Grundbegriffe der Informatik

Kapitel 14: Endliche Automaten

Mattias Ulbrich
(basierend auf Folien von Thomas Worsch)

KIT · Institut für Theoretische Informatik

Wintersemester 2023/2024

Überblick

Erstes Beispiel: ein Kasten mit Vokabelkärtchen

Moore-Automaten

Spezialfall: endliche Akzeptoren

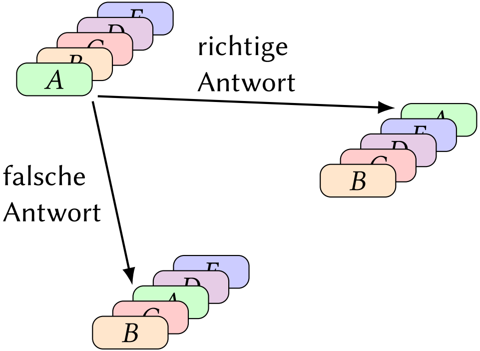
Wo sind wir?

Erstes Beispiel: ein Kasten mit Vokabelkärtchen

Moore-Automaten

Spezialfall: endliche Akzeptoren

Ein Kasten mit Vokabelkärtchen



- Vokabelkarten, zum Beispiel 5 Stück
  - A
  - B
  - C
  - D
  - E

  in einem Kasten
- die vorderste ist «als nächstes dran»
  - Übersetzung gewusst:
    Karte wird nach ganz hinten gesteckt
  - Übersetzung vergessen:
    Karte wird in die Mitte gesteckt

Vokabelkasten: Zustände

Karlsruher Institut für Technologie

- nach jeder Antwort wird
  eine neue Reihenfolge gespeichert
- Formalisierung: Quintupel $x_0x_1x_2x_3x_4$
  mit paarweise verschiedenen Karten
  - jedes $x_i \in \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$ und
  - wenn $i \neq j$, dann $x_i \neq x_j$
  - also Bijektionen $\mathbb{Z}_5 \longrightarrow \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$
- Zustandsmenge $Z = \{\mathtt{ABCDE}, \mathtt{ABCED}, \dots\}$
  enthält $5! = 120$ Listen/Wörter über $\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$

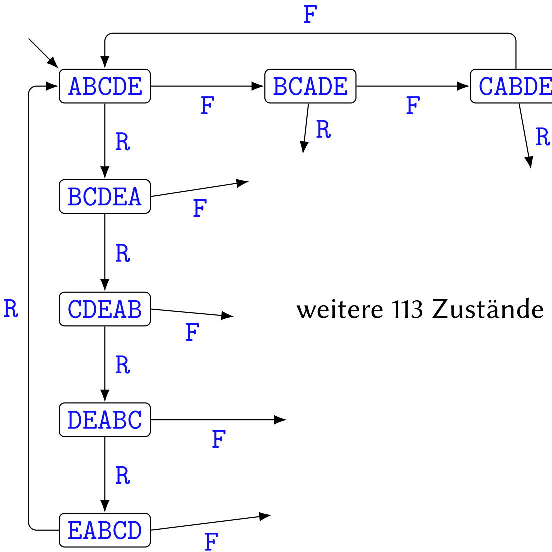
Vokabelkasten: Eingaben

- Eingaben führen zu *Zustandsänderungen.*
- Eingaben hier:
  - richtige oder falsche Antworten
- Modellierung der Eingaben:
  - Symbol $\mathtt{R}$ für *richtige* Antwort
  - Symbol $\mathtt{F}$ für *falsche* Antwort
  - Modellannahme: nie zwei Eingaben gleichzeitig
- Eingabealphabet $X = \{\mathtt{R}, \mathtt{F}\}$

Vokabelkasten: Zustandsübergänge (1)

- Zustandsübergang hängt ab von
  - aktuellem Zustand $z \in Z$ und
    aktuellem Eingabesymbol $x \in X$
  - z und x legen eindeutig den neuen Zustand fest
    - also *immer* und *eindeutig*
    - (zumindest in diesem Kapitel)
  - Zustandsüberführungsfunktion $f : Z \times X \to Z$
  - oft graphisch spezifiziert

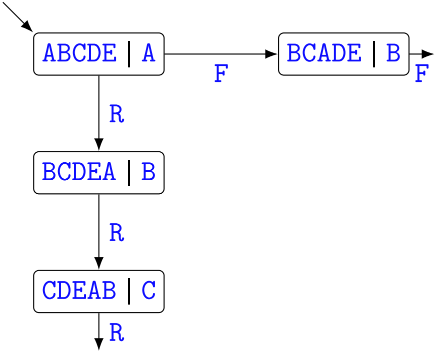
Vokabelkasten: Zustandsübergänge (2)



Vokabelkasten: Ausgaben

- zweiter wesentlicher Aspekt jedes Automaten: Ausgaben
- hier: Wort auf dem nächsten Kärtchen
- Ausgabealphabet Y
- Formalisierung der Ausgaben
  - abhängig von aktuellem Zustand $z \in Z$
  - z legt immer eindeutig die Ausgabe fest
    - im allgemeinen *Wörter* über Y
  - Formalisierung: Ausgabefunktion $h : Z \longrightarrow Y^*$
    - Funktionswert ε: «keine Ausgabe»
- h im Diagramm: Zustand mit $z|h(z)$

Vokabelkasten: Ausgaben im Zustandsdiagramm



- $Y = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$
- $h(x_0x_1x_2x_3x_4) = x_0$

Was ist wichtig

- Das sollten Sie mitnehmen:
  - alles endlich
  - alles endlich beschreibbar
  - im Beispiel Vokabelkasten:
    - aktueller Zustand und aktuelle Eingabe legen eindeutig nächsten Zustand fest
    - aktueller Zustand legt Ausgabe fest
- Das sollten Sie üben:
  - Zustandsdiagramm lesen

Wo sind wir?

