TU WIEN Informatics

Endliche Automaten II

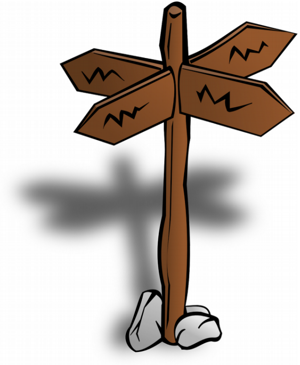
Grundzüge digitaler Systeme (192.134)

Vortrag von: Wolfgang Dvořák

Fakultät für Informatik
TU Wien
E-Mail: gds@list.tuwien.ac.at

Endliche Automaten – Übersicht

1. Endliche Automaten
2. Weitere Typen von Automaten
   - Transducer
   - Mealy-Automaten
   - Moore-Automaten
   - Büchi-Automaten
3. Umsetzung von Automaten als Schaltwerk
   - Schaltnetze vs. Schaltwerke
   - Automaten als Schaltwerke
4. Modellierung



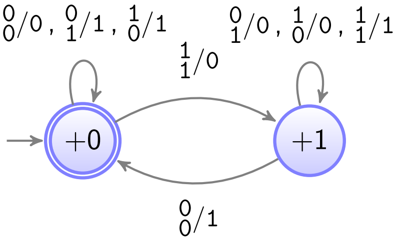
Transducer

Einführendes Beispiel

Wir wollen zwei Binärzahlen addieren:

- Als Eingabe bekommen wir von beiden Binärzahlen jeweils eine Ziffer, beginnend beim LSB.
- Die Berechnung kommt mit endlichen vielen Zuständen aus (Übertrag oder nicht)
- Wir müssen die Ausgabe codieren

$$
\begin{array}{cccccccr}
0 & 0 & 1 & 0 & 1 & 1 & = & 11_{10} \\
0 & 0 & 0 & 1 & 0 & 1 & = & 5_{10} \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & = & 16_{10}
\end{array}
$$



- Ausgabe hängt von Zustand und Eingabe ab
- Bei Übergängen notieren wir Eingabe/Ausgabe

Endlicher Transducer

Endlicher Transducer

- Sehr allgemeines Modell für endliche Automaten mit Ein- und Ausgabe
- Ausgabe kann von Zustand und Eingabe abhängen
- Nichtdeterministisches Verhalten
  - Übergangsrelation
  - ϵ-Übergänge
  - Mehrere Startzustände
- Mehrere Endzustände
- Statt einer Sprache definiert der Automat eine Übersetzungsrelation: Eingaben werden (mögliche) Ausgaben zuordnet.

Transducer

Endlicher Transducer

... wird beschrieben durch ein 6-Tupel $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, I, F \rangle$, wobei

- Q, Σ, F ... siehe DEAs
- Γ ... Ausgabealphabet (*output alphabet*)
- $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times Q$ ... Übergangsrelation
- $I \subseteq Q$ ... Anfangszustände

Erweiterte Übergangsrelation $\delta^* \subseteq Q \times \Sigma^* \times \Gamma^* \times Q$

δ^* ist die kleinste Menge mit folgenden Eigenschaften:

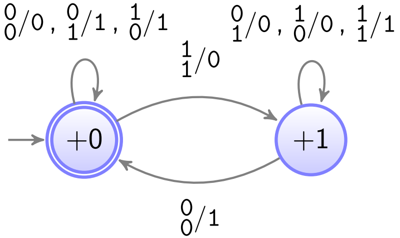
- $(q, \varepsilon, \varepsilon, q) \in \delta^*$ für alle $q \in Q$
- $(q_1, w, w', q_2) \in \delta^*,\ (q_2, s, s', q_3) \in \delta \Longrightarrow \quad (q_1, ws, w's', q_3) \in \delta^* \qquad (s, s' \in \Sigma, w, w' \in \Sigma^*)$.

Übersetzungsrelation $[\mathcal{A}] \subseteq \Sigma^* \times \Gamma^*$

$[\mathcal{A}] = \{ (w, w') \in \Sigma^* \times \Gamma^* \mid (i, w, w', f) \in \delta^* \text{ für ein } i \in I \text{ und ein } f \in F \}$

Beispiel: Binäraddition von rechts nach links

$$
\begin{array}{cccccccr}
0 & 0 & 1 & 0 & 1 & 1 & = & 11_{10} \\
0 & 0 & 0 & 1 & 0 & 1 & = & 5_{10} \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & = & 16_{10}
\end{array}
$$



als Tupel: $\mathcal{A} = \langle \{+0, +1\}, \{\begin{smallmatrix}0\\0\end{smallmatrix}, \begin{smallmatrix}0\\1\end{smallmatrix}, \begin{smallmatrix}1\\0\end{smallmatrix}, \begin{smallmatrix}1\\1\end{smallmatrix}\}, \{0, 1\}, \delta, \{+0\}, \{+0\} \rangle$, wobei

