Grundbegriffe der Informatik

Kapitel 18: Endliche Automaten

Thomas Worsch

KIT, Institut für Theoretische Informatik

Wintersemester 2015/2016

Überblick

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

Wo sind wir?

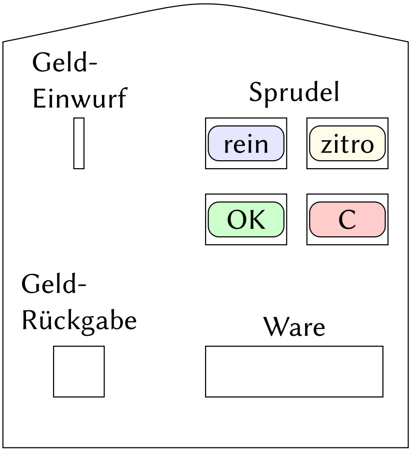
Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

Ein primitiver Getränkeautomat



Geld

- nur 1-Euro-Stücke
- erster Euro gespeichert
- weitere Münzen sofort zurück

vier Tasten

- rein Mineralwasser (1 Euro)
- zitro Zitronensprudel (1 Euro)
  - letzter Wunsch gemerkt
- C ggf. Geld zurück, ggf. Getränkewunsch gelöscht
- OK Geld und Getränkewunsch $\leadsto$ Getränk

Getränkeautomat: Zustände

zwischen den Eingaben müssen
Nachrichten gespeichert werden

und zwar

- schon Geld eingeworfen?
- schon Getränk ausgewählt?
- wenn ja: welches?

Formalisierung: Paare (x, y)

- $x \in \{0, 1\}$: schon eingeworfener Geldbetrag
- $y \in \{\mathtt{-}, \mathtt{R}, \mathtt{Z}\}$: Getränkewahl
- Zustandsmenge $Z = \{0, 1\} \times \{\mathtt{-}, \mathtt{R}, \mathtt{Z}\}$

Getränkeautomat: Eingaben

Eingaben führen zu *Zustandsänderungen.*

Eingaben hier:

- Einwurf eines Euros
- Drücken einer der vier Tasten

Modellierung der Eingaben: Symbole $\mathtt{1}$, $\mathtt{R}$, $\mathtt{Z}$, $\mathtt{C}$ und $\mathtt{O}$

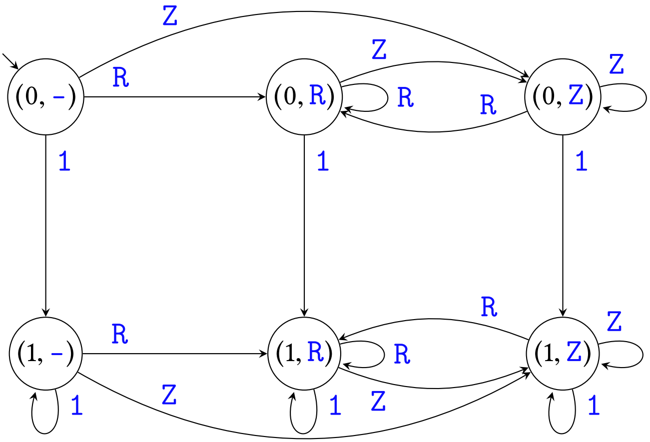
Eingabealphabet $X = \{\mathtt{1}, \mathtt{R}, \mathtt{Z}, \mathtt{C}, \mathtt{O}\}$

Getränkeautomat: Zustandsübergänge (1)

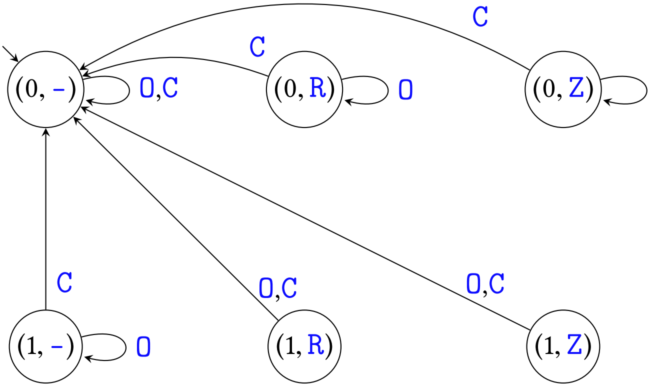
Zustandsübergang hängt ab von

- aktuellem Zustand $z \in Z$ und
  aktuellem Eingabesymbol $x \in X$
- z und x legen eindeutig den neuen Zustand fest
  - also *immer* und *eindeutig*
  - jedenfalls im ersten Semester
- Zustandsüberführungsfunktion $f : Z \times X \to Z$
- oft graphisch spezifiziert