Erstes Beispiel: ein Kasten mit Vokabelkärtchen

Moore-Automaten

Spezialfall: endliche Akzeptoren

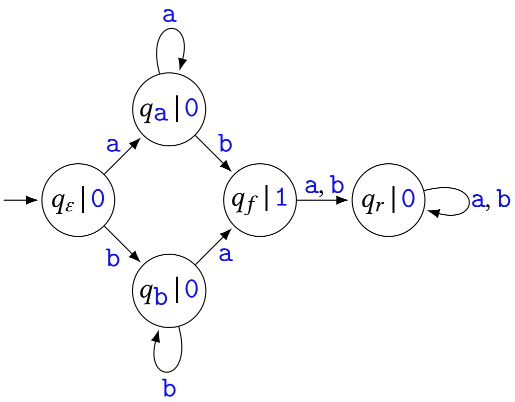
Moore-Automat

Definition

- endliche *Zustandsmenge* Z,
- *Anfangszustand* $z_0 \in Z$,
- *Eingabealphabet* X,
- *Zustandsüberführungsfunktion* $f : Z \times X \to Z$,
- *Ausgabealphabet* Y,
- *Ausgabefunktion* $h : Z \to Y^*$

Moore-Automat: Beispiel aus `tikz`-Dokumentation

Karlsruher Institut für Technologie



- mit Eingabesymbolen beschriftete Kanten für Zustandsübergänge
- Ausgaben «in den Zuständen»

Verallgemeinerte Zustandsübergangsfunktionen (1)

- nach Eingabe eines ganzen Wortes $w \in X^*$:
  - erreichter Zustand?
  - alle durchlaufenen Zustände?
- definiere passende Funktionen f_* und f_{**}
- f_* für am Ende erreichten Zustand:

Verallgemeinerte Zustandsübergangsfunktionen (1)

- nach Eingabe eines ganzen Wortes $w \in X^*$:
  - erreichter Zustand?
  - alle durchlaufenen Zustände?
- definiere passende Funktionen f_* und f_{**}
- f_* für am Ende erreichten Zustand:
  - $$\begin{aligned} f_* : Z \times X^* \longrightarrow Z : \quad & f_*(z, \varepsilon) = z \\ \forall w \in X^* : \forall x \in X : \quad & f_*(z, wx) = f(f_*(z, w), x) \end{aligned}$$

Verallgemeinerte Zustandsübergangsfunktionen (1)

- nach Eingabe eines ganzen Wortes $w \in X^*$:
  - erreichter Zustand?
  - alle durchlaufenen Zustände?
- definiere passende Funktionen f_* und f_{**}
- f_* für am Ende erreichten Zustand:
  - $$\begin{aligned} f_* : Z \times X^* \to Z : \quad & f_*(z, \varepsilon) = z \\ \forall w \in X^* : \forall x \in X : \quad & f_*(z, wx) = f(f_*(z, w), x) \end{aligned}$$
  - $$\begin{aligned} \text{alternativ} \quad & \bar{f}_*(z, \varepsilon) = z \\ \forall w \in X^* : \forall x \in X : \quad & \bar{f}_*(z, xw) = \bar{f}_*(f(z, x), w) \end{aligned}$$
  - Definitionen äquivalent: $f_* = \bar{f}_*$

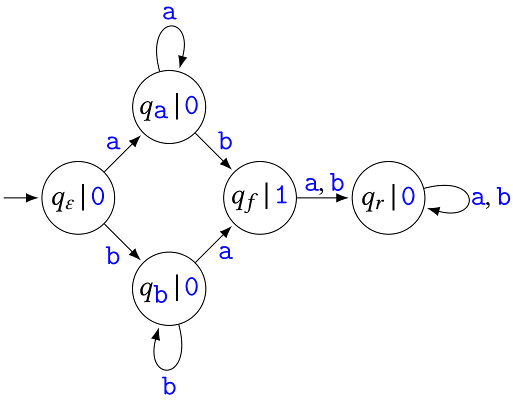
Verallgemeinerte Zustandsübergangsfunktionen (2)

- f_{**} für alle durchlaufenen Zustände:
  - $f_{**} : Z \times X^* \to Z^*$

$$\begin{aligned} f_{**}(z, \varepsilon) &= z \\ \forall w \in X^* : \forall x \in X : \quad f_{**}(z, wx) &= f_{**}(z, w) \cdot f_*(z, wx) \end{aligned}$$

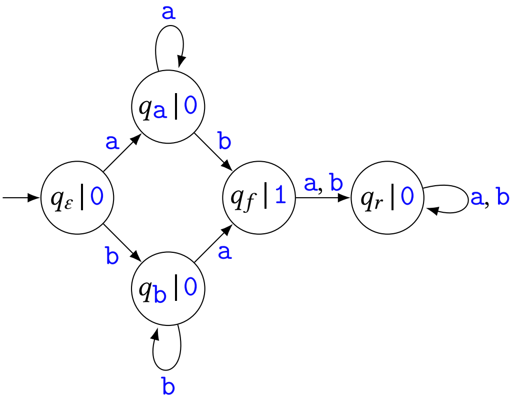
  - auch hier wieder eine alternative Definitionsmöglichkeit