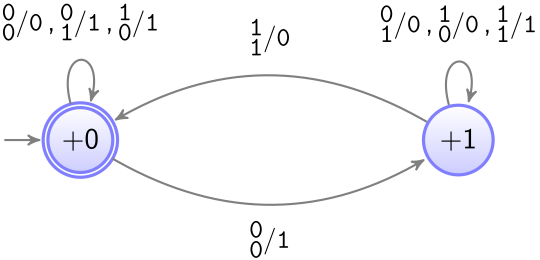
δ	$\begin{smallmatrix}0\\0\end{smallmatrix}$	$\begin{smallmatrix}0\\1\end{smallmatrix}$	$\begin{smallmatrix}1\\0\end{smallmatrix}$	$\begin{smallmatrix}1\\1\end{smallmatrix}$
$+0$	$\{(0, +0)\}$	$\{(1, +0)\}$	$\{(1, +0)\}$	$\{(0, +1)\}$
$+1$	$\{(1, +0)\}$	$\{(0, +1)\}$	$\{(0, +1)\}$	$\{(1, +1)\}$

$$[\mathcal{A}] = \{ (\varepsilon, \varepsilon), (\begin{smallmatrix}0\\0\end{smallmatrix}, 0), (\begin{smallmatrix}0\\1\end{smallmatrix}, 1), (\begin{smallmatrix}1\\0\end{smallmatrix}, 1), (\begin{smallmatrix}0\,0\\0\,0\end{smallmatrix}, 00), (\begin{smallmatrix}0\,0\\0\,1\end{smallmatrix}, 01), \ldots, (\begin{smallmatrix}1\,0\\1\,0\end{smallmatrix}, 01), \ldots, (\begin{smallmatrix}1\,0\,1\,0\\1\,0\,1\,0\end{smallmatrix}, 0101), \ldots \}$$

Hinweis: Die Ein- und Ausgabe beginnen mit den letzten Ziffern. Die Addition $0100 + 0111 = 1011$ resultiert in der Eingabe $\begin{smallmatrix}0\,0\,1\,0\\1\,1\,1\,0\end{smallmatrix}$ und der Ausgabe 1101.

Binäraddition von links nach rechts

$$
\begin{array}{cccccc|l}
0 & 0 & 1 & 0 & 1 & 1 & = 11_{10} \\
0 & 0 & 0 & 1 & 0 & 1 & = 5_{10} \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & = 16_{10}
\end{array}
$$

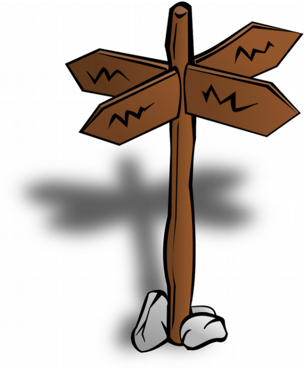


$+0$ … kein Übertrag
$+1$ … Übertrag

Achtung, Indeterminismus! Zustand „$+0$“ besitzt bei Eingabe $\genfrac{}{}{0pt}{}{0}{0}$ zwei Folgezustände, ebenso Zustand „$+1$“ bei Eingabe $\genfrac{}{}{0pt}{}{1}{1}$.

Endliche Automaten – Übersicht

1. Endliche Automaten
2. Weitere Typen von Automaten
   - Transducer
   - Mealy-Automaten
   - Moore-Automaten
   - Büchi-Automaten
3. Umsetzung von Automaten als Schaltwerk
   - Schaltnetze vs. Schaltwerke
   - Automaten als Schaltwerke
4. Modellierung



Mealy-Automaten

Mealy-Automaten

- deterministische Automaten
- mit Ausgabe
  - Ausgabe hängt von Zustand und Eingabe ab
- Ein Anfangszustand
- Keine expliziten Endzustände
  - Alle Zustände verhalten sich wie Endzustände

Mealy-Automaten

Mealy-Automat

... wird beschrieben durch ein 6-Tupel $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, \gamma, q_0 \rangle$, wobei

- Q, Σ, δ, q_0 ... siehe DEAs
- Γ ... Ausgabealphabet (*output alphabet*)
- $\gamma \colon Q \times \Sigma \to \Gamma$ ... Ausgabefunktion (*output function*)

Erweiterte Übergangsfunktion $\delta^* \colon Q \times \Sigma^* \to Q$ siehe DEA.