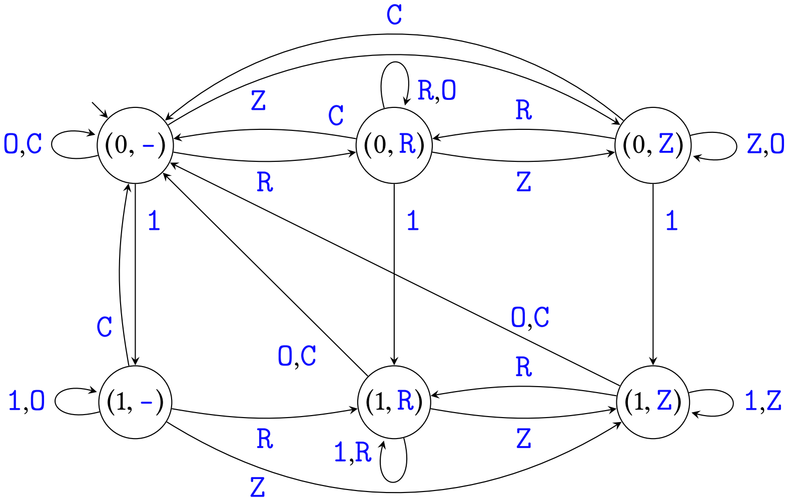
Getränkeautomat: Zustandsübergänge (2a)



Getränkeautomat: Zustandsübergänge (2b)



Getränkeautomat: Zustandsübergänge (2c)



Getränkeautomat: Ausgaben (1)

zweiter wesentlicher Aspekt jedes Automaten: Ausgaben

hier: Rückgeld, gewählte Flasche

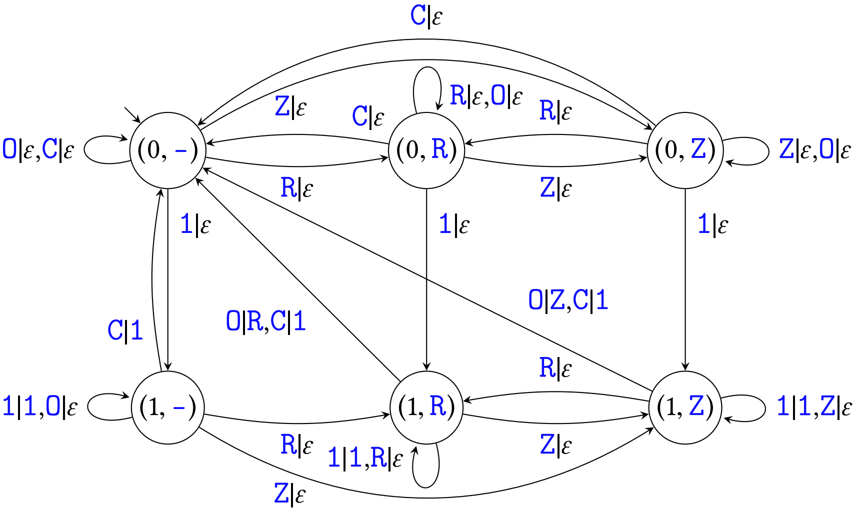
Ausgabealphabet $Y = \{\mathtt{1}, \mathtt{R}, \mathtt{Z}\}$

Formalisierung der Ausgaben

- abhängig von aktuellem Zustand $z \in Z$ und aktuellem Eingabesymbol $x \in X$
- z und x legen eindeutig die Ausgabe fest.
  - also *immer* und *eindeutig*
  - jedenfalls im ersten Semester
  - im allgemeinen *Wörter* über Y
- Formalisierung: Ausgabefunktion $g : Z \times X \to Y^*$
  - Funktionswert ε: „keine Ausgabe“

g im Zustandsübergangsdiagramm: $x|g(z,x)$

Getränkeautomat: Ausgaben (2)



Was ist wichtig

Das sollten Sie mitnehmen:

- alles endlich
- alles endlich beschreibbar
- im Beispiel Getränkeautomat:
  - aktueller Zustand und aktuelle Eingabe legen eindeutig fest:
  - nächsten Zustand
  - Ausgabe

Das sollten Sie üben:

- Zustandsdiagramm lesen

Wo sind wir?