Verallgemeinerte Zustandsübergangsfunktionen



- Definition von f_*, f_{**}
- Beispiele:
  - $f_*(q_\varepsilon, \texttt{aaaba}) = q_r$
  - $f_{**}(q_\varepsilon, \texttt{aaaba}) = q_\varepsilon q_\texttt{a} q_\texttt{a} q_\texttt{a} q_f q_r$

Verallgemeinerte Ausgabefunktionen g_* und g_{**}



- «letzte Ausgabe» $g_* = h \circ f_*$:

$$\forall (z, w) \in Z \times X^* : g_*(z, w) = h(f_*(z, w))$$

- «alle Ausgaben»: $g_{**} = h^{**} \circ f_{**}$ (ind. Hom.)

$$\forall (z, w) \in Z \times X^* : g_{**}(z, w) = h^{**}(f_{**}(z, w))$$

- Beispiel

$f_{**}(q_\varepsilon, \mathtt{aaaba}) = q_\varepsilon q_\mathtt{a} q_\mathtt{a} q_\mathtt{a} q_f q_r$ also

$g_*(q_\varepsilon, \mathtt{aaaba}) = h(f_*(q_\varepsilon, \mathtt{aaaba})) = h(q_r) = \mathtt{0}$

$$\begin{aligned} g_{**}(q_\varepsilon, \mathtt{aaaba}) &= h^{**}(f_{**}(q_\varepsilon, \mathtt{aaaba})) \\ &= h^{**}(q_\varepsilon q_\mathtt{a} q_\mathtt{a} q_\mathtt{a} q_f q_r) \\ &= h(q_\varepsilon)h(q_\mathtt{a})h(q_\mathtt{a})h(q_\mathtt{a})h(q_f)h(q_r) \\ &= \mathtt{000010} \end{aligned}$$

Was ist wichtig

- Das sollten Sie mitnehmen:
  - «offizielle» Definition von Moore-Automat:
    - Z
    - $z_0 \in Z$
    - X
    - $f : Z \times X \to Z$
    - Y
    - $h : Z \to Y^*$ Ausgabe hängt nur vom Zustand ab
- Das sollten Sie üben:
  - vorgegebenes Verhalten mit Moore-Automat realisieren
  - vorgebenenen Moore-Automaten analysieren

Wo sind wir?

Erstes Beispiel: ein Kasten mit Vokabelkärtchen

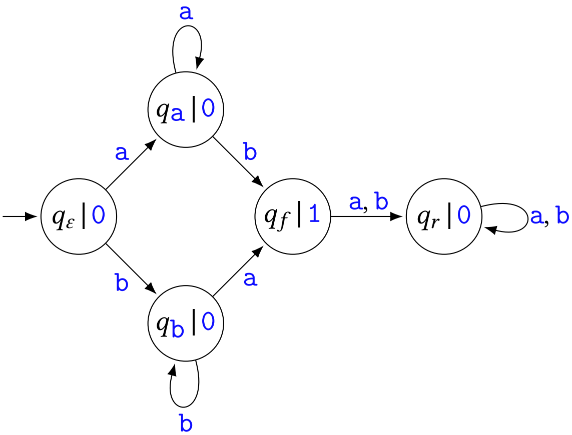
Moore-Automaten

Spezialfall: endliche Akzeptoren

Endliche Akzeptoren — ein wichtiger Sonderfall von Moore-Automaten

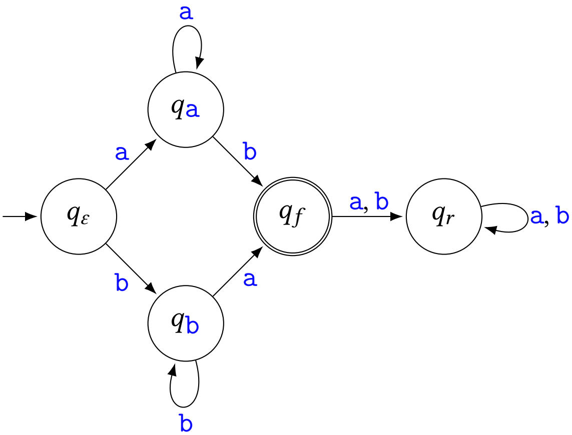
- immer genau ein Bit Ausgabe
  - $\forall z \in Z : h(z) \in Y = \{0, 1\}$
- Interpretation der Ausgabe:
  - Eingabe war «gut» oder «schlecht» bzw.
  - «syntaktisch korrekt» oder «syntaktisch falsch» (für eine gerade interessierende Syntax)
- bequemere Formalisierung:
  - Spezifikation der Menge F: *akzeptierende Zustände*
  - $F = \{z \in Z \mid h(z) = 1\}$
  - die anderen heißen *ablehnende Zustände*
- in graphischen Darstellungen
  - akzeptierende Zustände mit Doppelkringel: z