Erweiterte Ausgabefunktion $\gamma^* \colon Q \times \Sigma^* \to \Gamma^*$

$$\gamma^*(q, \varepsilon) = \varepsilon \qquad \text{für alle } q \in Q,\ s \in \Sigma,\ w \in \Sigma^*$$

$$\gamma^*(q, sw) = \gamma(q, s) \cdot \gamma^*(\delta(q, s), w)$$

Übersetzungsfunktion $[\mathcal{A}] \colon \Sigma^* \to \Gamma^*$

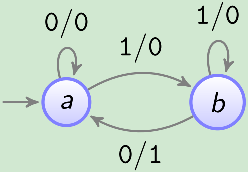
$$[\mathcal{A}](w) = \gamma^*(q_0, w)$$

Mealy-Automaten sind ein Spezialfall von Transducern:

- Nur ein Anfangszustand: $I = \{q_0\}$
- Die Übergangsrelation ist deterministisch:
  - Der Folgezustand $\delta(q, s)$ ist eindeutig durch q und s bestimmt.
  - Keine ε-Übergänge
- Relationstupel: $(q, s, \gamma(q, s), \delta(q, s))$
- Alle Zustände sind Endzustände: $F = Q$

Detektor für fallende Flanken

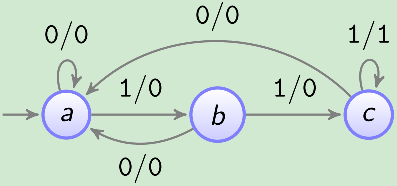
Ausgabe 1, wenn in der Eingabe ein Wechsel von 1 auf 0 stattfindet.



$$\begin{array}{r l} w: & 00110001010\cdots \\ \hline [\mathcal{A}](w): & 000010001010\cdots \end{array}$$

Detektor für 111-Blöcke

Ausgabe 1, wenn in der Eingabe drei 1er aufeinander folgen.



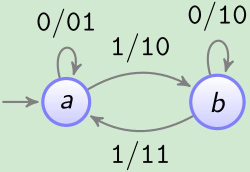
$$\begin{array}{rl} w: & 00110111110101110\cdots \\ \hline [\mathcal{A}](w): & 00000001110000010\cdots \end{array}$$

(1 : 2)-(0, 1)-RLL-Encoder als Mealy-Automat

$\mathcal{A} = \langle \{a, b\}, \{0, 1\}, \{01, 10, 11\}, \delta, \gamma, a \rangle$

δ	0	1
a	a	b
b	b	a

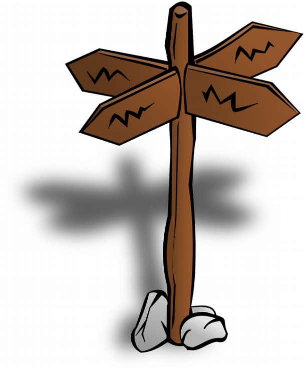
γ	0	1
a	01	10
b	10	11



$$\begin{array}{rcccccccccc} w: & \varepsilon & 0 & 1 & 00 & 10 & 01 & 11 & 000 & 100 & \cdots \\ \hline [\mathcal{A}](w): & \varepsilon & 01 & 10 & 0101 & 1010 & 0110 & 1011 & 010101 & 101010 & \cdots \end{array}$$

Endliche Automaten – Übersicht

1. Endliche Automaten
2. Weitere Typen von Automaten
   - Transducer
   - Mealy-Automaten
   - Moore-Automaten
   - Büchi-Automaten
3. Umsetzung von Automaten als Schaltwerk
   - Schaltnetze vs. Schaltwerke
   - Automaten als Schaltwerke
4. Modellierung



Moore-Automaten

Moore-Automaten

- deterministische Automaten
- mit Ausgabe
  - Ausgabe hängt vom momentanen Zustand ab
- Ein Anfangszustand
- Keine expliziten Endzustände
  - Alle Zustände verhalten sich wie Endzustände

Moore-Automat

... wird beschrieben durch ein 6-Tupel $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, \gamma, q_0 \rangle$, wobei

- Q, Σ, δ, q_0 ... siehe DEAs
- Γ ... Ausgabealphabet (*output alphabet*)
- $\gamma \colon Q \to \Gamma$ ... Ausgabefunktion (*output function*)

(Mealy: $\gamma \colon Q \times \Sigma \to \Gamma$)

Erweiterte Übergangsfunktion $\delta^* \colon Q \times \Sigma^* \to Q$ siehe DEA.

