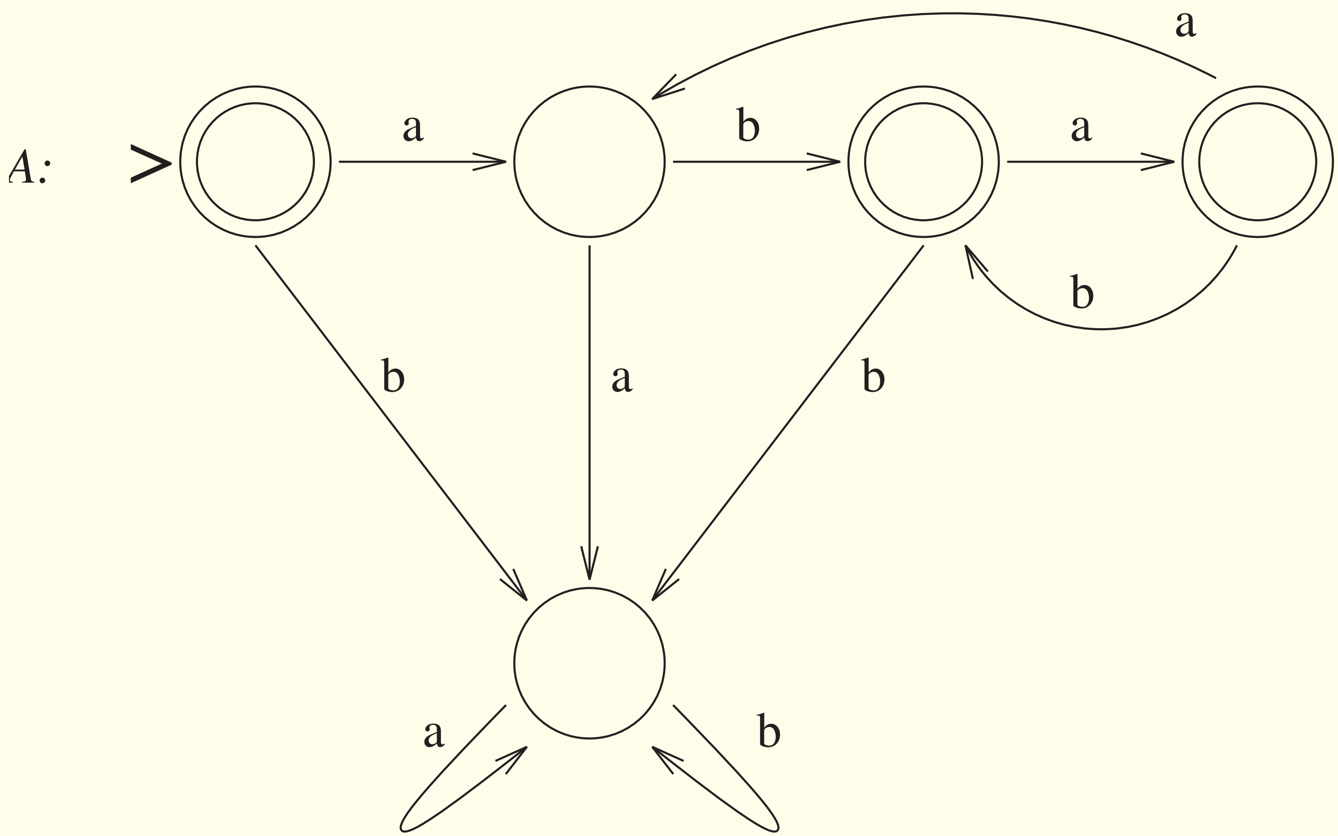
- **DEA** $=$ **Algorithmus**
- **NDEA** $+$ **Suchstrategie** $=$ **Algorithmus**

Zwei Sichtweisen auf indeterminierte Automaten

- Der Automat durchläuft **alle** Wege (**parallel** oder mittels **Backtracking**)
- Der Automat **rät**, welcher von mehreren möglichen Folgezuständen der richtige ist

