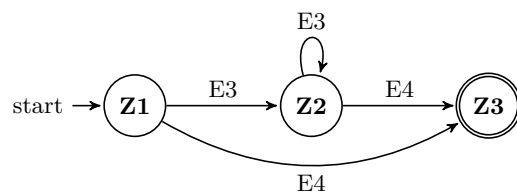


## 1 Verkehrsregeln

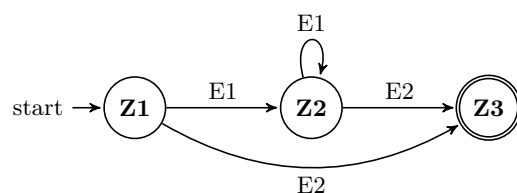
Zustände		Ereignisse	
Heranfahren	Z1	Fahrzeug bei B	E1
Warten	Z2	B ist frei	E2
Weiterfahren	Z3	Fahrzeug bei C	E3
		C ist frei	E4

(a) 'kommt von A, will nach B'



Startzustand	Ereignis	Folgezustand
Z1	E3	Z2
Z1	E4	Z3
Z2	E3	Z2
Z2	E4	Z3
Z3	-	-

(b.1) 'kommt von C, will nach A'



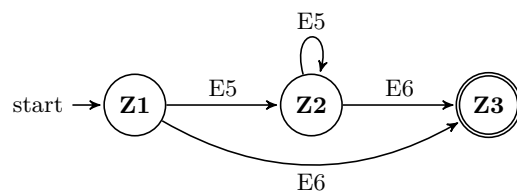
Startzustand	Ereignis	Folgezustand
Z1	E1	Z2
Z1	E2	Z3
Z2	E1	Z2
Z2	E2	Z3
Z3	-	-

(b.2) 'kommt von B, will nach C'

Wir definieren zuerst zwei neue Ereignisse:

Ereignisse	
Fahrzeug bei A	E5
A ist frei	E6

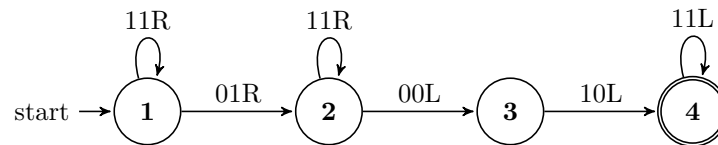
Der Automat:



Startzustand	Ereignis	Folgezustand
Z1	E5	Z2
Z1	E6	Z3
Z2	E5	Z2
Z2	E6	Z3
Z3	-	-

## 2 Turingmaschine

(a)



Eingabe → ↓ Zustand	1	0	*
1	1R1	1R2	-
2	1R2	0L3	-
3	0L4	-	-
4	1L4	-	-

'\*' steht für undefinierte Zeichen. Tripel geben Ausgabezeichen, Kopfbewegung und neuen Zustand an. Leere Zellen stellen nicht akzeptierte Eingabe-Zustand-Kombinationen dar, die z.B. zum Programmabbruch führen.

(b)

- i Das TM-Programm soll überprüfen, ob zwischen zwei Leerzeichen (#) auf dem Band ein Ausdruck mit gültiger Klammerung steht. D.h. genauso viele schließende ')' wie öffnende '(' Klammer und '(' vor ')', oder gar keine Klammer. Wenn der Ausdruck wohlgeformt ist, wird nach der Überprüfung eine 1 vorne geschrieben. Wenn nicht (mehr schließende als öffnende Klammer (oder umgekehrt) bzw. ')' vor '(', z.B. ')X(X') wird eine 0 geschrieben.

Die Zustände könnten z.B. folgende Bedeutungen haben:

Zustand	Bedeutung
1	Suche ')'
2	Suche '('
3	Überprüfung fertig
4	Entscheidung getroffen

- ii Wie oben, '\*' steht für undefinierte Zeichen. Tripel geben Ausgabezeichen, Kopfbewegung und neuen Zustand an. Leere Zellen stellen nicht akzeptierte Eingabe-Zustand-Kombinationen dar, die z.B. zum Programmabbruch führen.

Eingabe → ↓ Zustand	X	(	)	#	*
1	XR1	(R1	XL2	#L3	-
2	XL2	XR1	)L2	0#4	-
3	XL3	0#4	-	1#4	-
4	-	-	-	-	-

iii Beispielgestalt: #X(XX)#

Pos.	Zust.	Eing.	Ausg.	BR	Folgez.	Gestalt
# <u>X</u> (XX)#	1	X	X	R	1	#X(XX)#
#X( <u>XX</u> )#	1	(	(	R	1	#X(XX)#
#X(X <u>X</u> )#	1	X	X	R	1	#X(XX)#
#X(XX <u>)</u> #	1	X	X	R	1	#X(XX)#
#X(XX) <u>#</u>	1	)	X	L	2	#X(XXX#
#X( <u>XXX</u> )#	2	X	X	L	2	#X(XXX#
#X(X <u>XX</u> )#	2	X	X	L	2	#X(XXX#
#X(XXX <u>)</u> #	2	(	X	R	1	#XXXXX#
# <u>XXXXX</u> #	1	X	X	R	1	#XXXXX#
#XXXX <u>X</u> #	1	X	X	R	1	#XXXXX#
#XXXXX <u>#</u>	1	X	X	R	1	#XXXXX#
#XXXXX <u>#</u>	1	#	#	L	3	#XXXXX#
#XXXXX <u>#</u>	3	X	X	L	3	#XXXXX#
#XXXXX <u>#</u>	3	X	X	L	3	#XXXXX#
#XXXXX <u>#</u>	3	X	X	L	3	#XXXXX#
#XXXXX <u>#</u>	3	X	X	L	3	#XXXXX#
#XXXXX <u>#</u>	3	X	X	L	3	#XXXXX#
#XXXXX <u>#</u>	3	X	X	L	3	#XXXXX#
#XXXXX <u>#</u>	3	#	1	#	4	1XXXXX#
<u>1XXXXX</u> #	4	-	-	-	-	1XXXXX#

### 3 Uhren

Definieren wir folgende Variablen:

- normale Stunde (60 Min)  $\rightarrow t$
- Wanduhr-Stunde  $\rightarrow t_w$
- Wecker-Stunde  $\rightarrow t_r$
- Küchenuhr-Stunde  $\rightarrow t_k$
- Taschenuhr-Stunde  $\rightarrow t_t$

Der Wert des Ausdrucks  $t - t_t$  sagt uns, ob die Taschenuhr gegenüber der wahren Zeit vor, nach oder genau ging:

$$Taschenuhr : \begin{cases} \text{geht nach,} & \text{wenn } t - t_t > 0 \\ \text{geht vor,} & \text{wenn } t - t_t < 0 \\ \text{geht genau,} & \text{wenn } t - t_t = 0 \end{cases}$$

Wie zählen ab 12:00 Uhr. Dann:

- 'Als die Küchenuhr schließlich 14 Uhr erreichte, war es auf der Taschenuhr bereits 14:04'

$$2t_t = 2t_k + 4$$

- 'Als es auf dem Wecker 14 Uhr war, zeigte die Küchenuhr erst 13:56.'

$$\begin{aligned} 2t_k &= 2t_r - 4 \\ \Rightarrow 2t_t &= 2t_r - 4 + 4 = 2t_r \\ \Rightarrow t_t &= t_r \end{aligned}$$

- 'Als die Wanduhr 13 Uhr läutete, zeigte der Wecker schon 13:02.'

$$\begin{aligned} t_r &= t_w + 2 \\ \Rightarrow t_t &= t_w + 2 \end{aligned}$$

- 'Beim Zeitzeichen der 13-Uhr-Nachrichten ging die Wanduhr bereits 2 Minuten nach (12:58)'

$$\begin{aligned} t_w &= t - 2 \\ \Rightarrow t_t &= t - 2 + 2 \\ \Rightarrow t - t_t &= 0 \end{aligned}$$

$\Rightarrow$  Die Taschenuhr ging gegenüber der wahren Zeit **genau**.