Erweiterte Ausgabefunktion $\gamma^* \colon Q \times \Sigma^* \to \Gamma^*$

$$\gamma^*(q, \varepsilon) = \gamma(q) \qquad \text{für alle } q \in Q,\ s \in \Sigma,\ w \in \Sigma^*$$
$$\gamma^*(q, sw) = \gamma(q) \cdot \gamma^*(\delta(q, s), w)$$

(Mealy: $\gamma^*(q, sw) = \gamma(q, s) \cdot \gamma^*(\delta(q, s), w)$ und $\gamma^*(q, \varepsilon) = \varepsilon$)

Übersetzungsfunktion $[\mathcal{A}] \colon \Sigma^* \to \Gamma^*$

$$[\mathcal{A}](w) = \gamma^*(q_0, w)$$

Moore-Automaten sind (fast) ein Spezialfall von Transducern:

- Nur ein Anfangszustand: $I = \{q_0\}$
- Alle Übergange in einen Zustand geben dasselbe Symbol aus.
- Die Übergangsrelation ist deterministisch:
  - Der Folgezustand $\delta(q, s)$ ist eindeutig durch q und s bestimmt.
  - Keine ε-Übergänge
- Relationstupel: $(q, s, \gamma(q), \delta(q, s))$
- Alle Zustände sind Endzustände: $F = Q$

Vergleich von Moore- und Mealy-Automaten

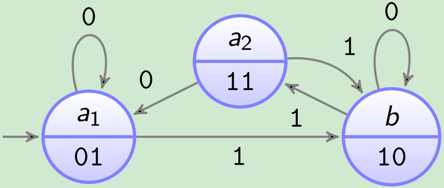
- Die Ausgabe erfolgt
  - ... bei Moore-Automaten durch den momentanen Zustand.
  - ... bei Mealy-Automaten beim Zustandswechsel, der durch Ursprungszustand und Eingabe festgelegt ist.
- Moore- und Mealy-Automaten besitzen dieselbe Ausdrucksstärke, sind aber schwächer als Transducer.
- Moore-Automaten haben in der Regel mehr Zustände als Mealy-Automaten.

$(1:2)$-$(0,1)$-RLL-Encoder als Moore-Automat

$\mathcal{A} = \langle \{a_1, a_2, b\}, \{0,1\}, \{01, 10, 11\}, \delta, \gamma, a\rangle$

δ	0	1
a_1	a_1	b
a_2	a_1	b
b	b	a_2

γ	
a_1	01
a_2	11
b	10

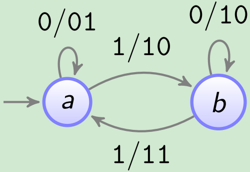


w:	ε	0	1	00	10	01	$\cdots$	100	$\cdots$
$[\mathcal{A}](w)$:	01	0101	0110	010101	011010	0110	$\cdots$	01101010	$\cdots$

Zum Vergleich: Mealy-Automat

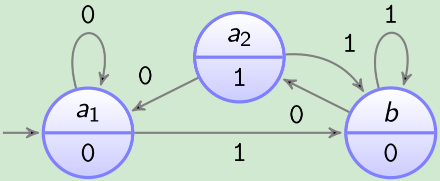
δ	0	1
a	a	b
b	b	a

γ	0	1
a	01	10
b	10	11



Detektor für fallende Flanken

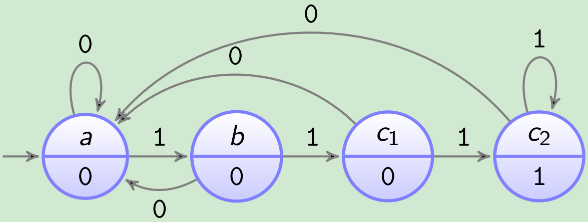
Ausgabe 1, wenn in der Eingabe ein Wechsel von 1 auf 0 stattfindet.



$$\begin{array}{rl} w\colon & 00110001010\cdots \\ \hline [\mathcal{A}](w)\colon & 000001000101\cdots \end{array}$$

Detektor für 111-Blöcke

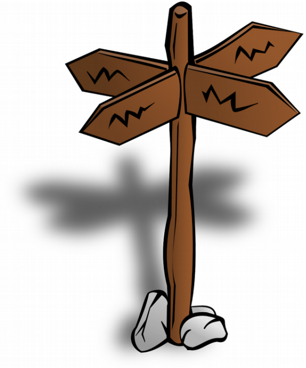
Ausgabe 1, wenn in der Eingabe drei 1er aufeinander folgen.



$$\begin{array}{rl} w\colon & 00110111110101110\cdots \\ \hline [\mathcal{A}](w)\colon & 000000001110000010\cdots \end{array}$$

Endliche Automaten – Übersicht

1. Endliche Automaten
2. Weitere Typen von Automaten
   - Transducer
   - Mealy-Automaten
   - Moore-Automaten
   - Büchi-Automaten
3. Umsetzung von Automaten als Schaltwerk
   - Schaltnetze vs. Schaltwerke
   - Automaten als Schaltwerke
4. Modellierung



Büchi-Automaten

Deterministischer Büchi-Automat

... wird beschrieben durch ein 5-Tupel $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$, wobei

- Q, Σ, δ, q_0, F wie bei DEAs definiert sind.