Endlicher Akzeptor: Beispiel



- der Moore-Automat von eben

Endlicher Akzeptor: Beispiel

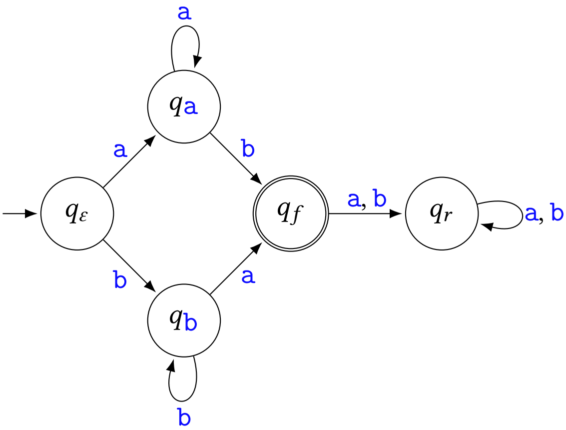


- der Moore-Automat von eben

Akzeptierte und abgelehnte Wörter

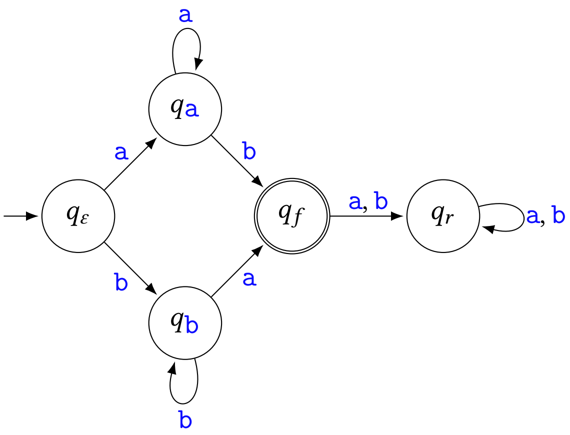
- Wort $w \in X^*$ wird *akzeptiert*, falls $f_*(z_0, w) \in F$.
- Wort $w \in X^*$ wird *abgelehnt*, falls $f_*(z_0, w) \notin F$.

Beispiel



- aaaba abgelehnt:
  $f_*(q_\varepsilon, \texttt{aaaba}) = q_r \notin F$
- aaab akzeptiert:
  $f_*(q_\varepsilon, \texttt{aaab}) = q_f \in F$.

Beispiel



- aaaba abgelehnt:
  $f_*(q_\varepsilon, \mathtt{aaaba}) = q_r \notin F$
- aaab akzeptiert:
  $f_*(q_\varepsilon, \mathtt{aaab}) = q_f \in F.$
- allgemein akzeptiert:
  - Wörter der Form $\mathtt{a}^k\mathtt{b}$ mit $k \geq 1$
  - Wörter der Form $\mathtt{b}^k\mathtt{a}$ mit $k \geq 1$
  - nichts anderes

Erkannte formale Sprache

- Die von einem Akzeptor $A = (Z, z_0, X, f, F)$ *akzeptierte* oder *erkannte formale Sprache* ist

$$L(A) = \{w \in X^* \mid f_*(z_0, w) \in F\}$$

- Das ist ganz einfache «Syntaxanalyse».
- in unserem Beispiel:

$$L(A) = \{\mathtt{a}\}^+\{\mathtt{b}\} \cup \{\mathtt{b}\}^+\{\mathtt{a}\}$$

Beispiel 2 einer erkennbaren Sprache

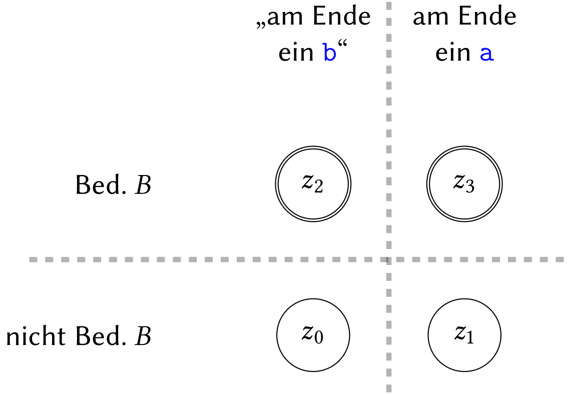
- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein $\mathtt{b}$ und
  - vor letztem $\mathtt{b}$ steht ein $\mathtt{a}$
- z. B.
  - in L:
    $\mathtt{ab}$, $\mathtt{abbab}$, $\mathtt{baab}$, $\mathtt{abaaa}$, …
    $\mathtt{aba}$, $\mathtt{bab}$, …
  - nicht in L:
    $\mathtt{b}$, $\mathtt{ba}$, $\mathtt{aaabba}$, $\mathtt{aaa}$, …
    $\mathtt{abb}$, $\mathtt{baa}$, …
- Behauptung: Es gibt EA, der L erkennt.

Beispiel 2 einer erkennbaren Sprache

	„am Ende ein b“	am Ende ein a
Bed. B		
nicht Bed. B		

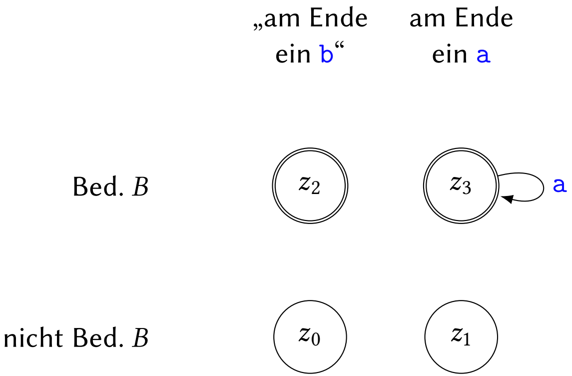
- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, ...
    aba, bab, ...
  - nicht in L:
    b, ba, aaabba, aaa, ...
    abb, baa, ...

