Endliche Automaten

Endliche Automaten

- sind ein Kalkül zur Spezifikation von realen oder abstrakten Maschinen
- regieren auf äußere Ereignisse (=Eingaben)
- ändern ihren inneren Zustand
- produzieren gegebenenfalls eine Ausgabe
- werden in der Regel durch gerichtete, markierte Graphen dargestellt.

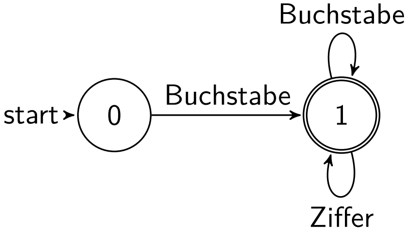
Endliche Automaten werden eingesetzt, um

- das Verhalten realer Maschinen (z.B. Ampelanlagen, Getränkeautomaten) zu spezifizieren
- das Verhalten von Softwarekomponenten (z.B. Benutzeroberflächen) zu spezifizieren
- einfache Sprachen (z.B. Bezeichner und Zahlen in Programmiersprache) zu spezifizieren.

Einführende Beispiele - Bezeichner in Programmiersprachen

Endliche Automaten und Sprachen

- Endlicher Automat definiert eine Sprache, d.h. Menge von Worten über einem Alphabet.

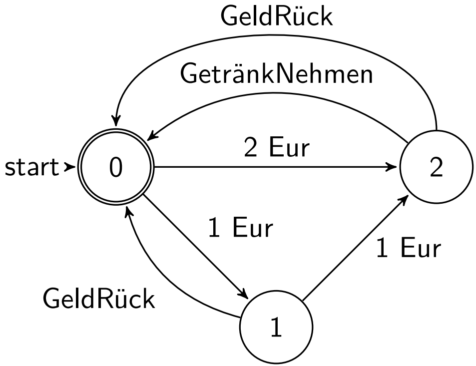


Beispiel: Automat akzeptiert jede Folge von Buchstaben und Ziffern, die mit einem Buchstaben beginnt.

Einführende Beispiele - Getränkeautomat

Endliche Automaten und Maschinen

- Endlicher Automat spezifiziert das Verhalten einer Maschine.



Beispiel: Automat akzeptiert Folgen von Ereignissen zur Bedienung eines Getränkeautomaten.

Deterministische endliche Automaten

Definition 1

Ein deterministischer endlicher Automat (DFA) (engl. deterministic finite automaton) A ist ein 5-Tupel $(\Sigma, Q, \delta, q_0, F)$, wobei

- Q eine endliche Menge von Zuständen ist,
- Σ ein endliches Alphabet ist,
- $\delta : Q \times \Sigma \to Q$ die (partielle) Übergangsfunktion ist,
- q_0 der Startzustand ist,
- $F \subseteq Q$ die Menge von Endzuständen (auch: akzeptierenden Zuständen) ist.

Erinnerung:

- Eine Funktion mit Definitionsbereich D und Bildbereich ist eine Relation $f \subseteq D \times B$ mit $((x, y) \in f \wedge (x, z) \in f) \Rightarrow y = z$.
- f heißt **total**, falls zu jedem $x \in D$ ein $y \in B$ existiert mit $(x, y) \in f$. Andernfalls heißt f **partiell**.

Deterministische endliche Automaten

Definition 1

Ein deterministischer endlicher Automat (DFA) (engl. deterministic finite automaton) A ist ein 5-Tupel $(\Sigma, Q, \delta, q_0, F)$, wobei

- Q eine endliche Menge von Zuständen ist,
- Σ ein endliches Alphabet ist,
- $\delta : Q \times \Sigma \to Q$ die (partielle) Übergangsfunktion ist,
- q_0 der Startzustand ist,
- $F \subseteq Q$ die Menge von Endzuständen (auch: akzeptierenden Zuständen) ist.

- Ist die Übergangsfunktion δ eines DFA total, so heißt A **vollständig**.
- $r = \delta(q, a)$ heißt **Nachfolgezustand** von q unter a.
- A heißt deterministisch, da es zu jedem Paar $(q, a), q \in Q, a \in \Sigma$, höchstens einen Nachfolgezustand $\delta(q, a)$ gibt.