Σ^ω ... Menge aller *unendlichen* Wörter ($= \omega$-Wörter) über Σ

Akzeptanz von Wörtern

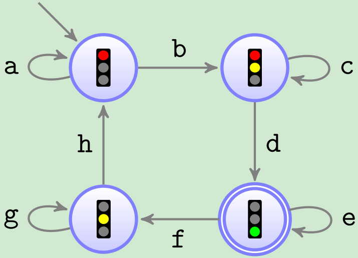
Ein deterministischer Büchi-Automat $\mathcal{A}$ akzeptiert ein Wort $s_1 s_2 s_3 \cdots \in \Sigma^\omega$, wenn es Zustände $q_0, q_1, q_2, q_3, \dots \in Q$ gibt, sodass

- $q_0 \in Q$ der Startzustand ist,
- $\delta(q_{i-1}, s_i) = q_i$ für alle $i \in \mathbb{N}$ gilt und
- es unendlich viele i gibt, sodass q_i ein Endzustand ist ($q_i \in F$).

Akzeptierte/Generierte Sprache

$\mathcal{L}(\mathcal{A}) = \{\, w \in \Sigma^\omega \mid w \text{ wird von } \mathcal{A} \text{ akzeptiert} \,\}$

Verkehrsampel



Liveness: Der Automat akzeptiert genau jene Wörter aus $\{\mathtt{a}, \ldots, \mathtt{h}\}^\omega$, bei denen es immer wieder grün wird.

Nichtdeterministischer Büchi-Automat

... wird beschrieben durch ein 5-Tupel $\mathcal{A} = \langle Q, \Sigma, \delta, I, F \rangle$, wobei

- Q, Σ, F ... siehe DEA
- $I \subseteq Q$ ... Anfangszustände
- $\delta \subseteq Q \times \Sigma \times Q$ ... Übergangsrelation

Akzeptanz von Wörtern

Ein nichtdeterministischer Büchi-Automat $\mathcal{A}$ akzeptiert ein Wort $s_1 s_2 s_3 \cdots \in \Sigma^\omega$, wenn es Zustände $q_0, q_1, q_2, q_3, \cdots \in Q$ gibt, sodass

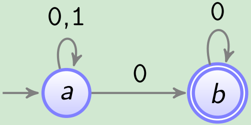
- $q_0 \in Q$ ein Startzustand ist ($q_0 \in I$),
- $(q_{i-1}, s_i, q_i) \in \delta$ für alle $i \in \mathbb{N}$ gilt und
- es unendlich viele i gibt, sodass q_i ein Endzustand ist ($q_i \in F$).

Akzeptierte/Generierte Sprache

$\mathcal{L}(\mathcal{A}) = \{\, w \in \Sigma^\omega \mid w \text{ wird von } \mathcal{A} \text{ akzeptiert} \,\}$

Nur endlich viele 1er

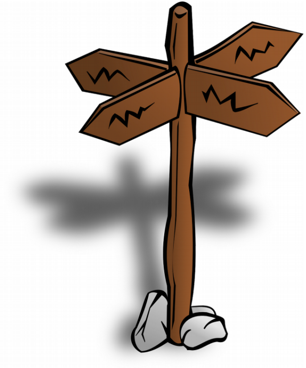
Gesucht: Ein Büchi-Automat, der genau jene ω-Wörter über $\{0, 1\}$ akzeptiert, die nur eine endliche Anzahl an 1ern enthalten.



Kein deterministischer Büchi-Automaten akzeptiert diese Sprache.
$\Longrightarrow$ Nichtdeterministische Büchi-Automaten sind ausdrucksstärker als deterministische!

