

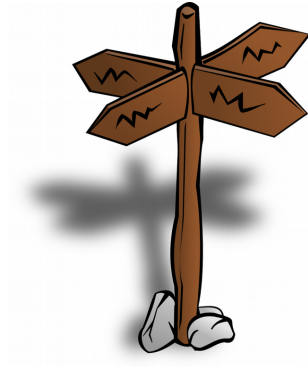
Endliche Automaten II

Grundzüge digitaler Systeme

Vortrag von: Wolfgang Dvořák

Endliche Automaten

- Endliche Automaten
- Weitere Typen von Automaten
 - **Transducer**
 - Mealy-Automaten
 - Moore-Automaten
 - Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk
- Modellierungsbeispiele



Transducer

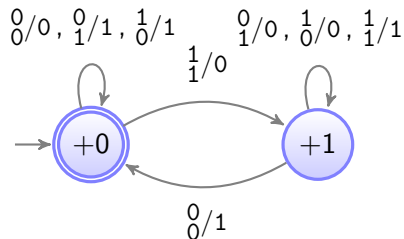
Einführendes Beispiel

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

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

←



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

Endlicher Transducer

- 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 $q \in Q$
- $(q_1, w, w', q_2) \in \delta^*, (q_2, s, s', q_3) \in \delta \implies (q_1, ws, w's', q_3) \in \delta^* \quad (s, s' \in \Sigma, w, w' \in \Sigma^*)$.

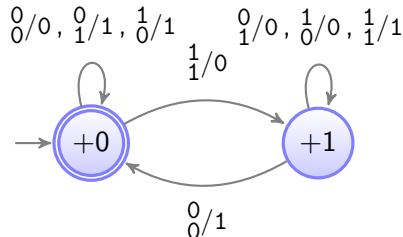
Übersetzungsrelation $[\mathcal{A}] \subseteq \Sigma^* \times \Gamma^*$

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

Beispiel: Binäraddition von rechts nach links

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

←



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

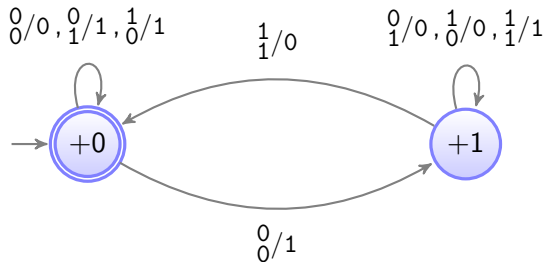
δ	$\overset{0}{0}$	$\overset{0}{1}$	$\overset{1}{0}$	$\overset{1}{1}$
+0	$\{(0, +0)\}$	$\{(1, +0)\}$	$\{(1, +0)\}$	$\{(0, +1)\}$
+1	$\{(1, +0)\}$	$\{(0, +1)\}$	$\{(0, +1)\}$	$\{(1, +1)\}$

$$[\mathcal{A}] = \{ (\varepsilon, \varepsilon), (\overset{0}{0}, 0), (\overset{0}{1}, 1), (\overset{1}{0}, 1), (\overset{0}{0}\overset{0}{0}, 00), (\overset{0}{0}\overset{0}{1}, 01), \dots, (\overset{1}{1}\overset{0}{0}, 01), \dots, (\overset{1}{1}\overset{0}{0}\overset{1}{0}, 0101), \dots \}$$

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

Binäraddition von links nach rechts

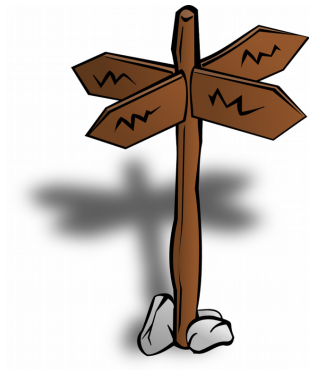
$$\begin{array}{rcccccc} 0 & 0 & 1 & 0 & 1 & 1 & = 11_{10} \\ 0 & 0 & 0 & 1 & 0 & 1 & = 5_{10} \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & = 16_{10} \end{array}$$



+0 ... kein Übertrag
+1 ... Übertrag

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

- Transducer
 - Mealy-Automaten
 - Moore-Automaten
- Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk
- Modellierungsbeispiele



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*)
- $\gamma: Q \times \Sigma \mapsto \Gamma$... Ausgabefunktion (*output function*)

Erweiterte Übergangsfunktion $\delta^*: Q \times \Sigma^* \mapsto Q$ siehe DEA.