NDEA und DEA: Beispiel

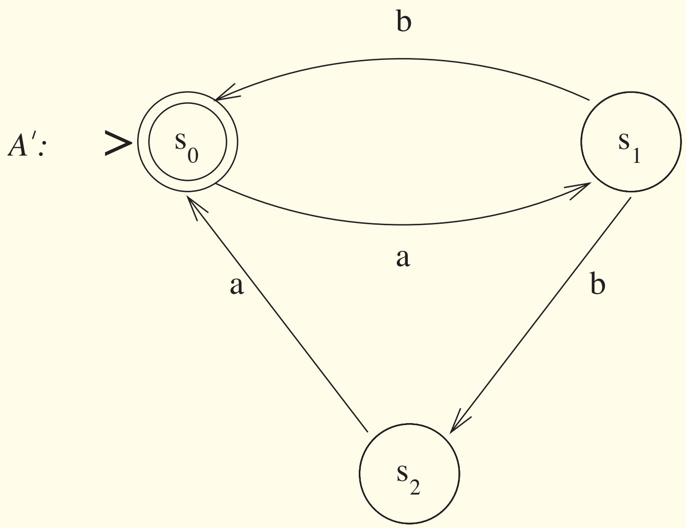
DEA für gleiche Sprache wie NDEA aus Seite 35



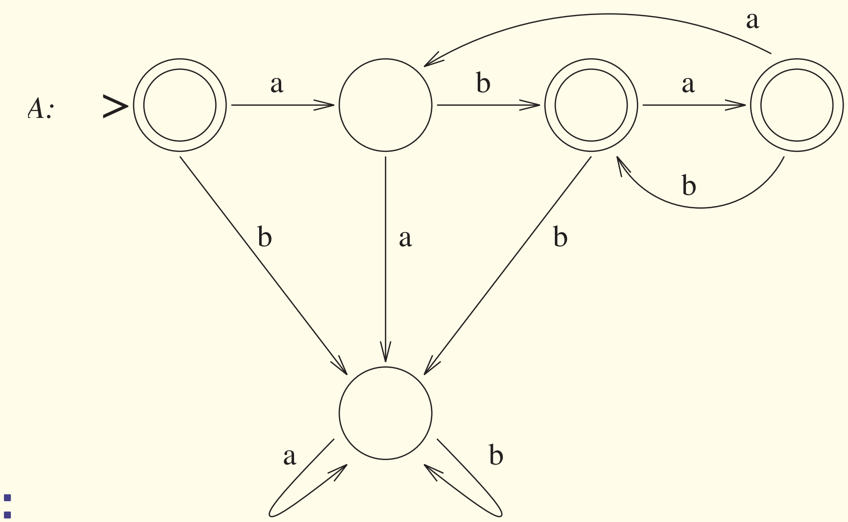
NDEA und DEA

Vergleich NDEA / DEA

NDEA:



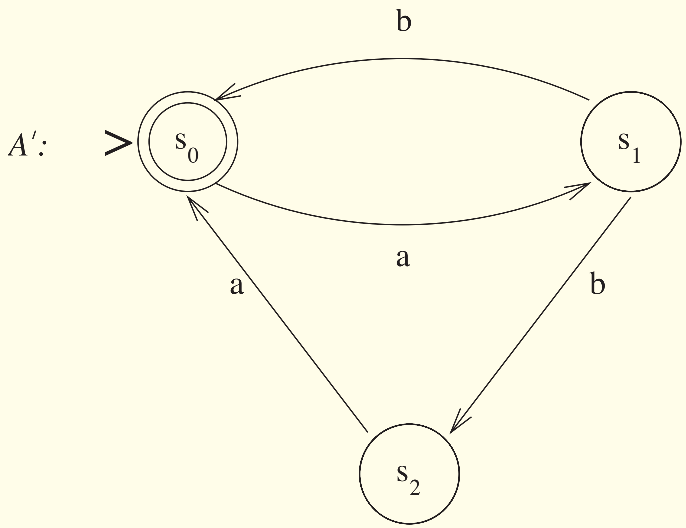
DEA:



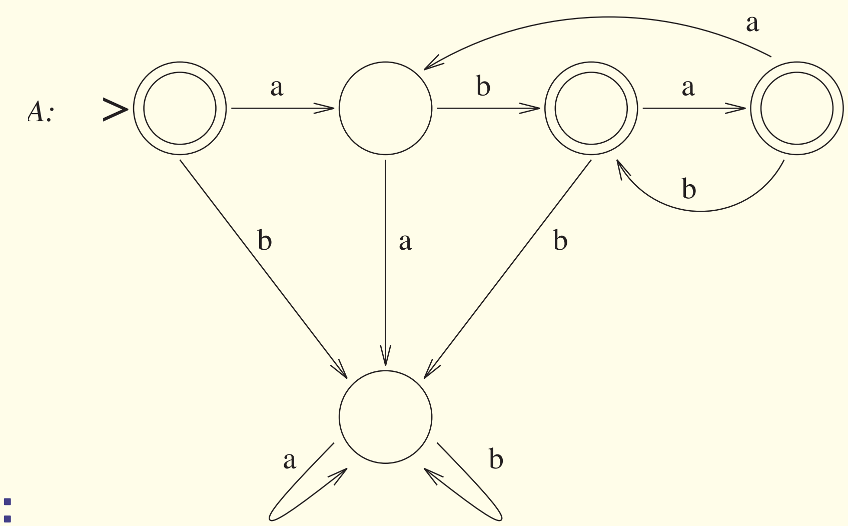
NDEA und DEA

Vergleich NDEA / DEA

NDEA:



DEA:



- DEA hat mehr Zustände, komplizierter
- DEA muss nicht „raten“
- DEA braucht genauso viele Schritte

NDEA und DEA

Wir zeigen später:

Für jeden indeterminierten Automaten A_{NDEA}
gibt es einen determinierten Automaten A_{DEA} mit

$$L(A_{\mathrm{NDEA}}) \;=\; L(A_{\mathrm{DEA}})$$

NDEA und DEA: Weiteres Beispiel

Determinierter Automat für die Sprache

$$L = \{a, b\}^*\{a\}\{a, b\}$$

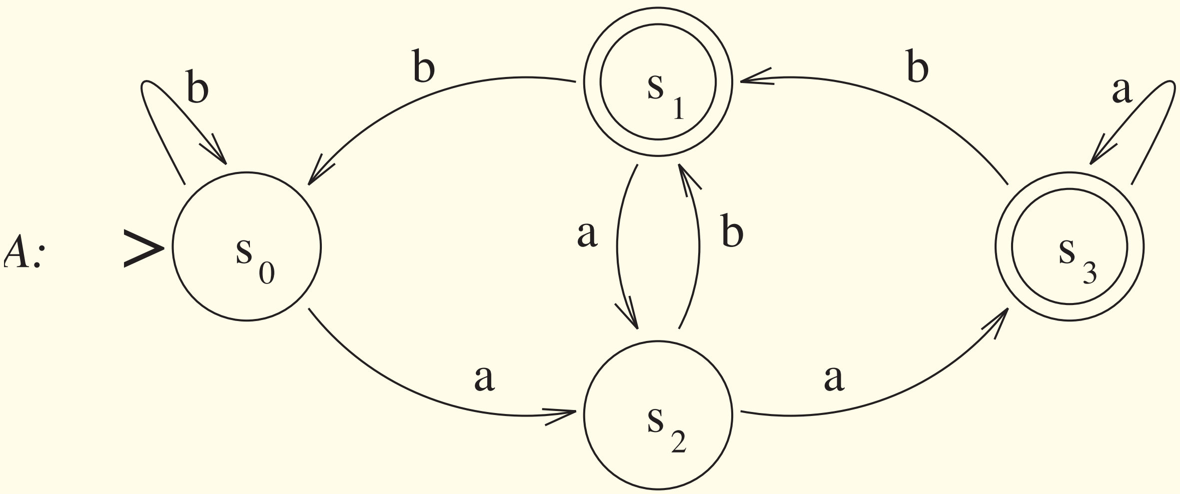
(die Sprache aller Wörter über $\{a, b\}$, deren zweitletzter Buchstabe a ist)

NDEA und DEA: Weiteres Beispiel

Determinierter Automat für die Sprache

$$L = \{a, b\}^*\{a\}\{a, b\}$$

(die Sprache aller Wörter über $\{a, b\}$, deren zweitletzter Buchstabe a ist)



Idee: Im Zustand jeweils die letzten zwei Buchstaben merken

NDEA und DEA: Weiteres Beispiel

Indeterminierter Automat für die Sprache

$$L = \{a, b\}^*\{a\}\{a, b\}$$

(die Sprache aller Wörter über $\{a, b\}$, deren zweitletzter Buchstabe a ist)

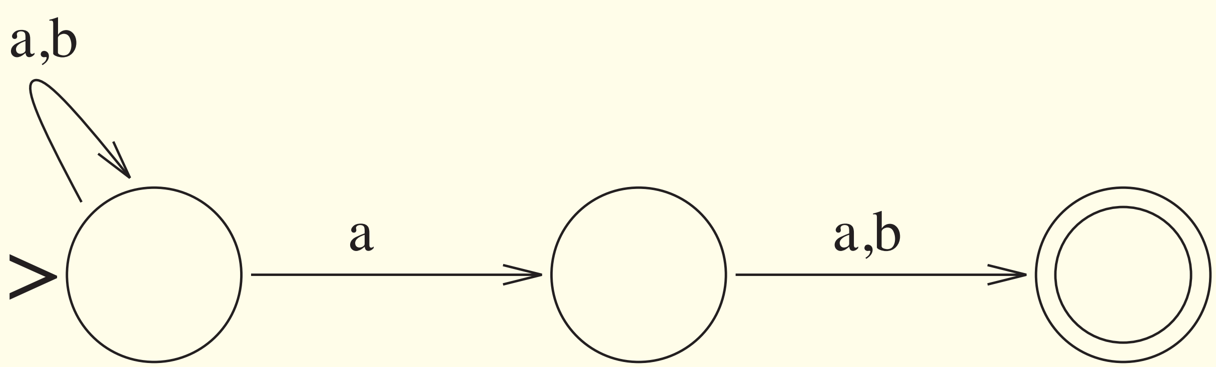
NDEA und DEA: Weiteres Beispiel

Indeterminierter Automat für die Sprache

$$L = \{a, b\}^*\{a\}\{a, b\}$$

(die Sprache aller Wörter über $\{a, b\}$, deren zweitletzter Buchstabe a ist)

A':



NDEA und DEA: Größenvergleich

Größenvergleich (Worst case)

Sprache über $\{a, b\}$ der Wörter, deren nt-letzter Buchstabe ein a ist

Determinierter Automat: $\mathbf{2^n}$ Zustände

(einen für jede Buchstabenkombination der Länge n)

Indeterminierter Automat: $\mathbf{n + 1}$ Zustände

Gleichmächtigkeit von DEA und NDEA

Theorem. DEA gleich mächtig wie NDEA

Eine Sprache ist rational

(es gibt einen determinierten endlichen Automaten, der sie akzeptiert)

gdw

es gibt einen indeterminierten endlichen Automaten, der sie akzeptiert.

Gleichmächtigkeit von DEA und NDEA

Theorem. DEA gleich mächtig wie NDEA
Eine Sprache ist rational
(es gibt einen determinierten endlichen Automaten, der sie akzeptiert)

gdw

es gibt einen indeterminierten endlichen Automaten, der sie akzeptiert.

Beweis. **„⇒“:**

- Sei L eine rationale Sprache.
- Dann gibt es laut Definition einen determinierten endlichen Automaten $\mathcal{A}_{\text{DEA}}$ mit $L = L(\mathcal{A}_{\text{DEA}})$.
- Jeder determinierte endliche Automat ist aber insbesondere auch ein (besonderer) indeterminierter endlicher Automat.

Gleichmächtigkeit von DEA und NDEA

Theorem. DEA gleich mächtig wie NDEA

Eine Sprache ist rational

(es gibt einen determinierten endlichen Automaten, der sie akzeptiert)

gdw

es gibt einen indeterminierten endlichen Automaten, der sie akzeptiert.

Beweis „$\Leftarrow$“:

Sei $\mathcal{A}_{\text{NDEA}} = (K, \Sigma, \Delta, I, F)$ ein (beliebiger) indeterminierter endlicher Automat. Er akzeptiert die Sprache $L(\mathcal{A}_{\text{NDEA}})$.

Beweisidee: Konstruiere aus $\mathcal{A}_{\text{NDEA}}$ einen determinierten Automaten $\mathcal{A}_{\text{DEA}}$ mit

$$L(\mathcal{A}_{\text{NDEA}}) = L(\mathcal{A}_{\text{DEA}})$$

mit Hilfe einer Potenzmengenkonstruktion ...

Gleichmächtigkeit von DEA und NDEA

Beweis (Forsetzung)

Fortsetzung ...

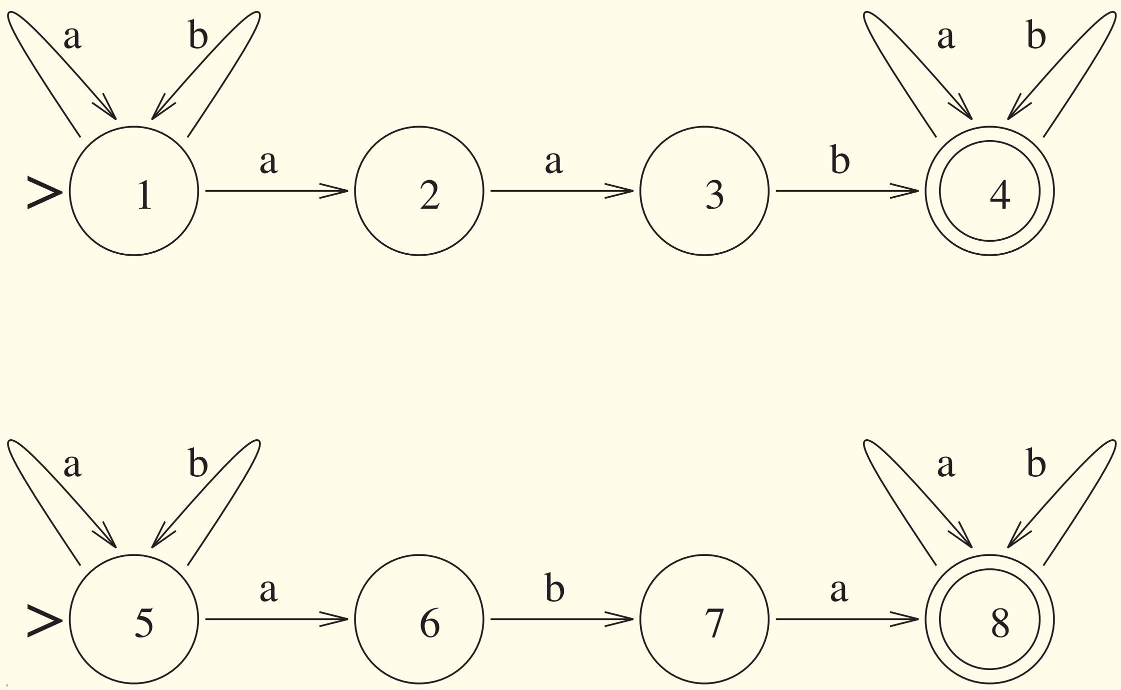
- Zustände in $\mathcal{A}_{\text{DEA}}$ bestehen aus Mengen von Zuständen von $\mathcal{A}_{\text{NDEA}}$
- Wenn man in $\mathcal{A}_{\text{NDEA}}$ mit w nach $q_1, \ldots, q_n$ gelangt,
  dann gelangt man in $\mathcal{A}_{\text{DEA}}$ mit w nach $q' = \{q_1, \ldots, q_n\}$.
- Initialer Zustand von $\mathcal{A}_{\text{DEA}}$:
  Menge aller initialen Zustände von $\mathcal{A}_{\text{NDEA}}$
- Finale Zustände von $\mathcal{A}_{\text{DEA}}$:
  Jede Menge von Zustände, die einen finalen Zustand von $\mathcal{A}_{\text{NDEA}}$ enthält

Gleichmächtigkeit von DEA und NDEA

Zunächst ein konkretes Beispiel

Ein NDEA, der die folgende Sprache akzeptiert:

$$L_{aab/aba} \; = \; \{w \in \{a, b\}^* \;\mid\; w \text{ hat } aab \text{ oder } aba \text{ als Teilwort}\}$$



Gleichmächtigkeit von DEA und NDEA

Zunächst ein konkretes Beispiel

Startzustand:
Menge der alten Startzustände, also $\{1, 5\}$.