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

Mealy-Automaten

formal

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

Darstellung als Graph:

- Knoten: Zustände
- Kanten: $z \xrightarrow{x|g(z,x)} f(z,x)$
- Anfangszustand: kleiner Pfeil $\longrightarrow z$

Verallgemeinerte Zustandsübergangsfunktionen (1)

nach Eingabe eines ganzen Wortes $w \in X^*$:

- erreichter Zustand?
- alle durchlaufenen Zustände?

definiere passende Funktionen f_* und f_{**}

Verallgemeinerte Zustandsübergangsfunktionen (1)

nach Eingabe eines ganzen Wortes $w \in X^*$:

- erreichter Zustand?
- alle durchlaufenen Zustände?

definiere passende Funktionen f_* und f_{**}

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

Verallgemeinerte Zustandsübergangsfunktionen (1)

nach Eingabe eines ganzen Wortes $w \in X^*$:

- erreichter Zustand?
- alle durchlaufenen Zustände?

definiere passende Funktionen f_* und f_{**}

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

$$\begin{aligned} \text{alternativ} &\quad \bar{f}_*(z, \varepsilon) = z \\ \forall w \in X^* : \forall x \in X : &\quad \bar{f}_*(z, xw) = \bar{f}_*(f(z, x), w) \end{aligned}$$

Definitionen äquivalent: $f_* = \bar{f}_*$

Verallgemeinerte Zustandsübergangsfunktionen (2)

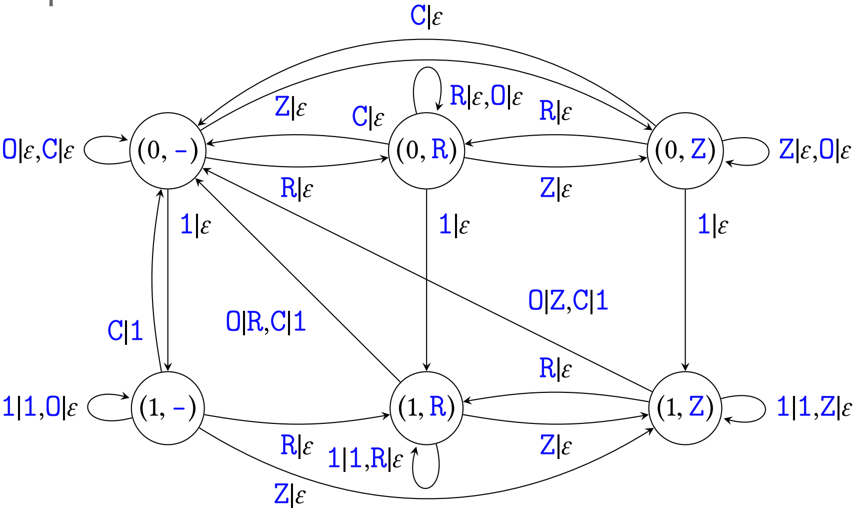
alle durchlaufenen Zustände:

- $f_{**} : Z \times X^* \to Z^*$

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

- auch hier wieder eine alternative Definitionsmöglichkeit

Bespiel Getränkeautomat



$f_{**}(\,(0, \texttt{-}), \texttt{R1RZO}) = (0, \texttt{-})\ (0, \texttt{R})\ (1, \texttt{R})\ (1, \texttt{R})\ (1, \texttt{Z})\ (0, \texttt{-})$

Verallgemeinerte Ausgabefunktionen

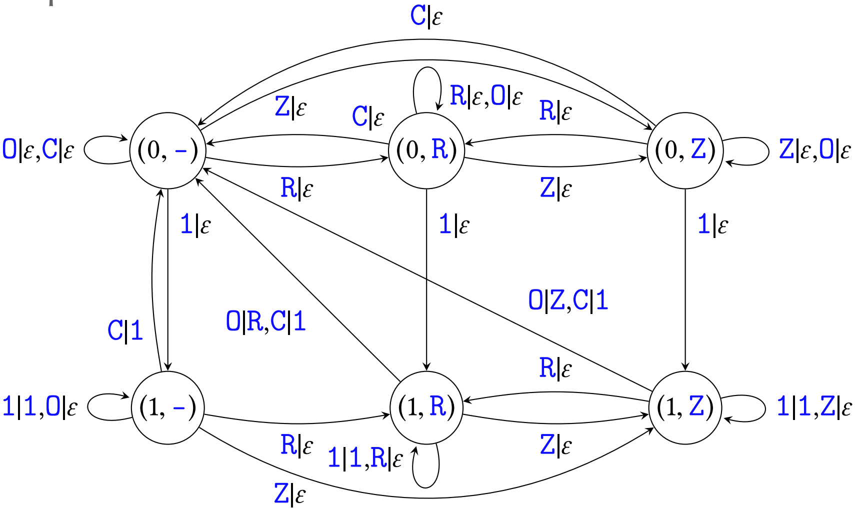
für „die letzte“ Ausgabe: $g_* : Z \times X^* \to Y^*$

$$\begin{aligned} g_*(z, \varepsilon) &= \varepsilon \\ g_*(z, wx) &= g(f_*(z, w), x) \end{aligned}$$