Beispiel 2 einer erkennbaren Sprache



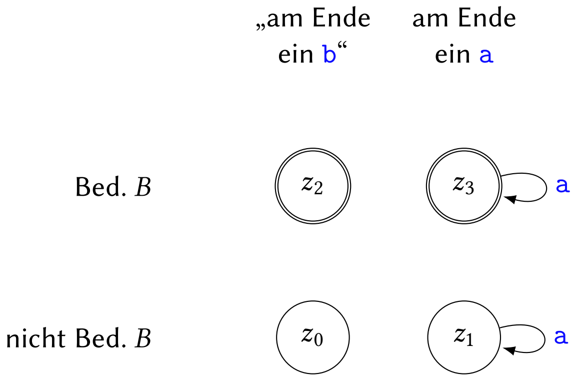
- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache



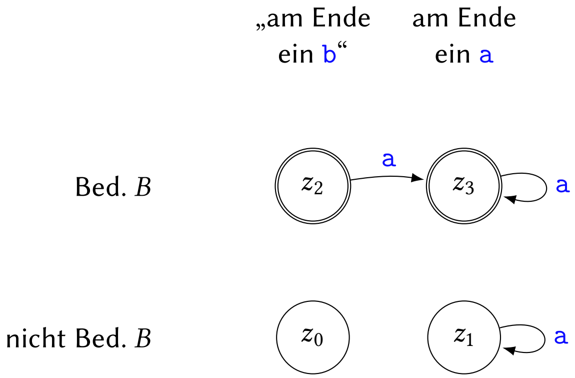
- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache



- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache



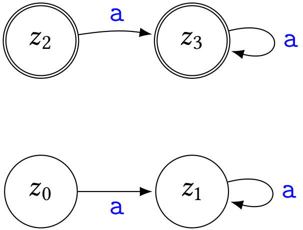
- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache

„am Ende ein b“ | am Ende ein a

Bed. B

nicht Bed. B



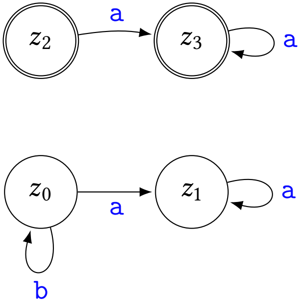
- formale Sprache L aller Wörter $w \in \{a, b\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache

„am Ende ein b“ | am Ende ein a

Bed. B

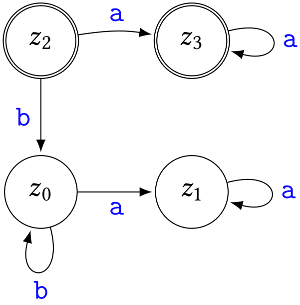
nicht Bed. B



- formale Sprache L aller Wörter $w \in \{a, b\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, ...
    aba, bab, ...
  - nicht in L:
    b, ba, aaabba, aaa, ...
    abb, baa, ...

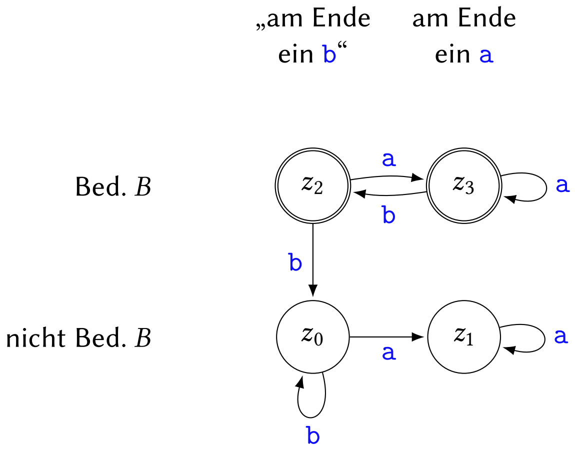
Beispiel 2 einer erkennbaren Sprache

	„am Ende ein b“	am Ende ein a
Bed. B		
nicht Bed. B		



- formale Sprache L aller Wörter $w \in \{a, b\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache



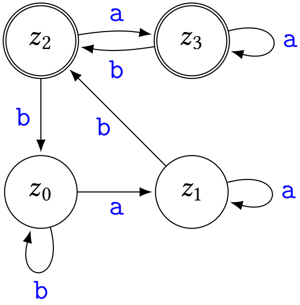
- formale Sprache L aller Wörter $w \in \{a, b\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache

„am Ende ein b“ | am Ende ein a

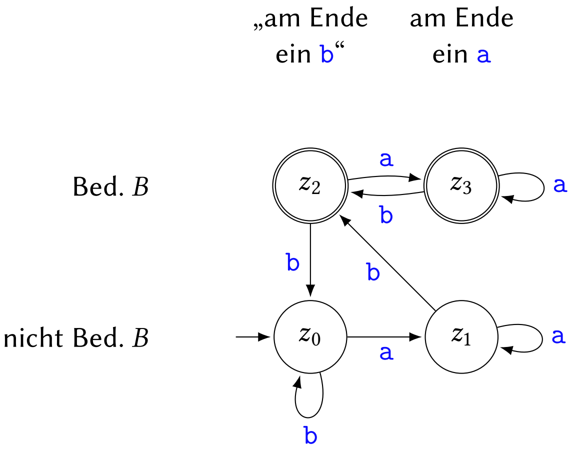
Bed. B

nicht Bed. B



- formale Sprache L aller Wörter $w \in \{a, b\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, …
    aba, bab, …
  - nicht in L:
    b, ba, aaabba, aaa, …
    abb, baa, …