Beispiele

DFA A_1 definiert durch

$$Q = \{0, 1\}, q_0 = 0$$
$$\Sigma = \{\text{Buchstabe, Ziffer}\}$$
$$F = \{1\}$$

und Übergangsfunktion δ definiert durch

δ	Buchstabe	Ziffer
0	1	–
1	1	1

DFA A_2 definiert durch

$$Q = \{0, 1, 2\}, q_0 = 0$$
$$\Sigma = \{\text{1 Eur, 2 Eur, GeldRück, GetränkNehmen}\}$$
$$F = \{0\}$$

und Übergangsfunktion δ definiert durch

δ	1 Eur	2 Eur	GR	GTN
0	1	2	–	–
1	2	–	0	–
2	–	–	0	0

Graphische Darstellung von DFAs

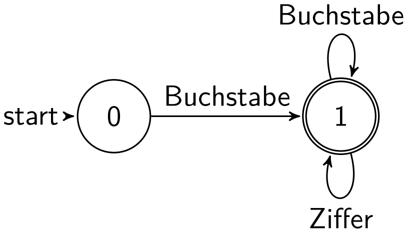
DFAs und Graphen

- pro Zustand ein Knoten, jeweils markiert mit Zustand, Startzustand und Endzustände gekennzeichnet
- Kante (q, r) markiert mit $a \in \Sigma$ genau dann, wenn $\delta(q, a) = r$
- erhalten Multigraphen, da Kanten sich nur durch Markierung unterscheiden können

DFA A_1 :

$$Q = \{0, 1\}, q_0 = 0$$
$$\Sigma = \{\text{Buchstabe, Ziffer}\}$$
$$F = \{1\}$$

δ	Buchstabe	Ziffer
0	1	–
1	1	1



Graphische Darstellung von DFAs

DFAs und Graphen

- pro Zustand ein Knoten, jeweils markiert mit Zustand, Startzustand und Endzustände gekennzeichnet
- Kante (q, r) markiert mit $a \in \Sigma$ genau dann, wenn $\delta(q, a) = r$
- erhalten Multigraphen, da Kanten sich nur durch Markierung unterscheiden können

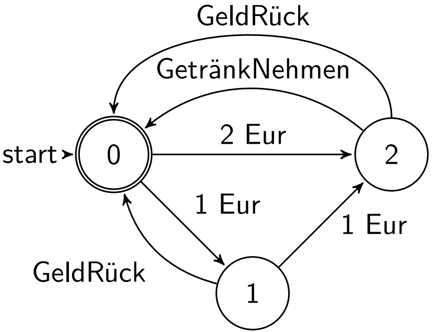
DFA A_2:

$Q = \{0, 1, 2\}, q_0 = 0$

$\Sigma = \{1 \text{ Eur}, 2 \text{ Eur}, \text{GeldRück}, \text{GetränkNehmen}\}$

$F = \{0\}$

δ	1 Eur	2 Eur	GR	GTN
0	1	2	–	–
1	2	–	0	–
2	–	–	0	0



Vervollständigung endlicher Automaten

Erinnerung: Ein DFA $A = (\Sigma, Q, \delta, q_0, F)$ heißt vollständig, wenn die Übergangsfunktion δ total ist.

Zum besseren Verständnis, zur Realisierung und zur formalen Argumentation kann es sinnvoll oder notwendig sein, einen DFA A so zu ergänzen, dass er vollständig ist.

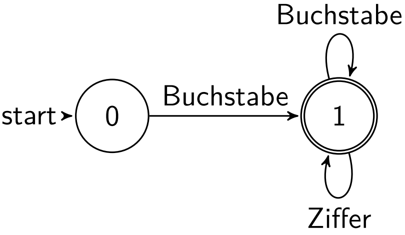
Ergänzung endlicher Automaten zu vollständigen Automaten

Sei $A = (\Sigma, Q, \delta, q_0, F)$ ein DFA. Zur Ergänzung wird

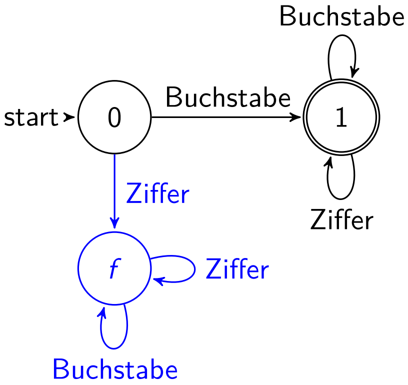
1. ein neuer Zustand f mit $f \notin F$ eingeführt,
2. für alle Paare $(q, a) \in Q \times \Sigma$, für die die Übergangsfunktion δ nicht definiert ist, $\delta(q, a) = f$ gesetzt,
3. für alle Paare $(f, a), a \in \Sigma$, $\delta(f, a) = f$ gesetzt.