Für „alle Ausgaben konkateniert“: $g_{**} : Z \times X^* \to Y^*$:

$$\begin{aligned} g_{**}(z, \varepsilon) &= \varepsilon \\ g_{**}(z, wx) &= g_{**}(z, w) \cdot g_*(z, wx) \end{aligned}$$

Bespiel Getränkeautomaten



$$g_{**}(\,(0, \texttt{-}), \texttt{R1RZO}) = \varepsilon\varepsilon\varepsilon\varepsilon\texttt{Z} = \texttt{Z}$$

Was ist wichtig

Das sollten Sie mitnehmen:

- „offizielle“ Definition von Mealy-Automat:
  - Z
  - $z_0 \in Z$
  - X
  - $f : Z \times X \to Z$
  - Y
  - $g : Z \times X \to Y^*$ Ausgabe hängt von der Eingabe ab

Das sollten Sie üben:

- zu vorgegebenem „Verhalten“ Beispielautomaten konstruieren
- von vorgebenenen Automaten ihr Verhalten verstehen

Wo sind wir?

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

Moore-Automat

Ausgabe «im Zustand» statt beim Zustandsübergang

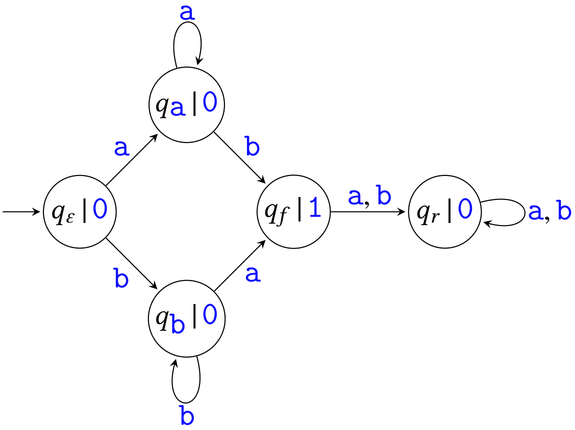
- genauer beim «Erreichen des Zustandes»

Definition

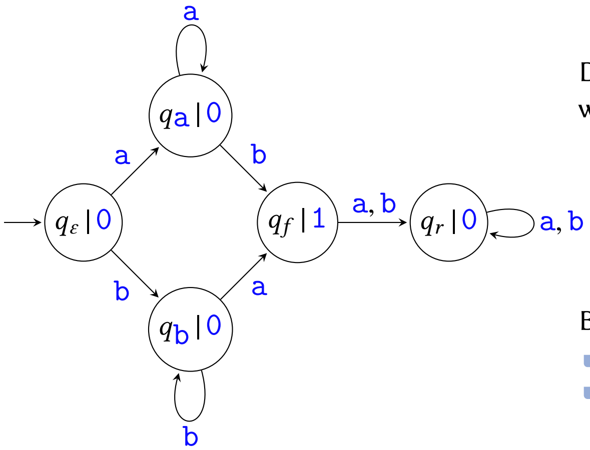
- endliche *Zustandsmenge* Z,
- *Anfangszustand* $z_0 \in Z$,
- *Eingabealphabet* X,
- *Zustandsüberführungsfunktion* $f : Z \times X \to Z$,
- *Ausgabealphabet* Y,
- *Ausgabefunktion* $h : Z \to Y^*$

Moore-Automat: Beispiel aus `tikz`-Dokumentation

graphische Darstellung analog zu Mealy-Automaten,
nur die Ausgaben in den Zuständen:



Verallgemeinerte Zustandsübergangsfunktionen



Definition von f_*, f_{**} wie bei Mealy-Automaten

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": $g_{**} = h^{**} \circ f_{**}$