Beispiel 2 einer erkennbaren Sprache



- formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$, die folgende Bedingung B erfüllen:
  - in w mindestens ein b und
  - vor letztem b steht ein a
- z. B.
  - in L:
    ab, abbab, baab, abaaa, ...
    aba, bab, ...
  - nicht in L:
    b, ba, aaabba, aaa, ...
    abb, baa, ...

Beispiel 3 einer erkennbaren Sprache

- Aufgabe aus dem Informatiker-Alltag
  - gegeben: Textdatei mit vielen Zeilen
  - gesucht: die Zeilen, in denen ein gewisses Wort m vorkommt

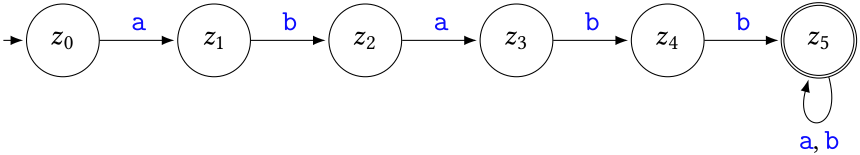
Beispiel 3 einer erkennbaren Sprache

- Aufgabe aus dem Informatiker-Alltag
  - gegeben: Textdatei mit vielen Zeilen
  - gesucht: die Zeilen, in denen ein gewisses Wort m vorkommt
- mit anderen Worten
  - gegeben: Zeichenkette w und Textmuster m
  - gesucht: Algorithmus, der feststellt, ob m in w vorkommt
  - das geht mit einem endlichen Akzeptor

Beispiel 3 einer erkennbaren Sprache

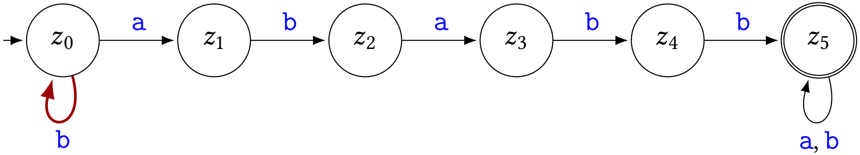
- Aufgabe aus dem Informatiker-Alltag
  - gegeben: Textdatei mit vielen Zeilen
  - gesucht: die Zeilen, in denen ein gewisses Wort m vorkommt
- mit anderen Worten
  - gegeben: Zeichenkette w und Textmuster m
  - gesucht: Algorithmus, der feststellt, ob m in w vorkommt
  - das geht mit einem endlichen Akzeptor
- Beispiel:
  - Eingabealphabet $X = \{\mathtt{a}, \mathtt{b}\}$
  - Textmuster $m = \mathtt{ababb}$
  - Ziel: endlicher Akzeptor A mit $L(A) = \{w_1\mathtt{ababb}w_2 \mid w_1, w_2 \in \{\mathtt{a}, \mathtt{b}\}^*\}$

Beispiel 3 — Entwicklung einer Lösung



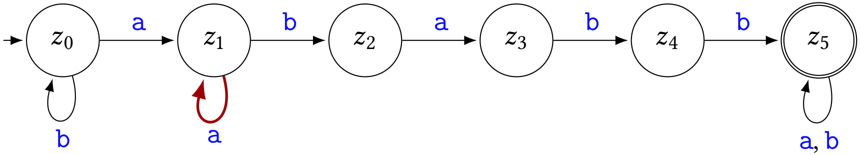
- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - `bbbababb`
  - `aababb`
  - `abbababb`
  - `abaababb`
  - `abababb`

Beispiel 3 — Entwicklung einer Lösung



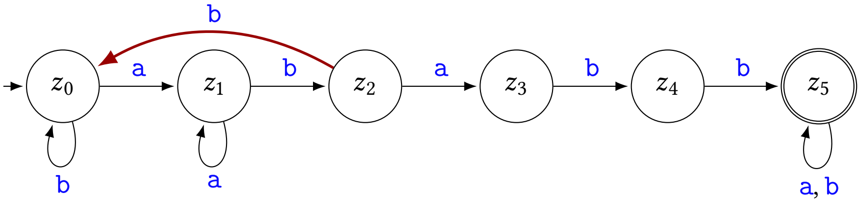
- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - bbbababb
  - aababb
  - abbababb
  - abaababb
  - abababb

Beispiel 3 — Entwicklung einer Lösung



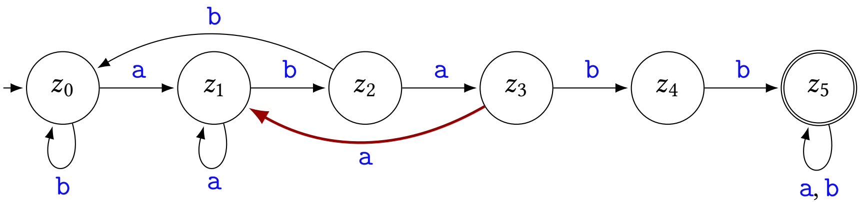
- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - bbbababb
  - aababb
  - abbababb
  - abaababb
  - abababb