Erweiterte Ausgabefunktion $\gamma^*: Q \times \Sigma^* \mapsto \Gamma^*$

$$\gamma^*(q, \varepsilon) = \varepsilon$$

für alle $q \in Q, s \in \Sigma, w \in \Sigma^*$

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

Übersetzungsfunktion $[\mathcal{A}]: \Sigma^* \mapsto \Gamma^*$

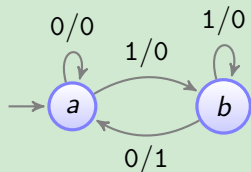
$$[\mathcal{A}](w) = \gamma^*(q_0, w)$$

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

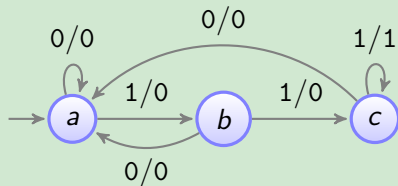
Ausgabe 1, wenn in der Eingabe ein Wechsel von 1 auf 0 stattfindet.



$w:$	00110001010...
$[\mathcal{A}](w):$	000010001010...

Detektor für 111-Blöcke

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



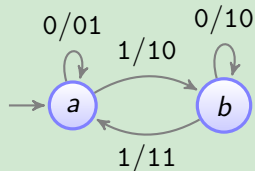
w : 001101111110101110...
 $[\mathcal{A}](w)$: 00000001110000010...

(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



w : ε 0 1 00 10 01 11 000 100 ...
 $[\mathcal{A}](w)$: ε 01 10 0101 1010 0110 1011 010101 101010 ...

- Transducer
 - Mealy-Automaten
 - Moore-Automaten
- Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk
- Modellierungsbeispiele



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*)
- $\gamma: Q \mapsto \Gamma$... Ausgabefunktion (*output function*)

(Mealy: $\gamma: Q \times \Sigma \mapsto \Gamma$)

Erweiterte Übergangsfunktion $\delta^*: Q \times \Sigma^* \mapsto Q$ siehe DEA.

Erweiterte Ausgabefunktion $\gamma^*: Q \times \Sigma^* \mapsto \Gamma^*$

$$\gamma^*(q, \varepsilon) = \gamma(q)$$

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 $[\mathcal{A}]: \Sigma^* \mapsto \Gamma^*$

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

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

- Nur ein Anfangszustand: $I = \{q_0\}$
- Alle Übergänge 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: $F = Q$

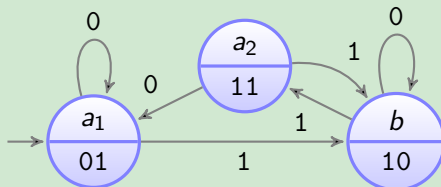
Vergleich von Moore- und Mealy-Automaten

- 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 Ausdruckskraft, 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

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

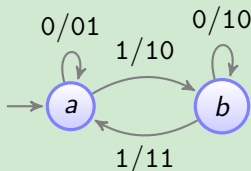
δ	0	1	γ	
a_1	a_1	b	a_1	01
a_2	a_1	b	a_2	11
b	b	a_2	b	10



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

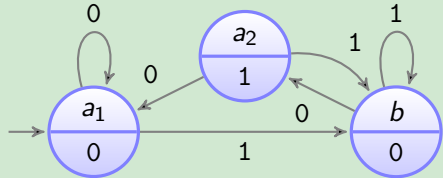
Zum Vergleich: Mealy-Automat

δ	0	1	γ	0	1
a	a	b	a	01	10
b	b	a	b	10	11



Detektor für fallende Flanken

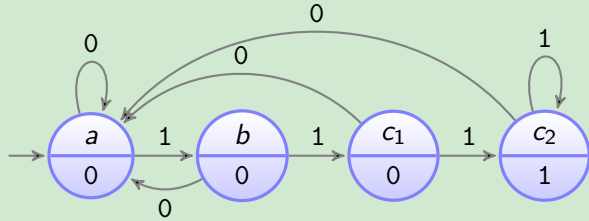
Ausgabe 1, wenn in der Eingabe ein Wechsel von 1 auf 0 stattfindet.



$$\begin{array}{rcl} w: & 00110001010\dots \\ \hline [\mathcal{A}](w): & 000001000101\dots \end{array}$$

Detektor für 111-Blöcke

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



$$\begin{array}{rcl} w: & 00110111110101110\dots \\ \hline [\mathcal{A}](w): & 000000001110000010\dots \end{array}$$

- Transducer
 - Mealy-Automaten
 - Moore-Automaten
- Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk
- Modellierungsbeispiele



Büchi-Automaten

Deterministischer Büchi-Automat

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

- $Q, \Sigma, \delta, q_0, F$ wie bei DEAs definiert sind.