Endliche Automaten – Übersicht

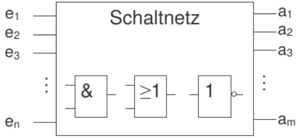
1. Endliche Automaten
2. Weitere Typen von Automaten
   - Transducer
   - Mealy-Automaten
   - Moore-Automaten
   - Büchi-Automaten
3. Umsetzung von Automaten als Schaltwerk
   - Schaltnetze vs. Schaltwerke
   - Automaten als Schaltwerke
4. Modellierung



Schaltnetze vs. Schaltwerke

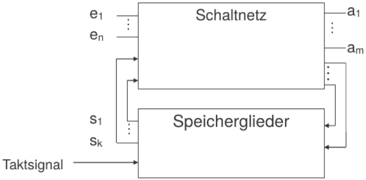
Schaltnetz

- Eine Funktionseinheit zum Verarbeiten von Schaltvariablen,deren Wert am Ausgang zu irgendeinem Zeitpunkt nur vom Wert am Eingang zu diesem Zeitpunkt abhängt.“ [DIN44300,ISO/IEC2382]



Schaltwerk

- Eine Funktionseinheit zum Verarbeiten von Schaltvariablen, wobei der Wert am Ausgang zu einem bestimmten Zeitpunkt abhängt von den Werten am Eingang zu diesem und endlich vielen vorangegangenen Zeitpunkten. [DIN44300,ISO/IEC2382]