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

- Erinnerung: h^{**} der durch h induzierte Homomorphismus

Beispiel

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

also

- $g_*(q_\varepsilon, \texttt{aaaba}) = h(f_*(q_\varepsilon, \texttt{aaaba})) = h(q_r) = \texttt{0}$
- $$\begin{aligned} g_{**}(q_\varepsilon, \texttt{aaaba}) &= h^{**}(f_{**}(q_\varepsilon, \texttt{aaaba})) = h^{**}(q_\varepsilon q_{\texttt{a}} q_{\texttt{a}} q_{\texttt{a}} q_f q_r) \\ &= h(q_\varepsilon) h(q_{\texttt{a}}) h(q_{\texttt{a}}) h(q_{\texttt{a}}) h(q_f) h(q_r) = \texttt{000010} \end{aligned}$$

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 *nicht* von der Eingabe ab

Das sollten Sie üben:

- vorgegebenes Verhalten mit Moore-Automat realisieren
- vorgebenenen Moore-Automaten analysieren

Wo sind wir?

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

Endliche Akzeptoren — ein wichtiger Sonderfall von Moore-Automaten

immer genau ein Bit Ausgabe

- $Y = \{0, 1\}$ und
- $\forall z : h(z) \in Y$

Interpretation der Ausgabe:

- Eingabe war „gut“ oder „schlecht“ bzw.
- „syntaktisch korrekt“ oder „syntaktisch falsch“ (für eine gerade interessierende Syntax)

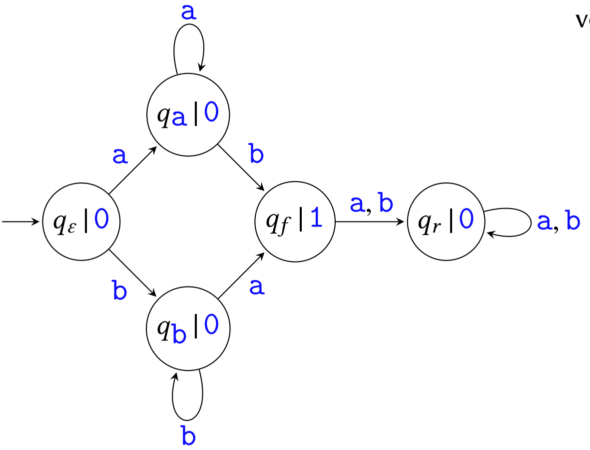
bequemere Formalisierung:

- Spezifikation der Menge F: *akzeptierende Zustände*
- $F = \{z \mid h(z) = 1\}$
- die anderen heißen *ablehnende Zustände*

in graphischen Darstellungen

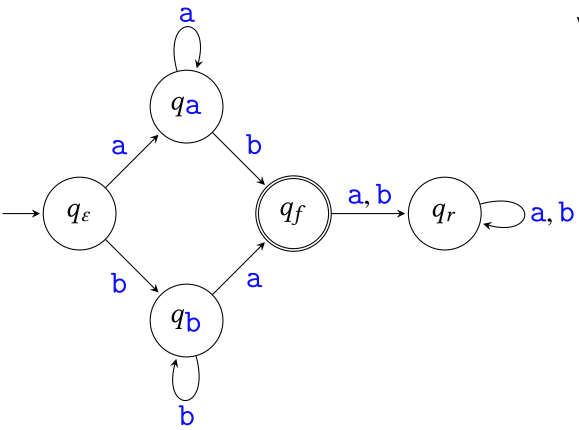
- akzeptierende Zustände mit Doppelkringel: z