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

Akzeptanz von Wörtern

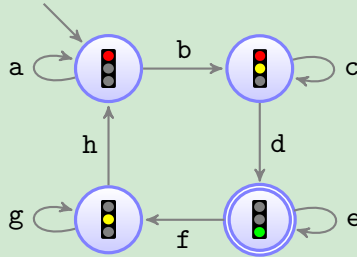
Ein deterministischer Büchi-Automat \mathcal{A} akzeptiert ein Wort $s_1 s_2 s_3 \dots \in \Sigma^\omega$, wenn es Zustände $q_0, q_1, q_2, q_3, \dots \in Q$ gibt, sodass

- $q_0 \in Q$ 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 $\{a, \dots, h\}^\omega$, 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 $s_1 s_2 s_3 \dots \in \Sigma^\omega$, wenn es Zustände $q_0, q_1, q_2, q_3, \dots \in Q$ gibt, sodass

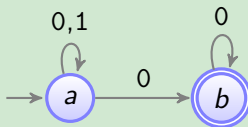
- $q_0 \in Q$ ein Startzustand ist ($q_0 \in I$),
- $(q_{i-1}, s_i, q_i) \in \delta$ für alle $i \in \mathbb{N}$ gilt und
- es unendlich viele i gibt, sodass q_i ein Endzustand ist ($q_i \in F$).

Akzeptierte/Generierte Sprache

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

Nur endlich viele 1er

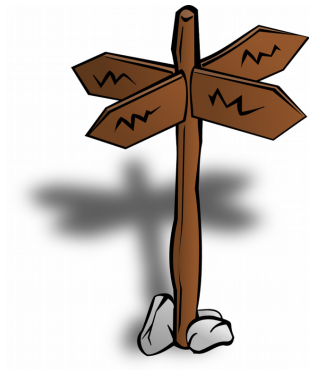
Gesucht: Ein Büchi-Automat, der genau jene ω -Wörter über $\{0, 1\}$ akzeptiert, die nur eine endliche Anzahl an 1ern enthalten.



Kein deterministischer Büchi-Automat akzeptiert diese Sprache.

\Rightarrow Nichtdeterministische Büchi-Automaten sind ausdrucksstärker als deterministische!

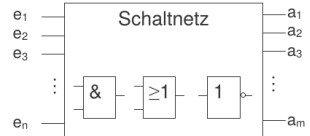
- Transducer
 - Mealy-Automaten
 - Moore-Automaten
- Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk



Schaltnetze vs. Schaltwerke

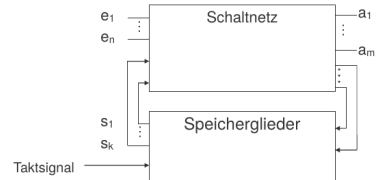
Schaltnetz

- 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

Eingang

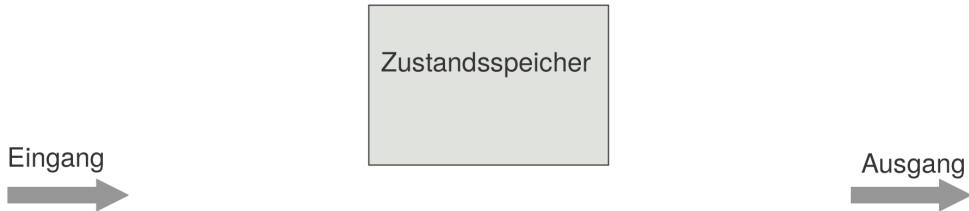


Ausgang



Schaltwerke

- Dabei wird ein interner Zustand gespeichert

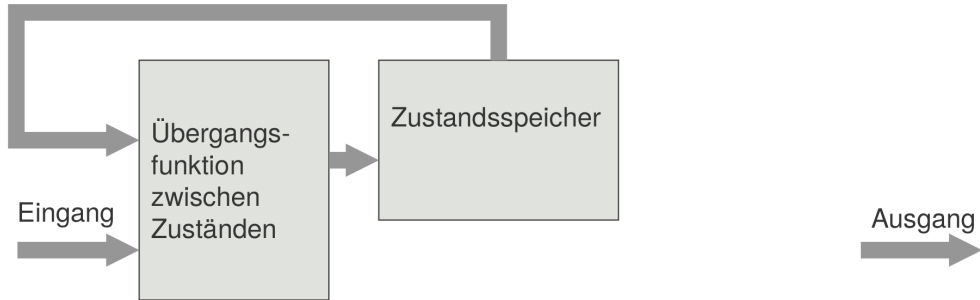


- Zustandswechsel hängen von den Eingängen ...