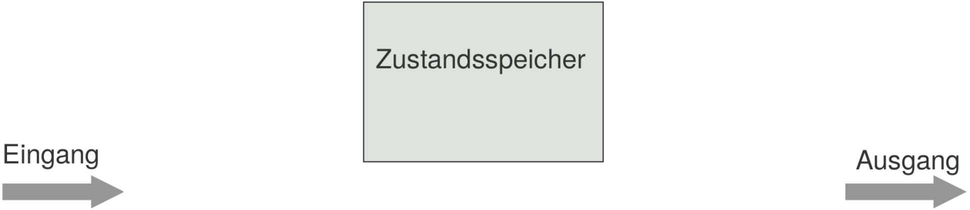


Schaltwerke

- Konzept - Eingänge werden auf Ausgänge abgebildet

Schaltwerke

- Dabei wird ein interner Zustand gespeichert

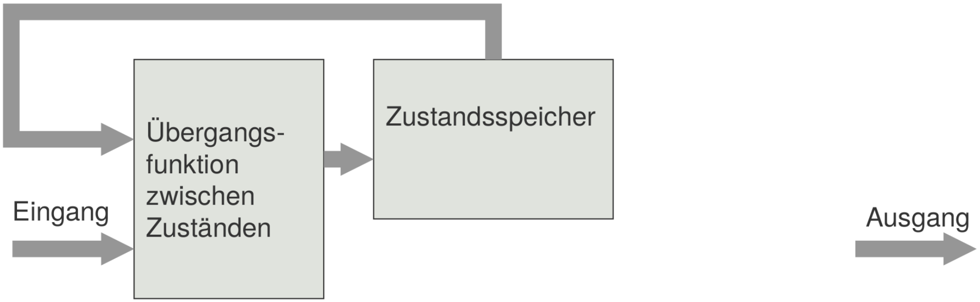


Schaltwerke

- Zustandswechsel hängen von den Eingängen …

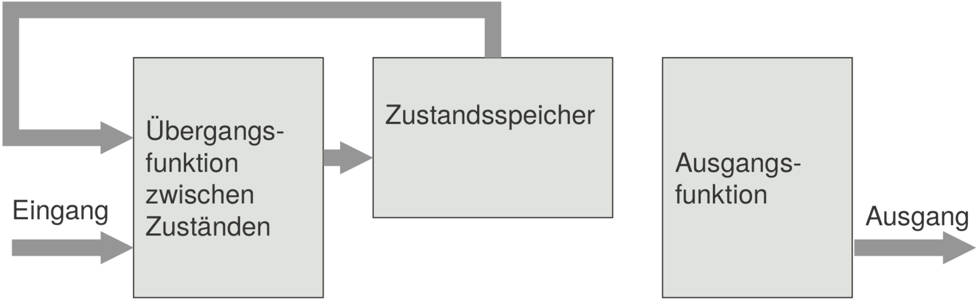
Schaltwerke

- … und vom aktuellen Zustand ab



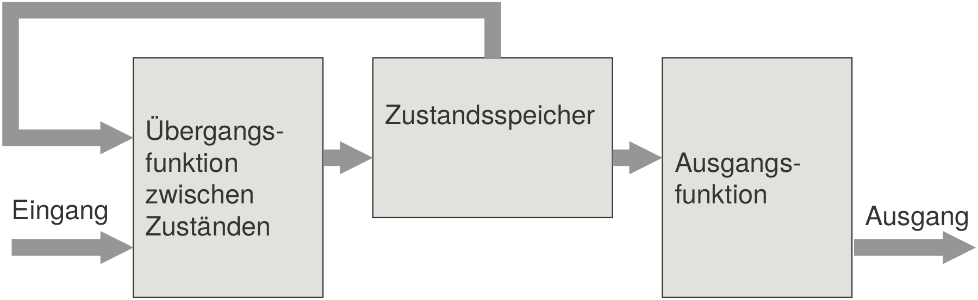
Schaltwerke

- Ausgänge werden von Ausgangsfunktion gesteuert



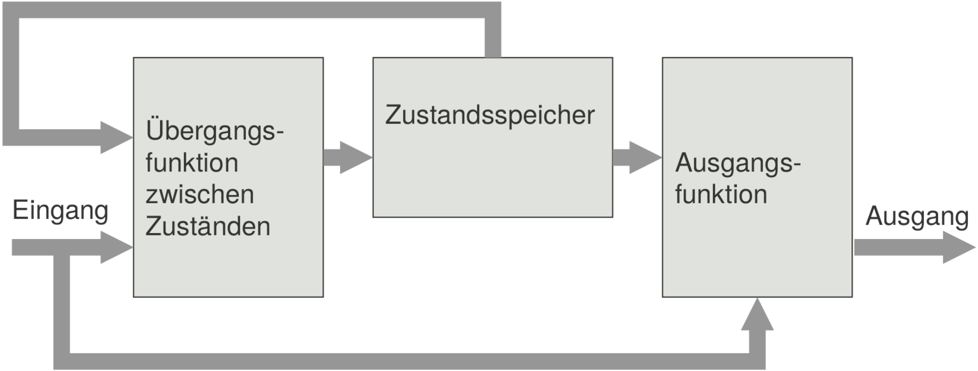
Schaltwerke

- Ausgangsfunktion hängt vom Zustand ab
  $\rightarrow$ Moore-Schaltwerk



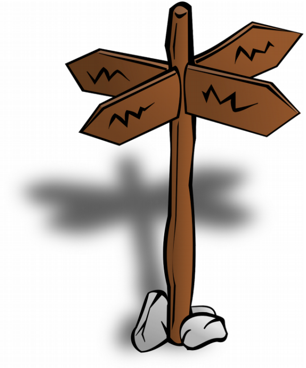
Schaltwerke

- Ausgangsfunktion hängt vom Zustand und Eingang ab
  → Mealy-Schaltwerk



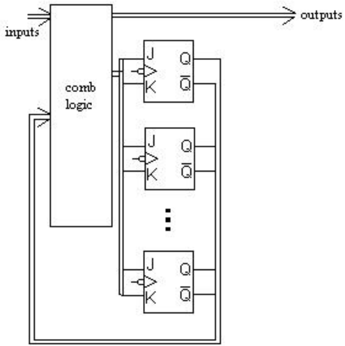
Endliche Automaten – Übersicht

1. Endliche Automaten
2. Weitere Typen von Automaten
   - Transducer
   - Mealy-Automaten
   - Moore-Automaten
   - Büchi-Automaten
3. Umsetzung von Automaten als Schaltwerk
   - Schaltnetze vs. Schaltwerke
   - Automaten als Schaltwerke
4. Modellierung

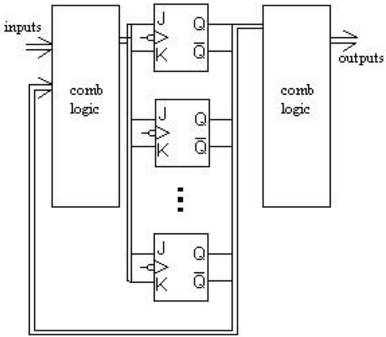


Automaten als Schaltwerke

Mealy-Schaltwerk



Moore-Schaltwerk



- „inputs“ stammen aus dem Eingabealphabet Σ
- „outputs“ stammen aus dem Ausgabealphabet Γ
- Flip-Flops speichern den Zustand aus Q; Reset: Anfangszustand q_0
- „combination logic“: realisiert die Übergangsfunktionen δ und γ.