Vervollständigung endlicher Automaten

DFA A_1:

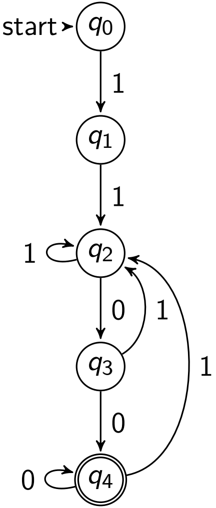


Vervollständigung von A_1 :

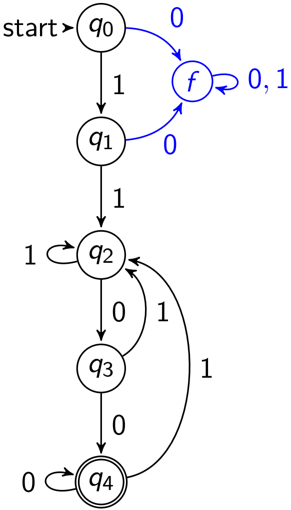


Vervollständigung endlicher Automaten

DFA A_3**:**



Vervollständigung von A_3 :



Erweiterung von Übergangsfunktionen

Um das Verhalten eines DFAs $A = (\Sigma, Q, \delta, q_0, F)$ bei Folgen von Symbolen zu beschreiben, erweitern wir die Übergangsfunktion.

Definition 2

Sei $A = (\Sigma, Q, \delta, q_0, F)$ deterministischer endlicher Automat. Dann wird die erweiterte Übergangsfunktion $\hat{\delta} : Q \times \Sigma^* \to Q$ rekursiv wie folgt definiert:

$$\hat{\delta}(q, \epsilon) = q \quad \text{für alle } q \in Q \text{ und das leere Wort } \epsilon$$

$$\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a) \quad \text{für alle } q \in Q, w \in \Sigma^*, a \in \Sigma.$$

Bemerkungen:

1. $\hat{\delta}(q, a) = \delta(q, a) \quad$ für alle $q \in Q$ und $a \in \Sigma$

Bemerkungen:

1. Wir unterscheiden häufig nicht zwischen δ und $\hat{\delta}$ und schreiben auch δ für die erweiterte Übergangsfunktion $\hat{\delta}$.

Erweiterung von Übergangsfunktionen

Definition 2

Sei $A = (\Sigma, Q, \delta, q_0, F)$ deterministischer endlicher Automat. Dann wird die erweiterte Übergangsfunktion $\hat{\delta} : Q \times \Sigma^* \to Q$ rekursiv wie folgt definiert:

$$\hat{\delta}(q, \epsilon) = q) \quad \text{für alle } q \in Q \text{ und das leere Wort } \epsilon$$

$$\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a) \quad \text{für alle } q \in Q, w \in \Sigma^*, a \in \Sigma.$$

Alternative Formulierungen

Für $q, q' \in Q$ und $w = a_1 \cdots a_n, a_i \in \Sigma, i = 1, \ldots, n$, gilt

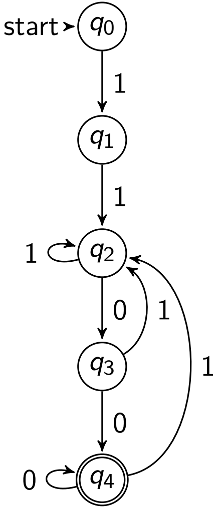
1. $$\hat{\delta}(q, w) = \delta(\ldots\delta(\delta(q, a_1), a_2)\ldots, a_n)$$

2. $\hat{\delta}(q, w) = q'$ genau dann, wenn es $r_0, \ldots, r_n \in Q$ gibt mit $r_0 = q, r_n = q'$ und

$$\delta(r_i, a_{i+1}) = r_{i+1} \quad \text{für } i = 0, \ldots, n-1.$$

Erweiterung von Übergangsfunktionen - Beispiele

DFA A_3:



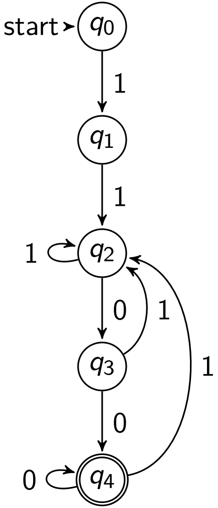
$$
\begin{aligned}
\delta(q_1, 110) &= \delta(\delta(q_1, 11), 0) \\
&= \delta(\delta(\delta(q_1, 1), 1), 0) \\
&= \delta(\delta(q_2, 1), 0) \\
&= \delta(q_2, 0) \\
&= q_3 \notin F
\end{aligned}
$$

Folge von Zuständen:

$r_0 = q_1, r_1 = q_2, r_2 = q_2, r_3 = q_3$

Erweiterung von Übergangsfunktionen - Beispiele

DFA A_3:



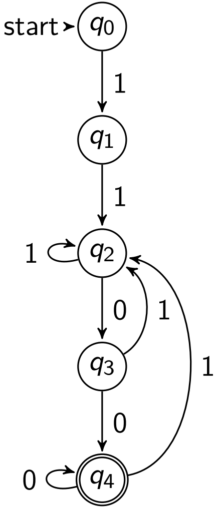
$$\begin{aligned}
\delta(q_0, 11100) &= \delta(\delta(q_0, 1110), 0) \\
&= \delta(\delta(\delta(q_0, 111), 0), 0) \\
&= \delta(\delta(\delta(\delta(q_0, 11), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(\delta(q_0, 1), 1), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(q_1, 1), 1), 0), 0) \\
&= \delta(\delta(\delta(q_2, 1), 0), 0) \\
&= \delta(\delta(q_2, 0), 0) \\
&= \delta(q_3, 0) \\
&= q_4 \in F
\end{aligned}$$

Folge von Zuständen:

$r_0 = q_0, r_1 = q_1, r_2 = q_2, r_3 = q_2, r_4 = q_3$
$r_5 = q_4$

Erweiterung von Übergangsfunktionen - Beispiele

DFA A_3**:**



$$
\begin{aligned}
\delta(q_0, 100) &= \delta(\delta(q_0, 10), 0) \\
&= \delta(\delta(\delta(q_0, 1), 0), 0) \\
&= \delta(\delta(q_1, 0), 0) \\
&= -;
\end{aligned}
$$

nicht definiert, da $\delta(q_1, 0)$

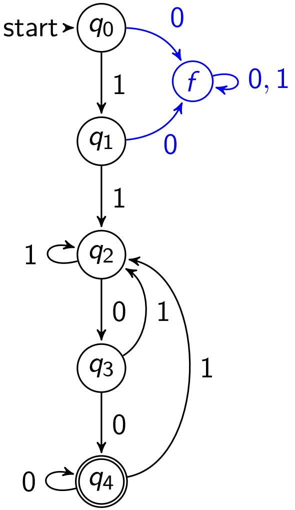
nicht definiert

Folge von Zuständen:

$r_0 = q_0, r_1 = q_1,$ r_2 nicht definiert

Erweiterung von Übergangsfunktionen - Beispiele

Vervollständigter A_3 :



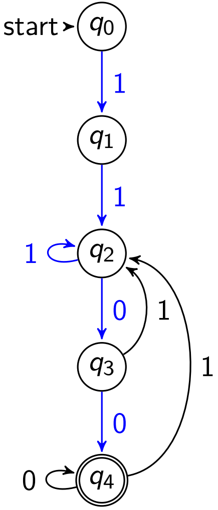
$$\begin{aligned}\delta(q_0,100) &= \delta(\delta(q_0,10),0)\\ &= \delta(\delta(\delta(q_0,1),0),0)\\ &= \delta(\delta(q_1,0),0)\\ &= \delta(f,0)\\ &= f\end{aligned}$$

Folge von Zuständen:

$r_0 = q_0, r_1 = q_1, r_2 = f, r_3 = f$

Erweiterung von Übergangsfunktionen - Berechnung als Weg

DFA A_3:



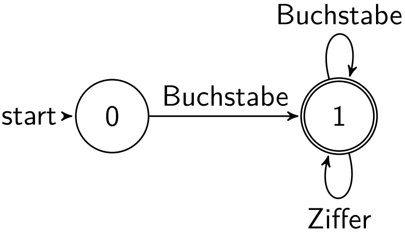
$$
\begin{aligned}
\delta(q_0, 11100) &= \delta(\delta(q_0, 1110), 0) \\
&= \delta(\delta(\delta(q_0, 111), 0), 0) \\
&= \delta(\delta(\delta(\delta(q_0, 11), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(\delta(q_0, 1), 1), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(q_1, 1), 1), 0), 0) \\
&= \delta(\delta(\delta(q_2, 1), 0), 0) \\
&= \delta(\delta(q_2, 0), 0) \\
&= \delta(q_3, 0) \\
&= q_4 \in F
\end{aligned}
$$

Folge von Zuständen:

$r_0 = q_0, r_1 = q_1, r_2 = q_2, r_3 = q_2, r_4 = q_3$
$r_5 = q_4$

Erweiterung von Übergangsfunktionen - Beispiele

DFA A_1**:**



B:= Buchstabe, Z:=Ziffer

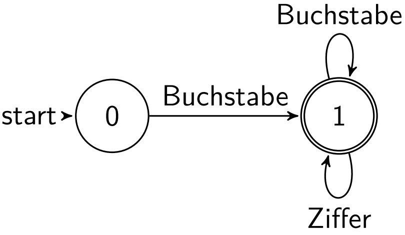
$$
\begin{aligned}
\delta(0, BZB) &= \delta(\delta(0, BZ), B) \\
&= \delta(\delta(\delta(0, B), Z), B) \\
&= \delta(\delta(1, Z), B) \\
&= \delta(1, B) \\
&= 1 \in F
\end{aligned}
$$

Folge von Zuständen:

$r_0 = 0, r_1 = 1, r_2 = 1, r_3 = 1$

Erweiterung von Übergangsfunktionen - Beispiele

DFA A_1:



B:= Buchstabe, Z:=Ziffer

$$\begin{aligned}\delta(0, ZBB) &= \delta(\delta(0, ZB), B)\\ &= \delta(\delta(\delta(0, Z), B), B)\\ &= -;\end{aligned}$$

nicht definiert, da $\delta(0, Z)$

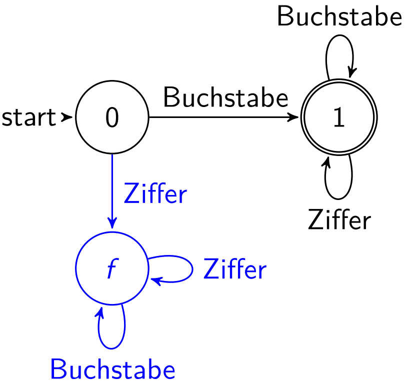
nicht definiert

Folge von Zuständen:

$r_0 = 0$, r_1 nicht definiert

Erweiterung von Übergangsfunktionen - Beispiele

Vervollständigter A_1 :



B:= Buchstabe, Z:=Ziffer

$$
\begin{aligned}
\delta(0, ZBB) &= \delta(\delta(0, ZB), B) \\
&= \delta(\delta(\delta(0, Z), B), B) \\
&= \delta(\delta(f, B), B) \\
&= \delta(f, B) \\
&= f
\end{aligned}
$$

Folge von Zuständen:

$r_0 = 0, r_1 = f, r_2 = f, r_3 = f$

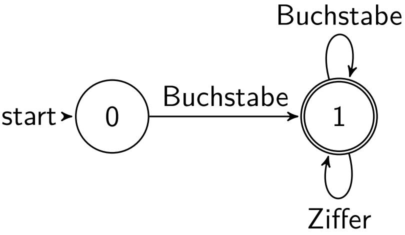
Automaten und Sprachen

Definition 3

Sei $A = (\Sigma, Q, \delta, q_0, F)$ ein deterministischer endlicher Automat.

- Der DFA A akzeptiert das Wort $w \in \Sigma^*$ genau dann, wenn $\delta(q_0, w) \in F$.
- Die Menge
$$L(A) := \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$$
heißt die von A akzeptierte Sprache.

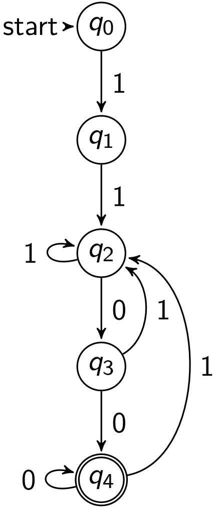
DFA A_1



B:= Buchstabe, Z:=Ziffer

$$L(A_1) = \{w \in \{B, Z\}^* \mid w \text{ beginnt mit } B\}$$
$$= L(B(B \mid Z)^*)$$

DFA A_3:



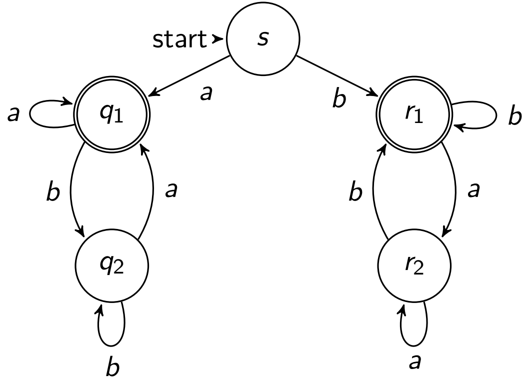
$$L(A_3) = \{w \in \{0,1\}^* \mid w \text{ beginnt mit 11 und endet mit 00}\}$$

$$= L(1^2(0 \mid 1)^*0^2)$$

(als regulärer Ausdruck)

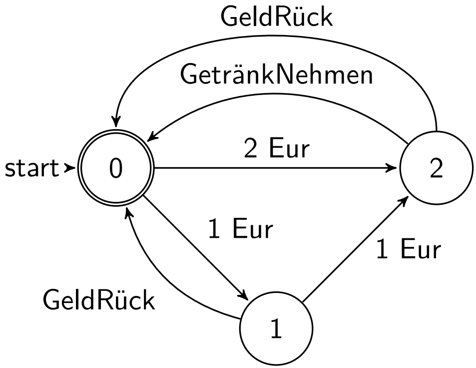
Automaten und Sprachen

DFA A_4**:**



$$L(A_4) = \{w \in \{a,b\}^+ \mid w \text{ beginnt und endet mit demselben Symbol}\}$$

DFA A_2:



$$L(A_2) = L\left(\left(\begin{array}{l}(1\text{ Eur GeldRück}) \mid (1\text{ Eur 1Eur GeldRück}) \mid \\ (1\text{ Eur 1 Eur GetränkNehmen}) \mid (2\text{ Eur GeldRück}) \mid \\ (2\text{ Eur GetränkNehmen})\end{array}\right)^*\right)$$

DFAs, NFAs und reguläre Ausdrücke

Satz 4

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ eine Sprache. Dann sind die folgenden Aussagen äquivalent:

1. *Es gibt einen DFA A mit $L = L(A)$.*
2. *Es gibt einen regulären Ausdruck R mit $L = L(R)$.*

Endliche Automaten mit Ausgabe

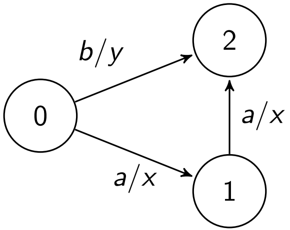
- Man kann mit endlichen Automaten auch Reaktionen der modellierten Maschine in Form von Ausgaben modellieren.
- Erweitern dazu Automaten um ein Ausgabealphabet T und eine Ausgabefunktion.
- Es gibt zwei Varianten:
  1. Mealy-Automaten ordnen Zustandsübergängen eine Ausgabe zu.
  2. Moore-Automaten ordnen Zuständen eine Ausgabe zu.

Mealy-Automaten

Definition 5

Sei $A = (\Sigma, Q, \delta, q_0, F)$ eine endlicher Automat, T ein endliches Alphabet und $\lambda : Q \times \Sigma \to T^*$ eine Funktion. Dann ist A erweitert um T und λ ein Mealy-Automat. T wird Ausgabealphabet und λ Ausgabefunktion genannt. Wir sagen, dass λ den Zustandsübergängen von A Ausgaben zuordnet.

Mealy-Automaten:



$\lambda(0, b) = y, \lambda(0, a) = \lambda(1, a) = x$

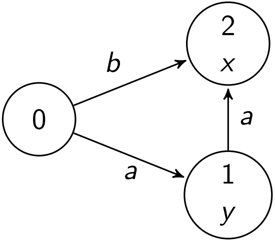
- Bildbereich von λ ist T^*, daher leeres Wort als Ausgabe möglich
- Automat muss nicht bei jedem Übergang mit einer Ausgabe reagieren

Endliche Automaten mit Ausgabe

Definition 6

Sei $A = (\Sigma, Q, \delta, q_0, F)$ eine endlicher Automat, T ein endliches Alphabet und $\mu : Q \rightarrow T^*$ eine Funktion. Dann ist A erweitert um T und μ ein Moore-Automat. T wird Ausgabealphabet und μ Ausgabefunktion genannt. Wir sagen, dass μ den Zuständen von A Ausgaben zuordnet.

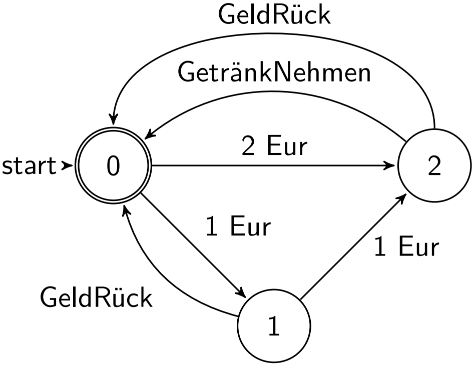
Moore-Automaten:



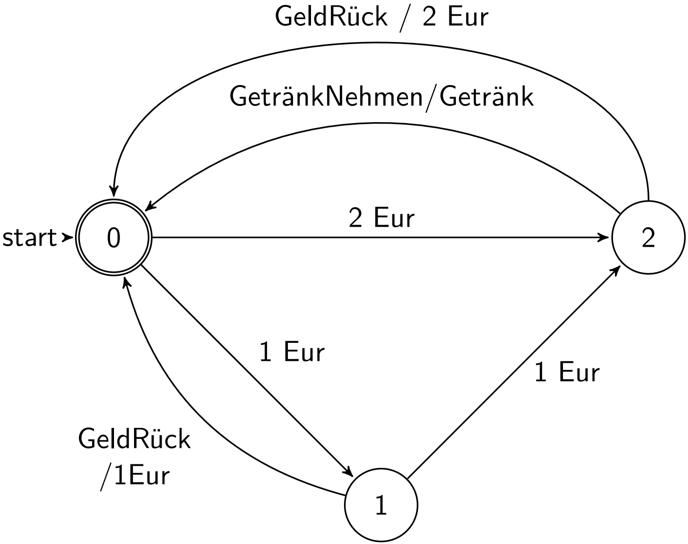
$\mu(1) = y, \mu(2) = x$

- Bildbereich von μ ist T^*, daher leeres Wort als Ausgabe möglich
- Automat muss nicht in jedem Zustand mit einer Ausgabe reagieren
- Moore-Automaten können Ausgaben nicht so fein differenzieren wie Mealy-Automaten

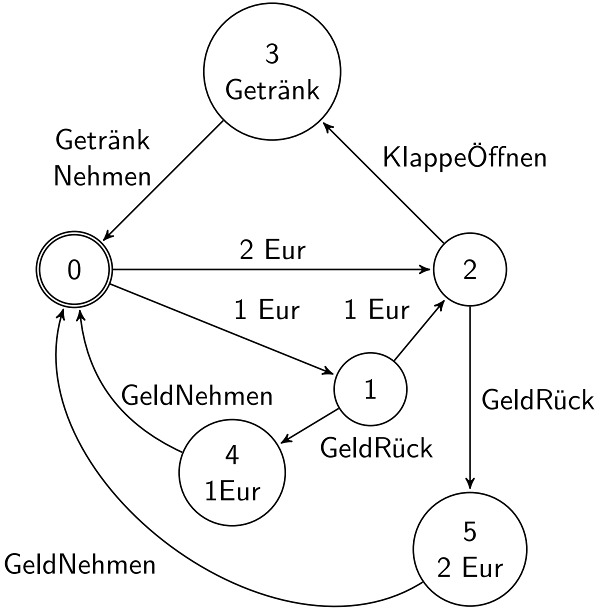
Getränkeautomat



Getränkeautomat mit Ausgabe - Mealy-Automat



Getränkeautomat mit Ausgabe - Moore-Automat



Vergleich Mealy- und Moore-Automaten

- In Mealy-Automaten können Übergänge, die zum selben Zustand führen, unterschiedliche Ausgaben erzeugen.

$\Rightarrow$ Moore-Automaten können Ausgaben nicht so fein differenzieren wie Mealy-Automaten.

- Aber jeder Mealy-Automat kann durch Einführung zusätzlicher Zustände in einen Moore-Automaten transformiert werden, der das gleiche Verhalten hat (siehe Beispiel Getränkeautomat)