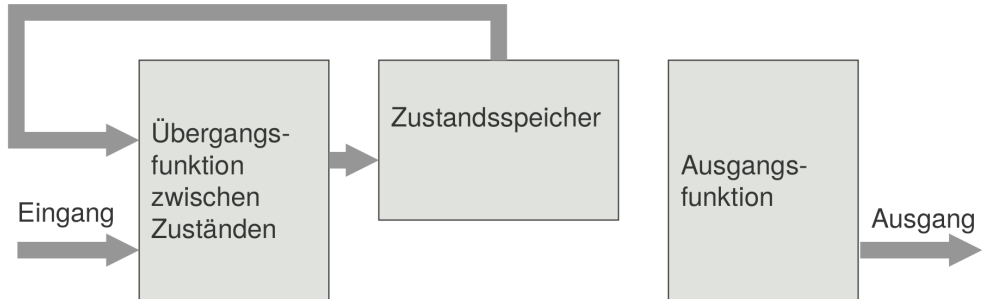


Schaltwerke

- ... und vom aktuellen Zustand ab

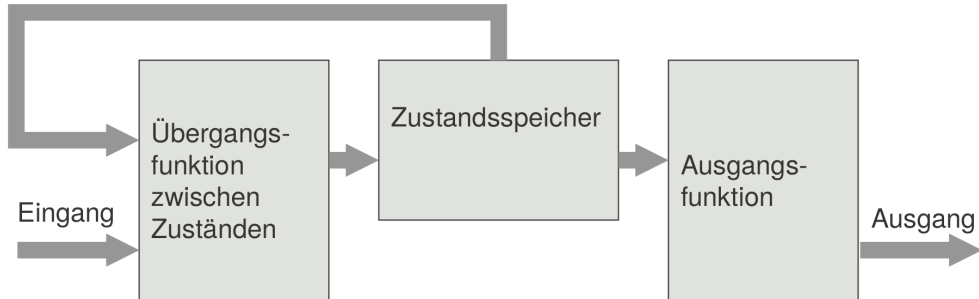


- Ausgänge werden von Ausgangsfunktion gesteuert



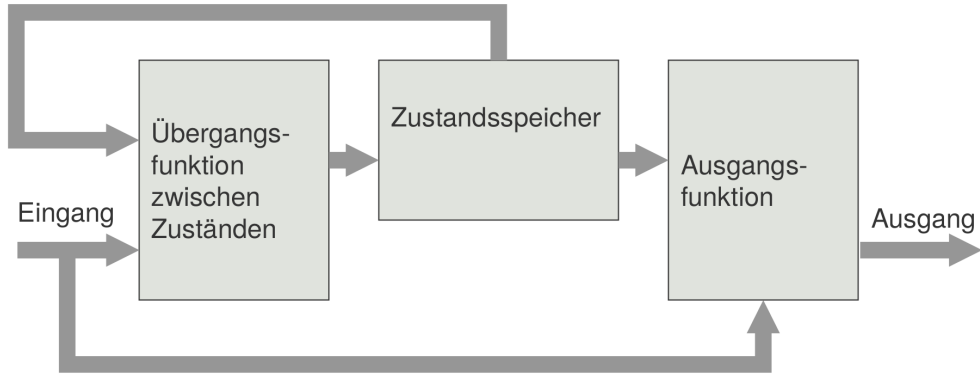
Schaltwerke

- Ausgangsfunktion hängt vom Zustand ab
→ Moore-Schaltwerk



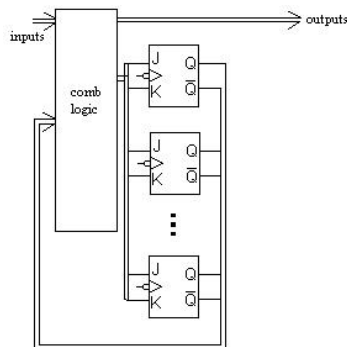
Schaltwerke

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

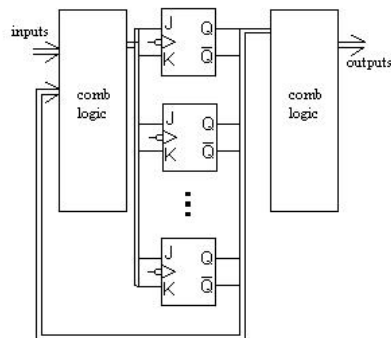


Automaten als Schaltwerke

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 .

Münzautomat: Aufgabenstellung

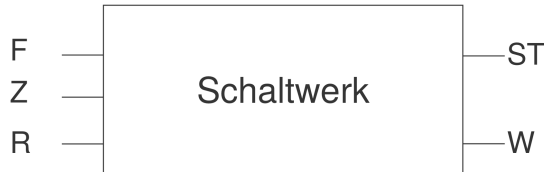
■ Eingabe:

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



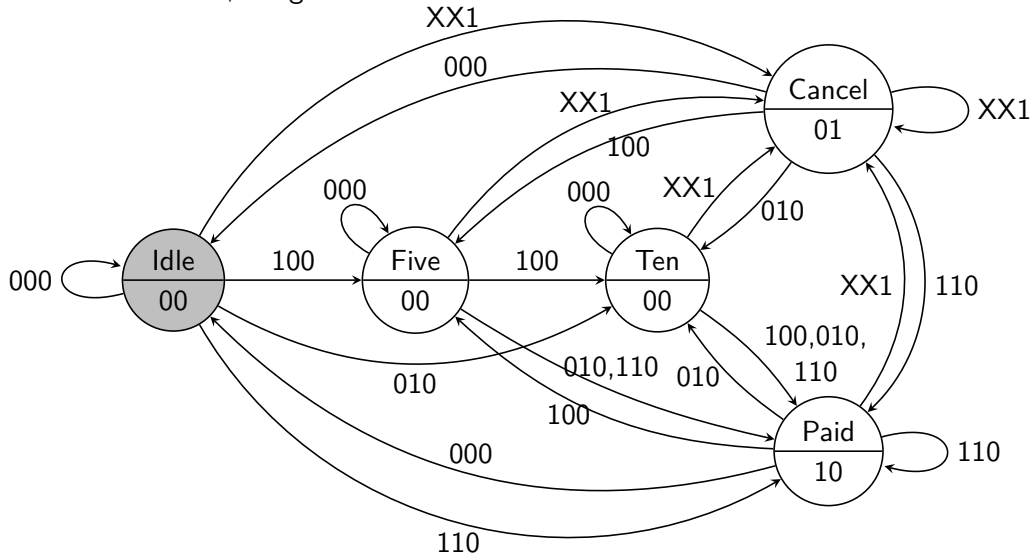
■ Ausgabe:

- 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: Zustandskodierung

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

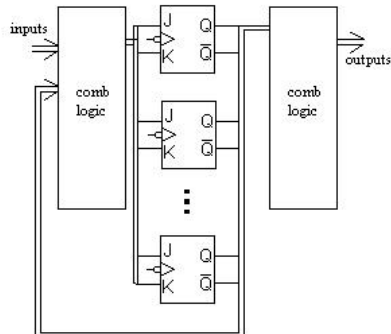
s_1	s_2	s_3	s_4	s_5	F	Z	R	s'_1	s'_2	s'_3	s'_4	s'_5
1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	1	0	0	0	0	1
1	0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0	1
1	0	0	0	0	1	1	0	0	0	0	1	0
1	0	0	0	0	1	1	1	0	0	0	0	1
0	1	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	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 $s'_1, s'_2, s'_3, s'_4, s'_5, ST, W$ 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.



- 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