Endlicher Akzeptor: Beispiel



der Moore-Automat von eben

Endlicher Akzeptor: Beispiel



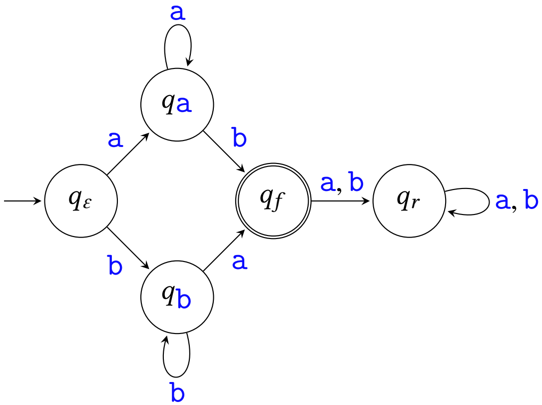
der Moore-Automat von eben

Akzeptierte und abgelehnte Wörter

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

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

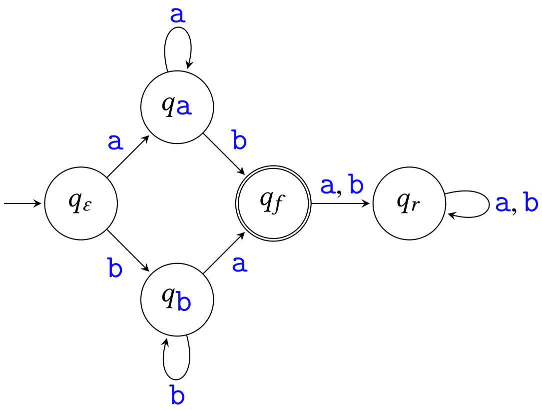
Beispiel



aaaba abgelehnt:
$f_*(z_0, \texttt{aaaba}) = q_r \notin F$

aaab akzeptiert:
$f_*(z_0, \texttt{aaab}) = q_f \in F.$

Beispiel



aaaba abgelehnt:
$f_*(z_0, \texttt{aaaba}) = q_r \notin F$

aaab akzeptiert:
$f_*(z_0, \texttt{aaab}) = q_f \in F$.

allgemein akzeptiert:

- Wörter der Form $\texttt{a}^k\texttt{b}$
- Wörter der Form $\texttt{b}^k\texttt{a}$
- nicht 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 $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein `b` vor und
- vor dem letzten `b` steht ein `a`

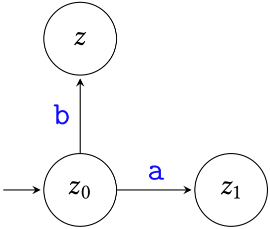
Behauptung: Es gibt EA, der L erkennt.

Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein $\mathtt{b}$ vor und
- vor dem letzten $\mathtt{b}$ steht ein $\mathtt{a}$

Behauptung: Es gibt EA, der L erkennt.

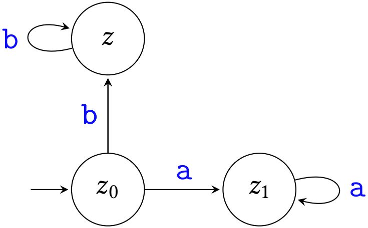


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein b vor und
- vor dem letzten b steht ein a

Behauptung: Es gibt EA, der L erkennt.

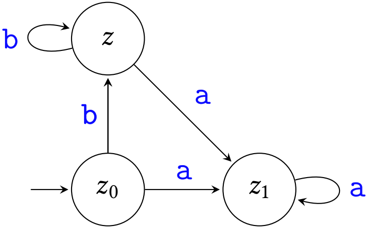


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\texttt{a}, \texttt{b}\}^*$ mit:

- in w kommt mindestens ein b vor und
- vor dem letzten b steht ein a

Behauptung: Es gibt EA, der L erkennt.

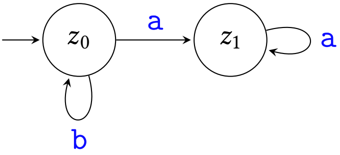


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein $\mathtt{b}$ vor und
- vor dem letzten $\mathtt{b}$ steht ein $\mathtt{a}$

Behauptung: Es gibt EA, der L erkennt.

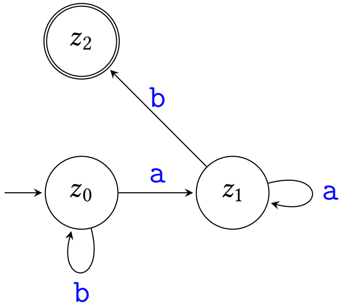


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein $\mathtt{b}$ vor und
- vor dem letzten $\mathtt{b}$ steht ein $\mathtt{a}$

Behauptung: Es gibt EA, der L erkennt.

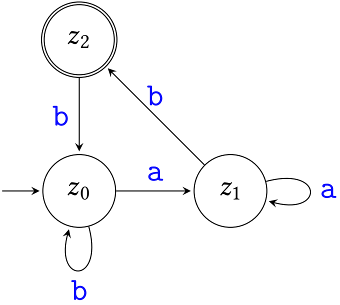


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein $\mathtt{b}$ vor und
- vor dem letzten $\mathtt{b}$ steht ein $\mathtt{a}$

Behauptung: Es gibt EA, der L erkennt.

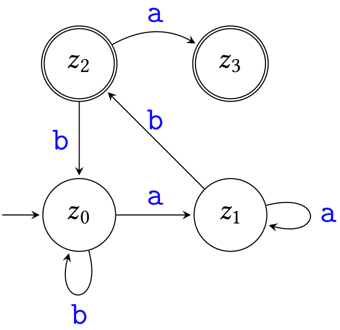


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\texttt{a}, \texttt{b}\}^*$ mit:

- in w kommt mindestens ein $\texttt{b}$ vor und
- vor dem letzten $\texttt{b}$ steht ein $\texttt{a}$

Behauptung: Es gibt EA, der L erkennt.