Nächster Schritt:
Übergang von $\{1, 5\}$ mit a

$$\begin{aligned} \Delta(1, a) &= \{1, 2\} \\ \Delta(5, a) &= \{5, 6\} \end{aligned}$$

Also, neuer Zustand $\{1, 2, 5, 6\}$ mit

$$\delta_{\mathcal{A}_{\mathrm{DEA}}}(\{1, 5\}, a) = \{1, 2, 5, 6\}$$

Gleichmächtigkeit von DEA und NDEA

Zunächst ein konkretes Beispiel

Nächster Schritt:
Übergang von $\{1, 5\}$ mit b.

$$\begin{aligned}\Delta(1, b) &= \{1\} \\ \Delta(5, b) &= \{5\}\end{aligned}$$

Für den Eingabebuchstaben b bleibt $\mathcal{A}_{\mathsf{DEA}}$ also im Startzustand.

Gleichmächtigkeit von DEA und NDEA

Zunächst ein konkretes Beispiel

Nächster Schritt:

Übergang von $\{1,2,5,6\}$ mit a

$$\begin{aligned}\Delta(1,a) &= \{1,2\}\\ \Delta(2,a) &= \{3\}\\ \Delta(5,a) &= \{5,6\}\\ \Delta(6,a) &= \emptyset\end{aligned}$$

Also, neuer Zustand $\{1,2,3,5,6\}$ mit

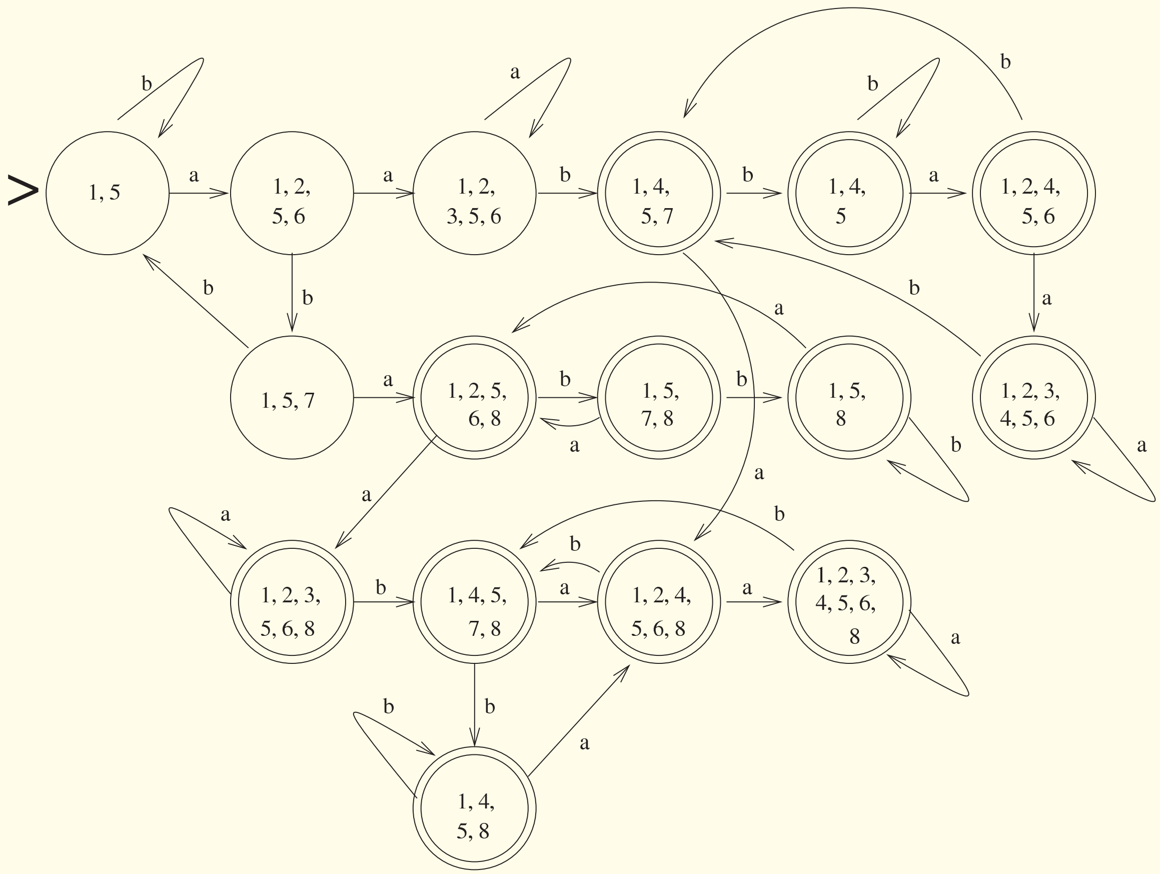
$$\delta_{\mathcal{A}_{\text{DEA}}}(\{1,2,5,6\},a) = \{1,2,3,5,6\}$$

usw.

Gleichmächtigkeit von DEA und NDEA

Zunächst ein konkretes Beispiel

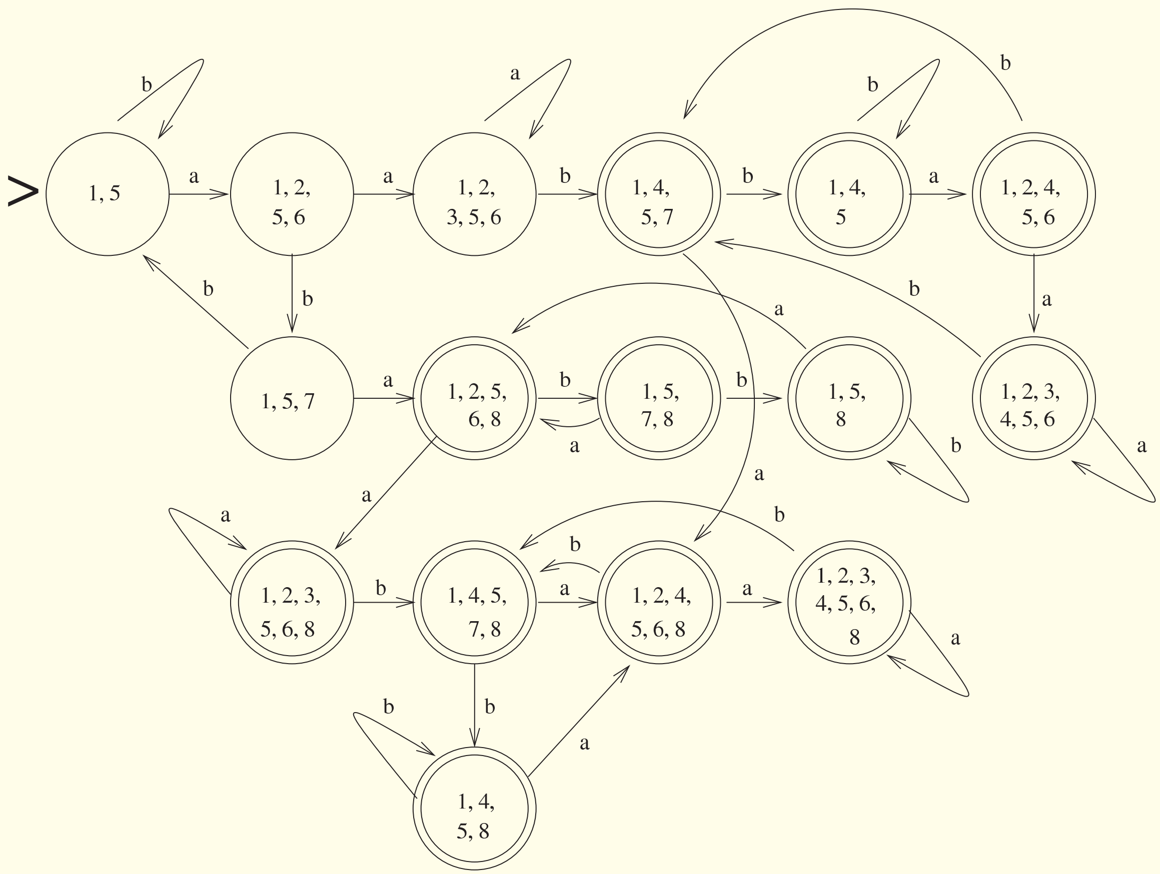
Es ergibt sich folgender determinierter Automat $\mathcal{A}_{\mathsf{DEA}}$:



Gleichmächtigkeit von DEA und NDEA

Zunächst ein konkretes Beispiel

Es ergibt sich folgender determinierter Automat $\mathcal{A}_{\text{DEA}}$:



Übergänge aller finalen Zustände führen in finale Zustände.

Vereinfachung: Man könnte die finalen Zustände zu einem zusammenfassen