Graphiken: Ken Reid, Indiana University – Purdue University Indianapolis

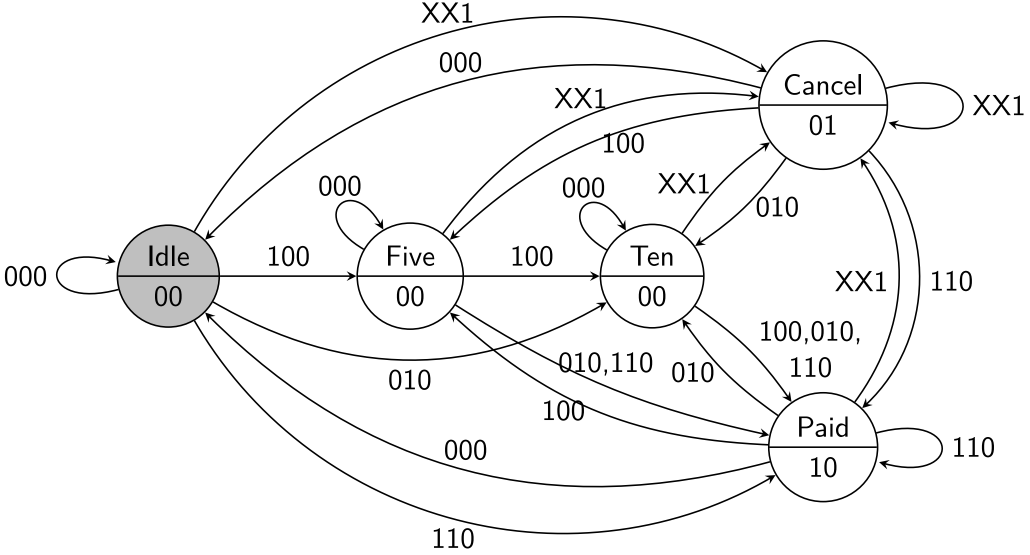
Münzautomat: Aufgabenstellung

- Eingabe:
  - Einwurfschlitze für 5 (F) und 10 (Z) Cent Münzen
    - gleichzeitiger Einwurf möglich
  - Geldrückgabetaste (R)
    - Einwurf und gleichzeitiges Bestätigen von R möglich

- Ausgabe:
  - Ware (W) kostet 15 Cent
    - bei Überbezahlung wird gesamtes Geld kassiert
  - bei Storno (St) wird eingeworfenes Geld rückerstattet

Münzautomat: Entwurf des Zustandsgraphen

Eingabebitmuster: FZR, Ausgabebitmuster: WSt



Münzautomat: Zustandscodierung

Wir müssen den Zustand Binärcodieren um diesen speichern zu können:

1-hot Codierung

	s_1	s_2	s_3	s_4	s_5
Idle	1	0	0	0	0
Five	0	1	0	0	0
Ten	0	0	1	0	0
Paid	0	0	0	1	0
Cancel	0	0	0	0	1

Dichte Codierung

	d_1	d_2	d_3
Idle	0	0	0
Five	0	0	1
Ten	0	1	0
Paid	0	1	1
Cancel	1	0	0

Münzautomat: Übergangsfunktion als boolesche Funktionen

Übergangsfunktion

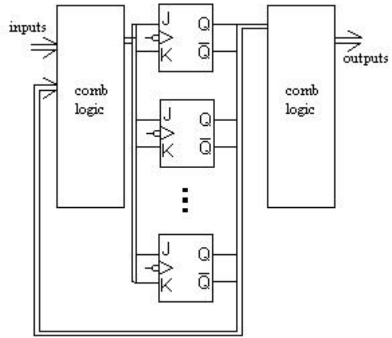
s_1	s_2	s_3	s_4	s_5	F	Z	R	s_1'	s_2'	s_3'	s_4'	s_5'
1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	1	0	0	0	0	1
1	0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0	1
1	0	0	0	0	1	1	0	0	0	0	1	0
1	0	0	0	0	1	1	1	0	0	0	0	1
0	1	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1
			⋮									
0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1
			⋮									

Ausgabefunktion

s_1	s_2	s_3	s_4	s_5	ST	W
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	1	0
0	0	0	0	1	0	1

Münzautomat: Fazit

- Wir verwenden 5 JK-Flipflops um die Zustände zu speichern
- Sei s_i der gespeicherte Wert der am Ausgang Q des i-ten JK-Flipflops anliegt
- Sei s_i' der Wert der bei der nächsten positiven Taktflanke in das i-te JK-Flipflop gespeichert wird.
- Die Übergangsfunktion können wir jetzt als Schaltnetz mit input s_1, s_2, s_3, s_4, s_5, F, Z, R und output s_1', s_2', s_3', s_4', s_5' realisieren.
- Die Ausgabefunktion können wir jetzt als Schaltnetz mit input s_1, s_2, s_3, s_4, s_5, und output ST, W realisieren.



Zusammenfassung

- Transducer
  - als endliche Automaten mit Ein- & Ausgabe
- Deterministische Transducer
  - Mealy-Automaten: Ausgabe hängt von Eingabe und Zustand ab
  - Moore-Automaten: Ausgabe hängt nur von Zustand ab
- Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk