Automatentheorie

endliche Maschinen

Prof. Dr. Franz-Karl Schmatzer
schmatzf@dhbw-loerrach.de

Literatur

- C.Wagenknecht, M.Hielscher; Formale Sprachen, abstrakte Automaten und Compiler; 2.Aufl. Springer Vieweg 2014;
- A.V.Aho, M.S.Lam,R.Savi,J.D.Ullman, *Compiler – Prinzipien,Techniken und Werkzeuge*. 2. Aufl., Pearson Studium, 2008.
- Güting, Erwin; *Übersetzerbau –Techniken, Werkzeuge, Anwendungen*, Springer Verlag 1999
- Sipser M.; Introduction to the Theory of Computation; 2.Aufl.; Thomson Course Technology 2006
- Hopecroft, T. et al; Introduction to Automata Theory, Language, and Computation; 3. Aufl. Pearson Verlag 2006

Agenda

- Moore-Maschine
- Mealy-Maschine
- Beispiele

Endliche Maschinen

- Endliche Maschinen sind Automaten, die um eine Ausgabefunktion erweitert werden.
- Eine Ausgabe kann dabei entweder durch einem Zustand (Moore) oder während einer Zustandsänderung erfolgen (Mealy).
- Daher unterscheidet man
  - Moore und Mealy Automaten
- Das Modell des endlichen Automaten muss nun um eine Ausgabefunktion erweitert werden.

Endliche Maschinen

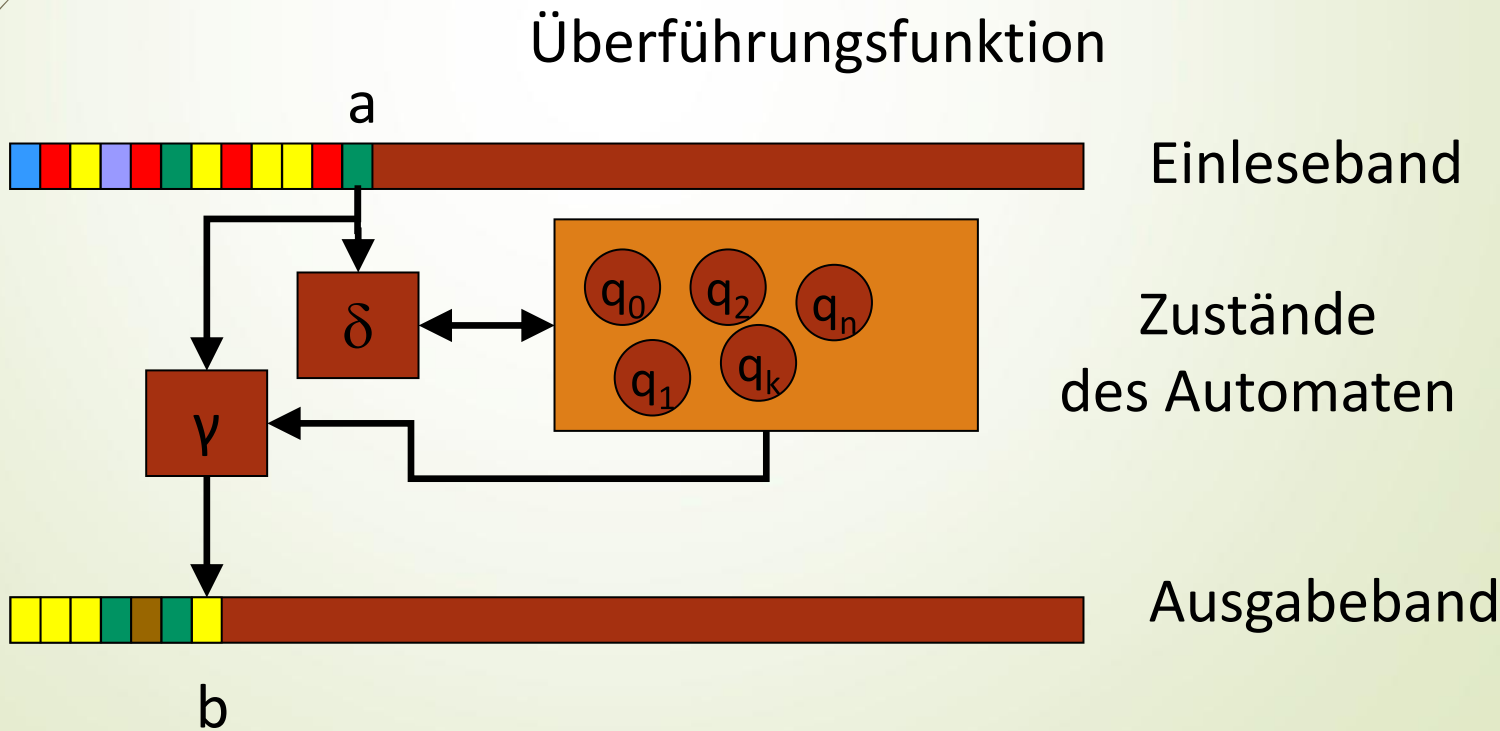
Einführung I

- Allgemeines Modell einer Maschine
  - Ein Einleseband mit Eingabezeichen,
  - Ein Ausgabeband mit Ausgabezeichen,
  - eine Maschine, die endliche viele interne Zustände haben kann und
  - eine Funktion, die abhängig von dem gelesenen Eingabezeichen und des momentanen Zustandes der Maschine, die Zustände der Maschine ändern kann.
  - Eine Ausgabefunktion
  - eine Startkonfiguration der Maschine
- Bem:
  - und Endkonfigurationen der Maschine existiert nicht !

Endliche Maschinen

Modell

- Beim Lesen des Zeichens a geht mittels δ die Maschine in einen neuen Zustand über und gibt über die Ausgabefunktion γ das Zeichen b aus.



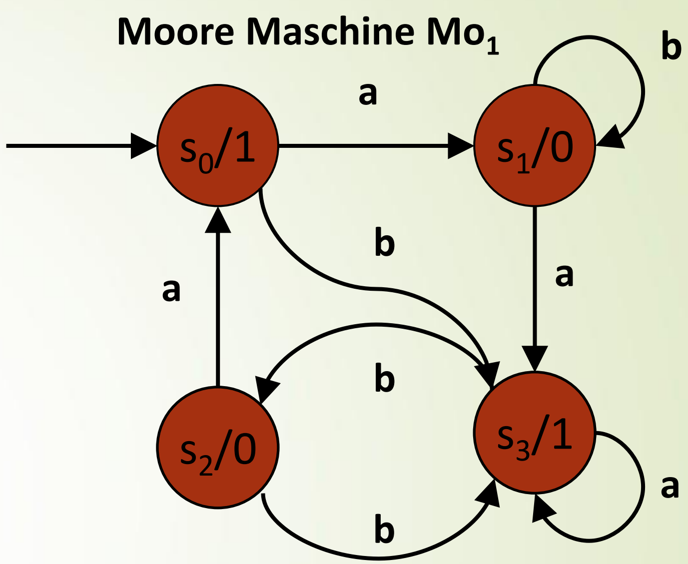
Endliche Maschinen

Einführung formal

- Sei $A = (Q, \Sigma, Z, \delta, \gamma, q_0)$ eine endliche Maschine.
  - $\Sigma = \{e_1,\ldots,e_n\}$ eine nicht leere Menge von Zeichen, das Eingabealphabet
  - $Q = \{q_0, \quad ,q_n\}$ eine nicht leere Menge von Zuständen
  - $Z = \{z_1,\ldots,z_n\}$ eine nicht leere Menge von Zeichen, das Ausgabealphabet
  - $\delta : Q \times \Sigma \rightarrow Q$ eine Funktion, die Überführungsfunktion
  - $\gamma : Q \times \Sigma \rightarrow Z$ eine Funktion, die Ausgabefunktion
  - $q_0 \in Q$ der Anfangszustand

Moore-Automat

- Ausgabe erfolgt an den Knoten
- Überführungsfunktion δ und Ausgabefunktion γ
- $Mo_1 = (\{s_0, s_1, s_2, s_3\}, \{a, b\}, \{0, 1\}, \delta, \gamma, s_0)$
- Wie arbeitet die Maschine?
- Einlesen des Wortes w = ababba



String		a	b	a	b	b	a
Zustand	s_0	s_1	s_1	s_3	s_2	s_3	s_3
Ausgabe	1	0	0	1	0	1	1

Zustände	δ		γ
	a	b	
s_0	s_1	s_3	1
s_1	s_3	s_1	0
s_2	s_0	s_3	0
s_3	s_3	s_2	1

Moore-Automat

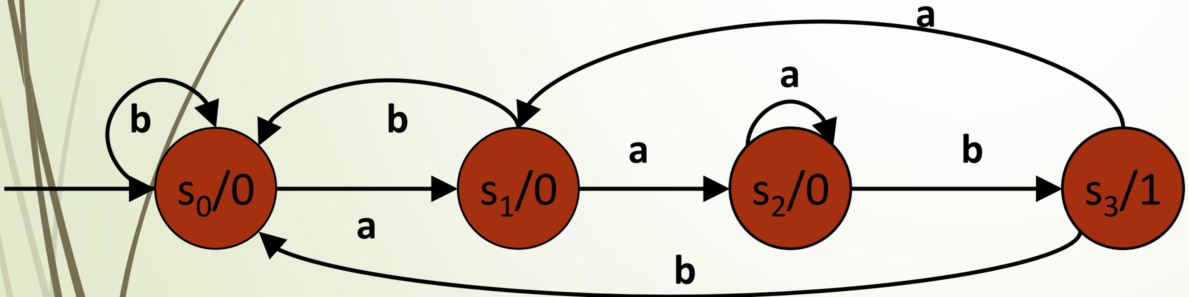
Beispiel Konstruktion

- Konstruieren Sie einen Moore Automaten, der jedes Mal eine 1 ausgibt, wenn der Zeichenstring z = aab in einem Wort erkannt wird. Sonst gibt der Automat eine 0 aus.
  - Geben Sie den Automatengraph und die Überführungsfunktion mit der Ausgabe an.
  - Lesen des Wortes w = aaababbaabb und geben Sie die Ausgabe beim Lesen an.

Moore-Automat

Beispiel Konstruktion

- Konstruieren Sie einen Moore Automaten, der jedes Mal eine 1 ausgibt, wenn der Zeichenstring z = aab in einem Wort erkannt wird. Sonst gibt der Automat eine 0 aus.
- Lösung:
  - Jedes Mal, wenn der substring z= aab erkannt wird, soll die Maschine eine 1 ausgeben, sonst eine 0.
  - Die Anzahl der 1er ist die gesuchte Lösung!



Moore Maschine Mo_1

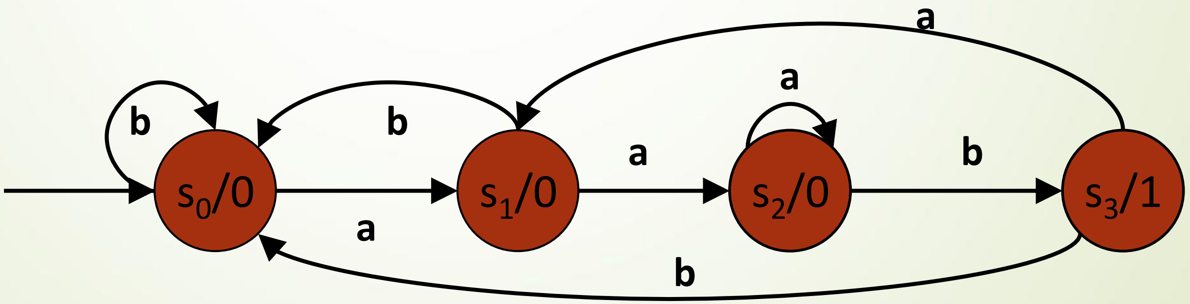
Zustände	δ		γ
	a	b	
s_0	s_1	s_0	0
s_1	s_2	s_0	0
s_2	s_2	s_3	0
s_3	s_1	s_0	1

Moore-Automat

Beispiel Arbeitsweise

- Lesen des Wortes w = aaababbaabb

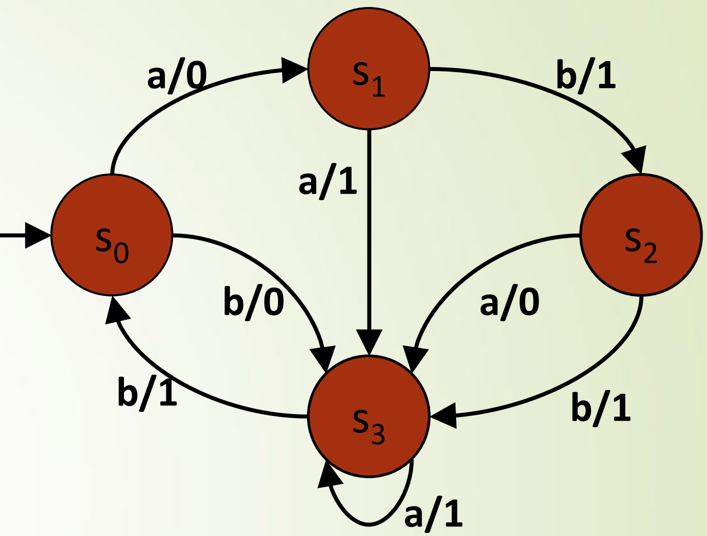
String		a	a	a	b	a	b	b	a	a	b	b
Zustand	s_0	s_1	s_2	s_2	s_3	s_1	s_0	s_0	s_1	s_2	s_3	s_0
Ausgabe	0	0	0	0	1	0	0	0	0	0	1	0



Moore Maschine Mo_1

Mealy-Automat

- Ausgabe erfolgt an den Übergängen (Kanten)
- Überführungsfunktion δ und Ausgabefunktion γ
- $Me_1 = (\{s_0, s_1, s_2, s_3\}, \{a, b\}, \{0, 1\}, \delta, \gamma, s_0)$
- Wie arbeitet die Maschine?
- Einlesen des Wortes w = ababba



Mealy Maschine Me_1

Zustände	δ/ γ	
	a	b
s_0	$s_1/0$	$s_3/0$
s_1	$s_3/1$	$s_2/1$
s_2	$s_3/0$	$s_3/1$
s_3	$s_3/1$	$s_0/1$

String		a	a	a	b	b	a
Zustand	s_0	s_1	s_3	s_3	s_0	s_3	s_3
Ausgabe		0	1	1	1	0	1

Mealy-Automat

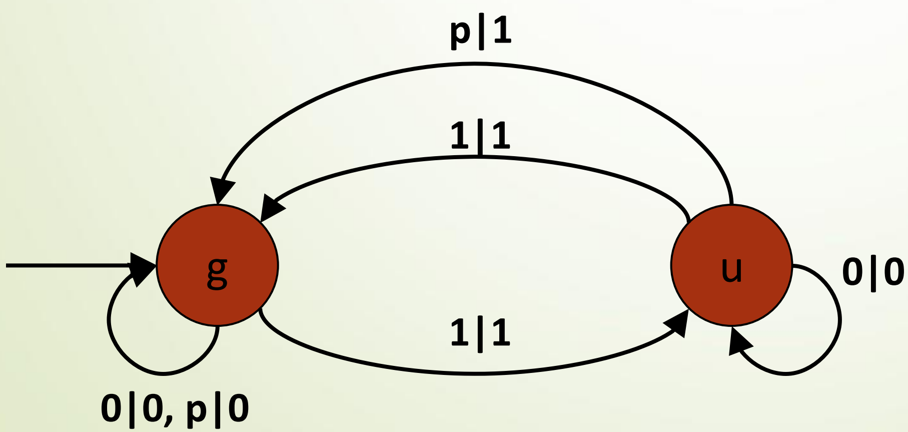
Beispiel Konstruktion

- Konstruieren Sie einen Mealy Automaten, der ein Paritätsbit an eine Zeichenkette anfügt, d.h.
  - für eine gerade Anzahl von 1 Bits wird ein 0 angefügt.
  - für eine ungerade Anzahl von 1 Bits wird eine 1 angefügt
- Was ist das Eingabe, was das Ausgabealphabet
- Geben Sie den Automatengraphen an.
- Geben Sie die Überführungsfunktion und die Ausgabe an.
- Zeigen Sie die Arbeitsweise für das Wort w = 01011011p

Mealy-Automat

Beispiel Konstruktion

- Konstruieren Sie einen Mealy Automaten, der ein Paritätsbit an eine Zeichenkette anfügt, d.h.
  - für eine gerade Anzahl von 1 Bits wird ein 0 angefügt.
  - für eine ungerade Anzahl von 1 Bits wird eine 1 angefügt
- Lösung:
  - Die Maschine hat 2 Zustände gerade und ungerade
  - Eingabe {0, 1, p}; Ausgabe {0,1}
  - $Me_1 = (\{0, 1, p\}, \{g,u\}, \{0, 1\}, \delta, \gamma, g)$



Mealy Maschine Me_2

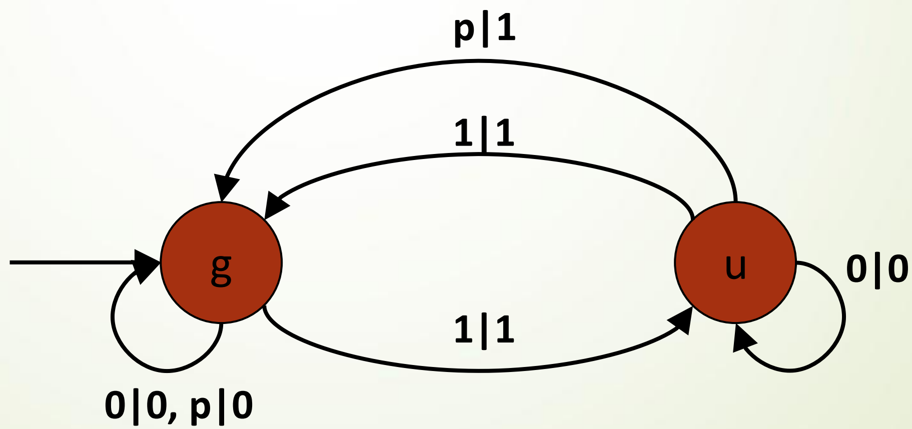
Zustände			
	0	1	p
9	g/0	u/1	g/0
u	u/0	g/1	g/1

Mealy-Automat

Beispiel Arbeitsweise

- Lesen des Wortes w = 01011011p

String	0	1	0	1	1	0	1	1	p
Zustand	g	u	u	g	u	u	g	u	u
Ausgabe	0	1	0	1	1	0	1	1	1



Mealy Maschine Me_2

Aufgaben endliche Maschinen

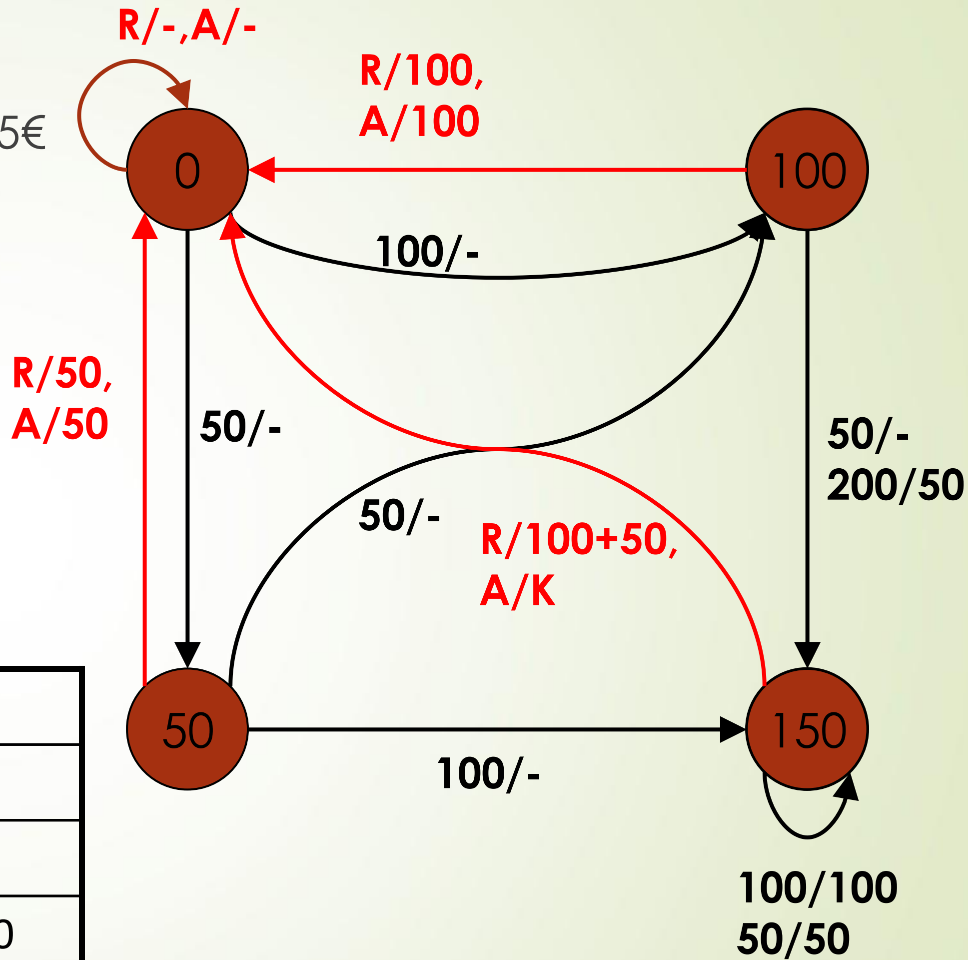
1. Konstruieren Sie einen Kaffee-Automaten.
   1. Er soll nur 1€ und 50 Cent akzeptieren.
   2. Der Kaffeepreis beträgt 1,50€
2. Was ist das Eingabe-, was das Ausgabealphabet?
3. Welche Zustände hat der Automat?
4. Geben Sie den Graphen und die Übertragungsfunktion Γ an.

Kaffee-Automat 1

A) Münzmenge M{0,5€, 1€}. Ein Kaffeepreis 1,5€

- Eingabemenge (50,100,R, A)
- Ausgabemenge(-,K,50,100)
- Zustände (0,50,100,150)

- Übertragungsfunktion Γ



	50E	100E	R	A
0	50/-	100/-	0/-	0/-
50	100/-	150/-	0/50	0/50
100	150/-	150/50	0/100	0/100
150	150/50	150/100	0/(100+50)	0/K

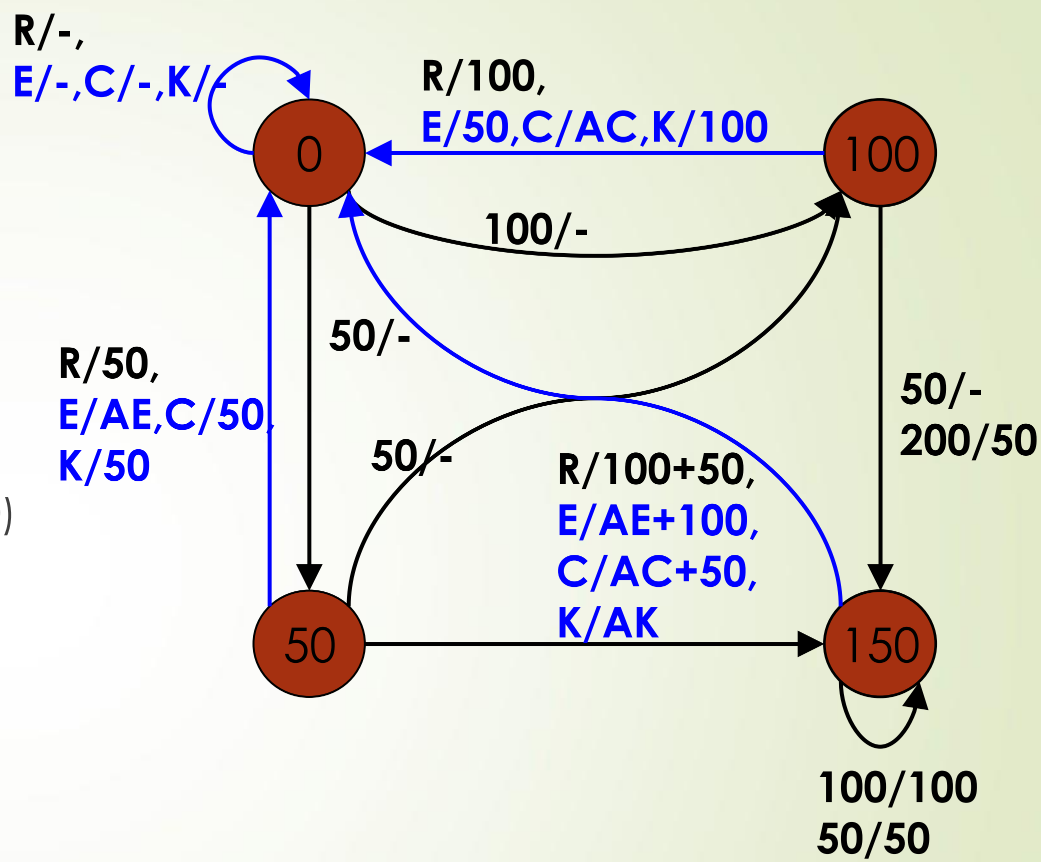
Aufgaben endliche Maschinen

1. Konstruieren Sie einen Kaffee-Automaten.
   1. Er soll nur 1€ und 50 Cent akzeptieren.

   Es gibt 3 Kaffeevarianten
      - 0,5€ Espresso
      - 1,0€ Cappuccino
      - 1,5€ Kaffee
   - Es gibt eine Abbruchtaste.
2. Was ist das Eingabe-, was das Ausgabealphabet?
3. Welche Zustände hat der Automat?
4. Geben Sie den Graphen und die Übertragungsfunktion Γ an.

Kaffee-Automat 2

A) Münzmenge M{0,5€, 1€}.
   - Kaffeepreise
     - 0,5€ Espresso
     - 1,0€ Cappuccino
     - 1,5€ Kaffee
   - Eingabemenge (50,100,R, E,C,K)
   - Ausgabemenge(-,AE,AC,AK,50,100)
   - Zustände (0,50,100,150)
- Übertragungsfunktion Γ



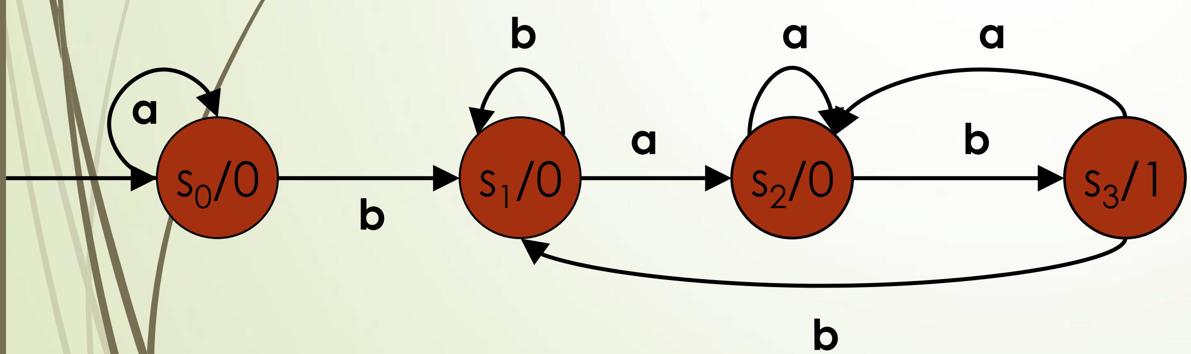
	50E	100E	R	E	C	K
0	50/-	100/-	0/-	0/-	0/-	0/-
50	100/-	150/-	0/50	0/E	0/50	0/50
100	150/-	150/50	0/100	0/E+50	0/C	0/100
150	150/50	150/100	0/(100+50)	0/E+100	0/C+50	0/K

Aufgaben endliche Maschinen

1. Konstruieren Sie einen Moore Automaten, der jedes Mal eine 1 ausgibt, wenn der Zeichenstring z = bab in einem Wort erkannt wird. Sonst gibt der Automat eine 0 aus. Geben Sie den Automatengraph und die Überführungsfunktion mit der Ausgabe an.
2. Konstruieren Sie eine Mealy-Maschine, die aus einer binären Zahl d, die zugehörige negative Zahl im 2er-Komplement erstellt.

Lösung Aufgabe 2

- Lösungsidee:
  - Jedes Mal, wenn der substring z= bab erkannt wird, soll die Maschine eine 1 ausgeben, sonst eine 0.
  - Die Anzahl der 1er ist die gesuchte Lösung!
- Es muss der String bab erkannt werden, d.h man benötigt eine Maschine mit mindestens 4 Zuständen.



Zustände	δ		γ
	a	b	
s_0	s_0	s_1	0
s_1	s_2	s_1	0
s_2	s_2	s_3	0
s_3	s_2	s_1	1

Lösung Aufgabe 3

- Konstruieren Sie eine Mealy-Maschine, die aus einer binären Zahl d, die zugehörige negative Zahl im 2er-Komplement erstellt.
  - Sei $d := d_n d_{n-1} \ldots d_1 d_0$ eine Zeichen Kette mit $d_i \in \{0,1\}$
  - Das 2er-Komplement ist: $d^C := d_n^C d_{n-1}^C \ldots d_1^C d_0^C + 1$ mit $d_i^C := XOR(1, d_i)$
  - Beispiel: $d=100101 \Rightarrow d^C=011010+1=011011$

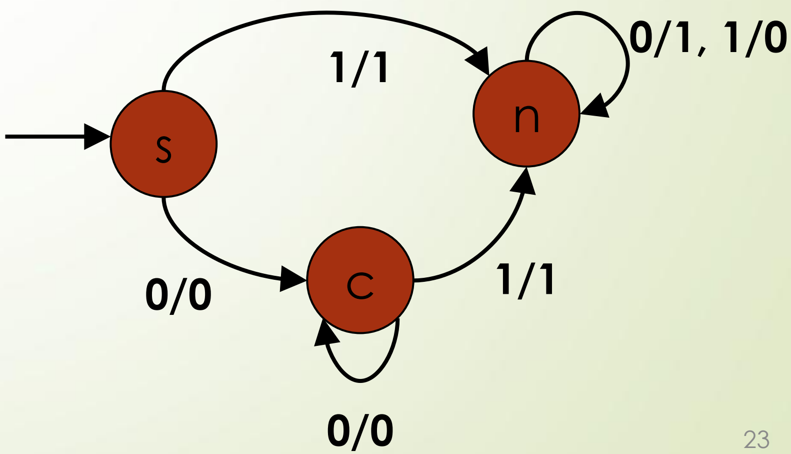
d	$\mathbf{d^C}$
1011100	0100100
0100110	1011010
1100011	0011101

Lösung Aufgabe 3

- Beobachtung 0 und 1 werden zuerst in 1 und 0 gewandelt und dann eine 1 addiert.
- Falls die letzte Ziffer eine 0 war, wird bei der Addition daraus eine 1 und es erfolgt keine weiteres Propagieren der 1 (no carry) in die nachfolgenden Ziffern.
- Falls die letzte Ziffer eine 1 war, wird bei der Addition daraus eine 0 und die 1 wird in die nächste Ziffer weiter propagiert (carry).

Wir starten mit 3 Zustände (Start s , carry c und no carry n)

- Geht es auch mit 2 Zuständen?



Lösung Aufgaben 1

- Konstruieren Sie einen Mealy-Automaten oder Moore-Automaten über das Eingabealphabet {0,1} mit folgenden Eigenschaften.
  - Die Eingabezeichen werden mit einer NOT-Funktion verknüpft und ausgegeben.

    011011011.. $\Rightarrow$ 100100100..
  - Je zwei Folgezeichen werden über ein AND verknüpft und ausgegeben.

    011011011.. $\Rightarrow$ 011011011.. AND 11011011.. = 01001001..

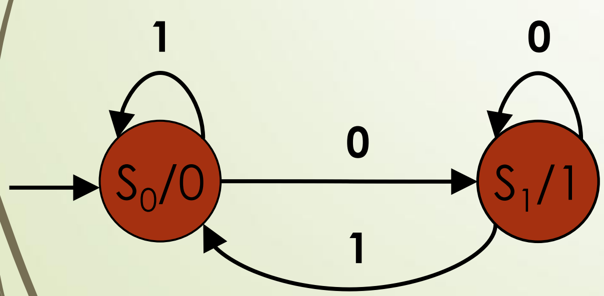
Lösung Aufgabe 4 (NOT- Funktion)

- Moore-Maschine

$Mo = (\{0, 1\}, \{s_0, s_1\}, \{0, 1\}, \delta, \gamma, s_0)$

Die erste Ausgabe (0) der Maschine wird verworfen.

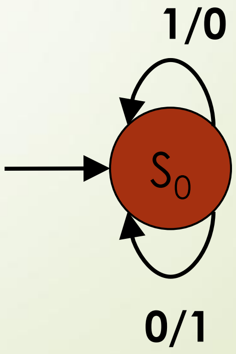
Zustände	δ		γ
	0	1	
s_0	s_1	s_0	0
s_1	s_1	s_0	1



- Mealy-Maschine

$Me = (\{0, 1\}, \{s_0\}, \{0, 1\}, \delta, \gamma, s_0)$

Zustände	δ/ γ	
	0	1
s_0	$s_0/1$	$s_0/0$



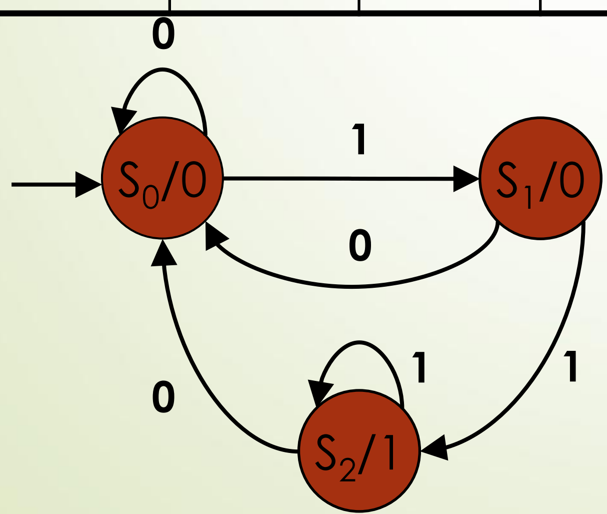
Lösung Aufgabe 5 (AND- Funktion)

- Moore-Maschine

$Mo = (\{0, 1\}, \{s_0, s_1, s_2\}, \{0, 1\}, \delta, \gamma, s_0)$

Die erste beiden Ausgaben der Maschine (00) werden verworfen.

Zustände	δ		γ
	0	1	
s_0	s_1	s_1	0
s_1	s_0	s_2	0
s_2	s_0	s_2	1

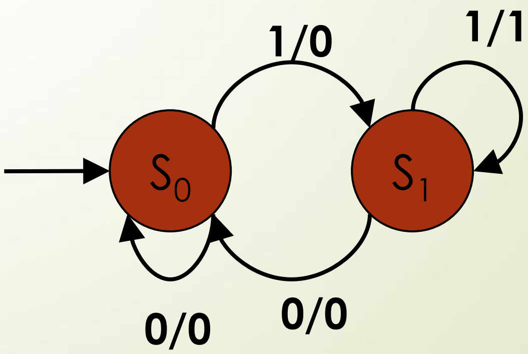


- Mealy-Maschine

$Me = (\{0, 1\}, \{s_0, s_1\}, \{0, 1\}, \delta, \gamma, s_0)$

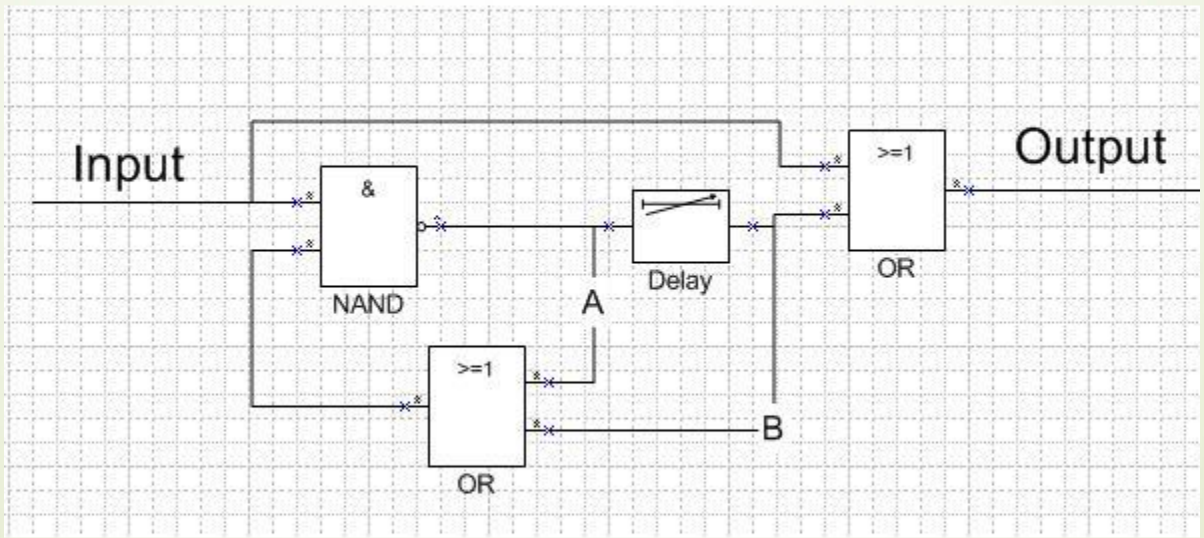
Die erste Ausgabe (0) wird verworfen

Zustände	δ/ γ	
	0	1
s_0	$s_0/0$	$s_1/0$
s_1	$s_0/0$	$s_1/1$



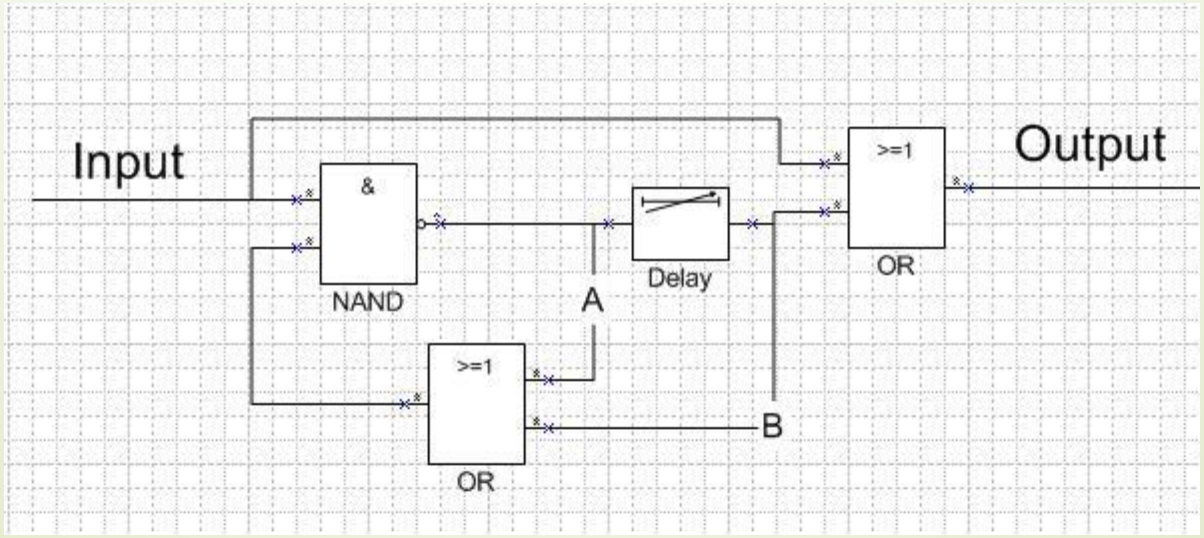
Aufgabe logische Schaltung

- Konstruieren Sie eine Mealy-Maschine zu folgender logischen Schaltung
- In der Komponente „Delay“ wird das Eingabesignal um 1 Takt verzögert.



Lösung Aufgabe (1. Schritt)

- Zur Lösung wichtig sind die Signale an den Punkten A und B
- Dazu schauen wir uns an, wie die Maschine arbeitet
  - new B = old A
  - new A = (input) NAND (old A OR old B)
  - output = (input) OR (old B)
- Um aus einem Input einen Output zu generieren, müssen wir uns die Zustände in A und B merken, dh. Insgesamt 4 Zustände.



Lösung Aufgabe (2. Schritt)

- Aufstellen der Übergangsfunktion

  new B = old A

  new A = NAND(input, OR(old A, old B))

  output = OR(input, old B)
- Setze s_0 als den Zustand mit A=B=0, den Zustand s_1 mit A=0 und B=1, usw.

Input = 0 und old A = old B = 0 $\Rightarrow$

- new A = NAND(0,OR(0,0))=1 und
- new B = old A = 0; d,h. Zustand s_2
- output = OR(0,0) = 0;

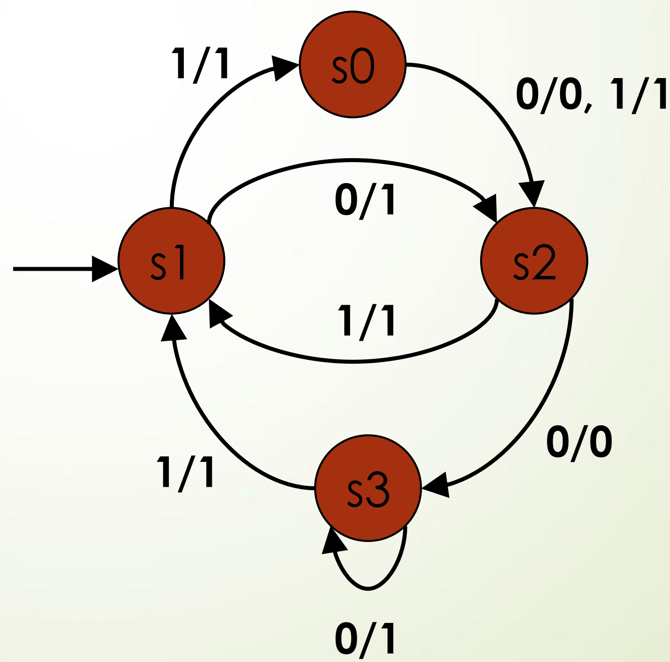
Input = 1 und old A = old B = 0 $\Rightarrow$

- new A = NAND(1,OR(0,0))=1 und
- new B = old A = 0; d.h. Zustand s_2
- output = OR(1,0) = 1;

Zustände A,B			Input 0		Input 1	
A	**B**	**old**	**Zustand**	**output**	**Zustand**	**output**
0	0	s_0	s_2	0	s_2	1
0	1	s_1	s_2	1	s_0	1
1	0	s_2	s_3	0	s_1	1
1	1	s_3	s_3	1	s_1	1

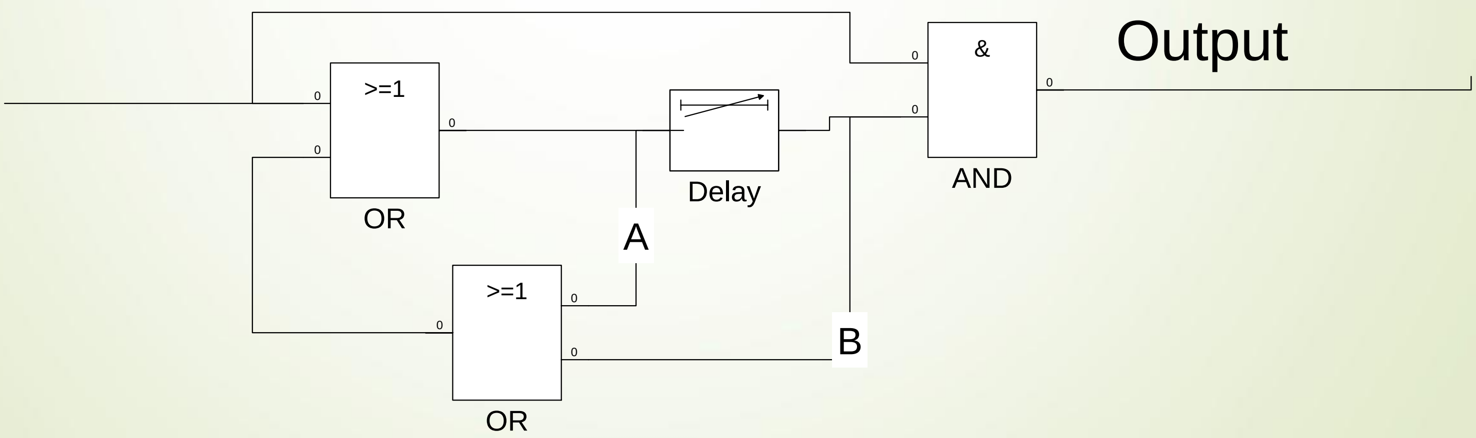
Lösung Aufgabe (3. Schritt) Mealy-Maschine

- Start mit 00 und befindet sich in s_3 unabhängig wo die Maschine zuvor war.
- Die Eingabe 011011 produziert dann 111011



Aufgabe logische Schaltung 2

- Konstruieren Sie eine Mealy-Maschine zu folgender logischen Schaltung
- In der Komponente „Delay“ wird das Eingabesignal um 1 Takt verzögert.



Lösung Aufgabe 2 (1. Schritt)

- Zur Lösung wichtig sind die Signale an den Punkten A und B
- Dazu schauen wir uns an, wie die Maschine arbeitet
  - new B = old A
  - new A = (input) OR (old A OR old B)
  - output = (input) AND (old B)
- Um aus einem Input einen Output zu generieren, müssen wir uns die Zustände in A und B merken, d.h. Insgesamt 4 Zustände.

Lösung Aufgabe 2 (2. Schritt)

- Aufstellen der Übergangsfunktion

  new B = old A

  new A = (input) OR (old A OR old B)

  output = (input) AND (old B)

- Setze s_0 als den Zustand mit A=B=0,

  den Zustand s_1 mit A=0 und B=1, usw.

Input = 0 und old A = old B = 0 $\Rightarrow$

- new A = OR(0,OR(0,0))=0 und
- new B = old A = 0; d.h. Zustand s_0
- output = AND(0,0) = 0;

Input = 1 und old A = old B = 0 $\Rightarrow$

- new A = OR(1,OR(0,0))=1 und
- new B = old A = 0; d.h. Zustand s_2
- output = AND(1,0) = 0;

Zustände A,B			Input 0		Input 1	
A	**B**	**old**	**Zustand**	**output**	**Zustand**	**output**
0	0	s_0	s_0	0	s_2	0
0	1	s_1	s_2	0	s_2	1
1	0	s_2	s_3	0	s_3	1
1	1	s_3	s_3	0	s_3	1