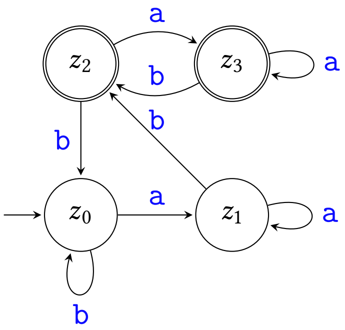


Beispiel 2 einer erkennbaren Sprache

formale Sprache L aller Wörter $w \in \{\mathtt{a}, \mathtt{b}\}^*$ mit:

- in w kommt mindestens ein $\mathtt{b}$ vor und
- vor dem letzten $\mathtt{b}$ steht ein $\mathtt{a}$

Behauptung: Es gibt EA, der L erkennt.



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

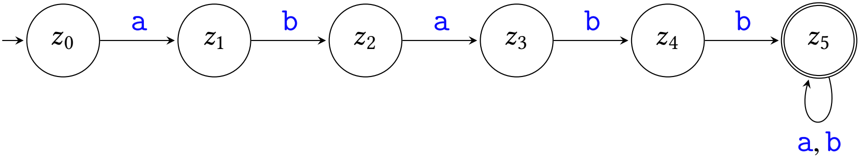
mit anderen Worten

- 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

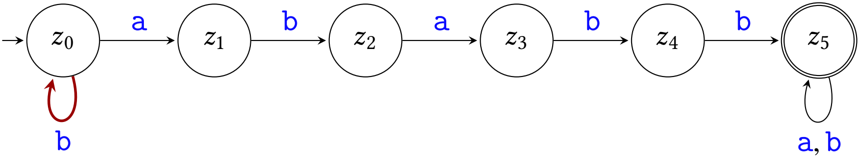


es fehlen noch diverse Übergänge

was ist z. B. mit:

- bbbababb
- aaaababb
- abbababb
- abaababb
- abababb

Beispiel 3 — Entwicklung einer Lösung

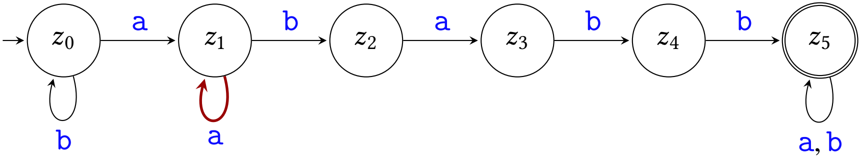


es fehlen noch diverse Übergänge

was ist z. B. mit:

- bbbababb
- aaaababb
- abbababb
- abaababb
- abababb

Beispiel 3 — Entwicklung einer Lösung

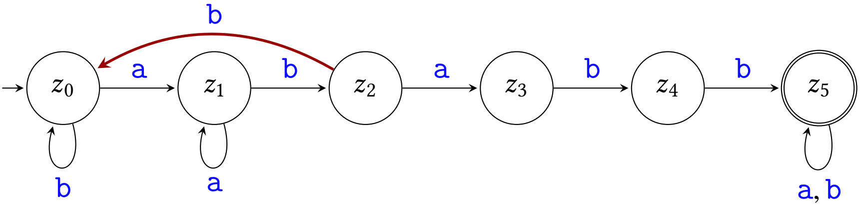


es fehlen noch diverse Übergänge

was ist z. B. mit:

- bbbababb
- aaaababb
- abbababb
- abaababb
- abababb

Beispiel 3 — Entwicklung einer Lösung

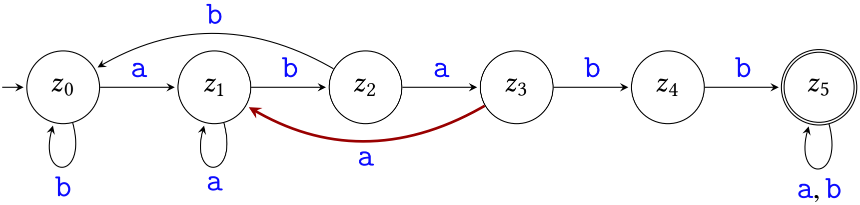


es fehlen noch diverse Übergänge

was ist z. B. mit:

- bbbababb
- aaaababb
- abbababb
- abaababb
- abababb

Beispiel 3 — Entwicklung einer Lösung

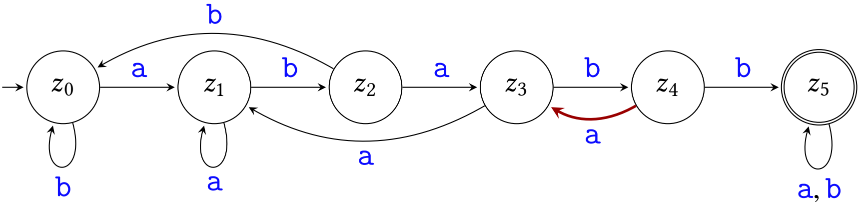


es fehlen noch diverse Übergänge

was ist z. B. mit:

- bbbababb
- aaaababb
- abbababb
- abaababb
- abababb

Beispiel 3 — Entwicklung einer Lösung

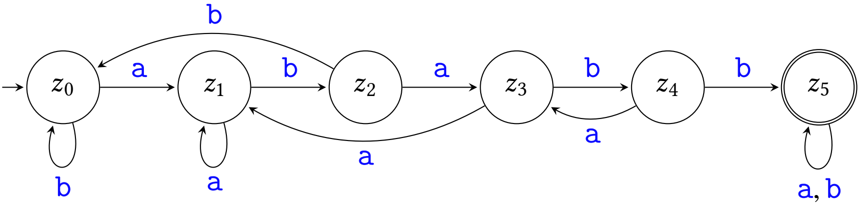


es fehlen noch diverse Übergänge

was ist z. B. mit:

- bbbababb
- aaaababb
- abbababb
- abaababb
- abababb

Beispiel 3 — Entwicklung einer Lösung



es fehlen noch diverse Übergänge

was ist z. B. mit:

- `bbbababb`
- `aaaababb`
- `abbababb`
- `abaababb`
- `abababb`

Beispiel einer *nicht* erkennbaren Sprache

Behauptung: 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

Behauptung: 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 $L(A) \neq L$?
  - $L \nsubseteq L(A)$ oder $L(A) \nsubseteq L$
- «zielführende» Fallunterscheidung:
  - 1. Fall: $L \nsubseteq L(A)$
    offensichtlich $L(A) \neq L$
  - 2. Fall: $L \subseteq L(A)$
    zeige $L(A) \nsubseteq L$, d. h. ein „falsches“ Wort wird akzeptiert

Beispiel einer *nicht* erkennbaren Sprache (2)

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

$A = (Z, \dots)$ beliebig mit $m = |Z|$

zeige: wenn $L \subseteq L(A)$, dann $L(A) \nsubseteq L$

Beispiel einer *nicht* erkennbaren Sprache (2)

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

$A = (Z, \dots)$ beliebig mit $m = |Z|$

zeige: wenn $L \subseteq L(A)$, dann $L(A) \nsubseteq L$

betrachte $w = \mathtt{a}^m\mathtt{b}^m$

Beispiel einer *nicht* erkennbaren Sprache (2)

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

$A = (Z, \dots)$ beliebig mit $m = |Z|$

zeige: wenn $L \subseteq L(A)$, dann $L(A) \nsubseteq L$

betrachte $w = \mathtt{a}^m\mathtt{b}^m$

$f_{**}(z_0, \mathtt{a}^m)$ 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 (3)

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

betrachte: $w' = \mathtt{a}^{m-\ell}\mathtt{b}^m \notin L$

$$\begin{aligned}
f_*(z_0, w') &= f_*(z_0, \mathtt{a}^{m-\ell}\mathtt{b}^m) \\
&= f_*(f_*(z_0, \mathtt{a}^i), \mathtt{a}^{m-\ell-i}\mathtt{b}^m) \\
&= f_*(f_*(z_0, \mathtt{a}^{i+\ell}), \mathtt{a}^{m-\ell-i}\mathtt{b}^m) \\
&= f_*(z_0, \mathtt{a}^m\mathtt{b}^m) \in F
\end{aligned}$$

Beispiel einer *nicht* erkennbaren Sprache (3)

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

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» «uniformes» 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

Mealy-Automaten

Moore-Automaten

- taucht im Zusammenhang mit diversen *Protokollen* z. B. in Betriebssystemen und bei Kommunikationssystemen auf

insbesonderen Akzeptoren

- primitive Syntaxanalyse
- oft nützlich, z. B.
  - bei Compilerbau-Werkzeugen
  - Suche nach Text-Vorkommen, etc.