Beispiel 3 — Entwicklung einer Lösung



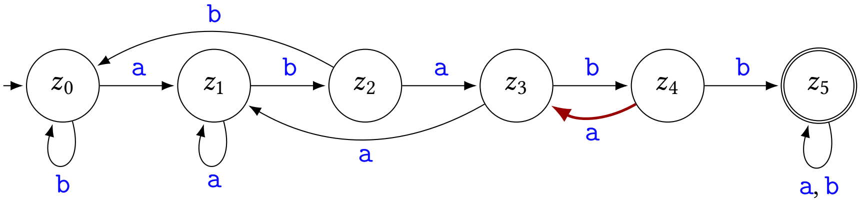
- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - bbbababb
  - aababb
  - abbababb
  - abaababb
  - abababb

Beispiel 3 — Entwicklung einer Lösung



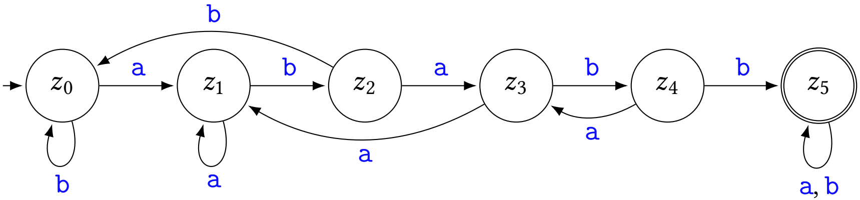
- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - bbbababb
  - aababb
  - abbababb
  - abaababb
  - abababb

Beispiel 3 — Entwicklung einer Lösung



- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - bbbababb
  - aababb
  - abbababb
  - abaababb
  - abababb

Beispiel 3 — Entwicklung einer Lösung



- es fehlen noch diverse Übergänge
- was ist z. B. mit:
  - bbbababb
  - aababb
  - abbababb
  - abaababb
  - abababb

Beispiel einer *nicht* erkennbaren Sprache

Satz Die formale Sprache

$$L = \{\mathtt{a}^k \mathtt{b}^k \mid k \in \mathbb{N}_0\}$$

kann von keinem endlichen Akzeptor erkannt werden.

Beweis «schwieriger» als einen Akzeptor hinzumalen:

Beispiel einer *nicht* erkennbaren Sprache

Karlsruher Institut für Technologie

Satz Die formale Sprache

$$L = \{\mathtt{a}^k \mathtt{b}^k \mid k \in \mathbb{N}_0\}$$

kann von keinem endlichen Akzeptor erkannt werden.

Beweis «schwieriger» als einen Akzeptor hinzumalen:

- betrachte «beliebigen» Akzeptor A und zeige $L(A) \neq L$
- Was bedeutet $L(A) \neq L$?
  - $L \not\subseteq L(A)$ oder
  - $L(A) \not\subseteq L$

Beispiel einer *nicht* erkennbaren Sprache (2)

Behauptung

- $L = \{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
  wird von keinem endlichen Akzeptor erkannt.

Beweis

«zielführende» Fallunterscheidung:

- 1. Fall: $L \not\subseteq L(A)$
  dann offensichtlich $L(A) \neq L$
- 2. Fall: $L \subseteq L(A)$
  zeige $L(A) \not\subseteq L$, d. h. ein «falsches» Wort wird akzeptiert

Beispiel einer *nicht* erkennbaren Sprache (3)

- $L = \{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
- $A = (Z, \ldots)$ beliebig mit $m = |Z|$
- zeige: wenn $L \subseteq L(A)$, dann $L(A) \nsubseteq L$

Beispiel einer *nicht* erkennbaren Sprache (3)

- $L = \{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
- $A = (Z, \ldots)$ beliebig mit $m = |Z|$
- zeige: wenn $L \subseteq L(A)$, dann $L(A) \not\subseteq L$
- betrachte $w = \mathtt{a}^m\mathtt{b}^m \in L$

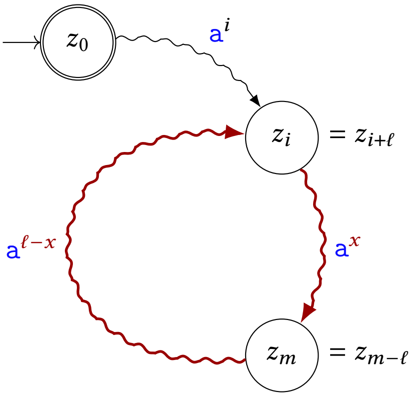
Beispiel einer *nicht* erkennbaren Sprache (3)

- $L = \{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
- $A = (Z, \dots)$ beliebig mit $m = |Z|$
- zeige: wenn $L \subseteq L(A)$, dann $L(A) \nsubseteq L$
- betrachte $w = \mathtt{a}^m\mathtt{b}^m \in L$
- $f_{**}(z_0, \mathtt{a}^m)$ besteht aus $m + 1$ Zuständen:
  - z_0
  - $z_1 = f(z_0, \mathtt{a})$
  - $z_2 = f(z_1, \mathtt{a})$
  - $\vdots$
  - $z_m = f(z_{m-1}, \mathtt{a})$

Beispiel einer *nicht* erkennbaren Sprache (3)

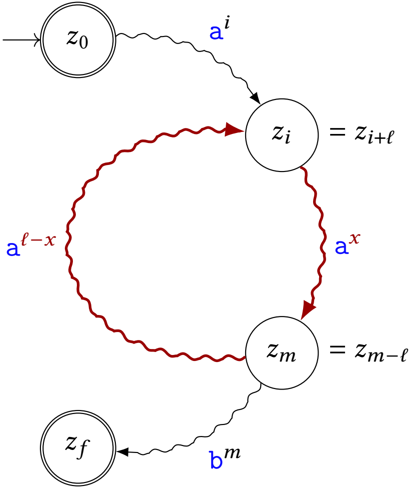
- $L = \{\mathtt{a}^k\mathtt{b}^k \mid k \in \mathbb{N}_0\}$
- $A = (Z, \dots)$ beliebig mit $m = |Z|$
- zeige: wenn $L \subseteq L(A)$, dann $L(A) \nsubseteq L$
- betrachte $w = \mathtt{a}^m\mathtt{b}^m \in L$
- $f_{**}(z_0, \mathtt{a}^m)$ besteht aus $m+1$ Zuständen:
  - z_0
  - $z_1 = f(z_0, \mathtt{a})$
  - $z_2 = f(z_1, \mathtt{a})$
  - $\vdots$
  - $z_m = f(z_{m-1}, \mathtt{a})$
- ein Zustand muss doppelt vorkommen:
  A läuft in einer Schleife

Beispiel einer *nicht* erkennbaren Sprache (4)



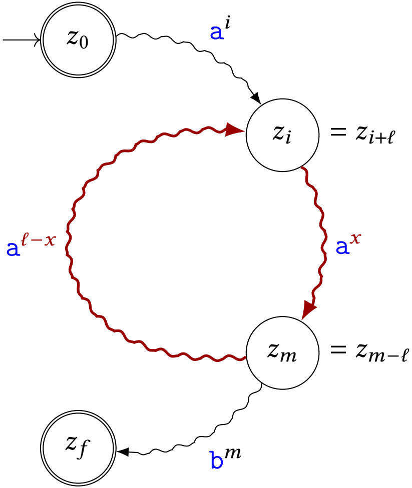
- seien $i \geq 0$ und $\ell \geq 1$ Zahlen mit $z_i = z_{i+\ell}$, also $f_*(z_0, \mathtt{a}^i) = f_*(z_0, \mathtt{a}^{i+\ell})$
- A «kann irgendwann nicht mehr unterscheiden», ob ℓ $\mathtt{a}$ mehr oder weniger in der Eingabe

Beispiel einer *nicht* erkennbaren Sprache (4)



- seien $i \geq 0$ und $\ell \geq 1$ Zahlen mit $z_i = z_{i+\ell}$, also $f_*(z_0, \mathtt{a}^i) = f_*(z_0, \mathtt{a}^{i+\ell})$
- A «kann irgendwann nicht mehr unterscheiden», ob ℓ $\mathtt{a}$ mehr oder weniger in der Eingabe
- $w = \mathtt{a}^m\mathtt{b}^m \in L$

Beispiel einer *nicht* erkennbaren Sprache (4)



- seien $i \geq 0$ und $\ell \geq 1$ Zahlen mit $z_i = z_{i+\ell}$, also $f_*(z_0, \mathtt{a}^i) = f_*(z_0, \mathtt{a}^{i+\ell})$
- A «kann irgendwann nicht mehr unterscheiden», ob ℓ $\mathtt{a}$ mehr oder weniger in der Eingabe
- $w = \mathtt{a}^m\mathtt{b}^m \in L$
- betrachte: $w' = \mathtt{a}^{m-\ell}\mathtt{b}^m \notin L$
- $$\begin{aligned} f_*(z_0, w') &= f_*(z_0, \mathtt{a}^{m-\ell}\mathtt{b}^m) \\ &= f_*(f_*(z_0, \mathtt{a}^i), \mathtt{a}^{m-\ell-i}\mathtt{b}^m) \\ &= f_*(f_*(z_0, \mathtt{a}^{i+\ell}), \mathtt{a}^{m-\ell-i}\mathtt{b}^m) \\ &= f_*(z_0, \mathtt{a}^m\mathtt{b}^m) \in F \end{aligned}$$
- $w' \in L(A)$ aber $w' \notin L$

Was ist wichtig

- Das sollten Sie mitnehmen:
  - «offizielle» Definition endlicher Akzeptoren:
    - Z
    - $z_0 \in Z$
    - X
    - $f : Z \times X \to Z$
    - $F \subseteq Z$
  - Wenn ein «sehr langes» Wort akzeptiert wird,
    - dann läuft der Automat in einer Schleife,
    - die beliebig oft durchlaufen werden kann ohne Akzeptanz zu ändern
- Das sollten Sie üben:
  - gegeben L: konstruiere A mit $L(A) = L$
  - gegeben A: bestimme $L(A)$

Zusammenfassung

- Moore-Automaten
  - tauchen z. B. im Zusammenhang mit diversen *Protokollen* in Betriebssystemen und bei Kommunikationssystemen auf
- insbesonderen Akzeptoren
  - primitive Syntaxanalyse
  - oft nützlich, z. B.
    - bei Compilerbau-Werkzeugen
    - Suche nach Text-Vorkommen, etc.