11.3 Eindimensionale Turingmaschinen

Turing ging vom schriftlichen Rechnen aus, also vom Beschreiben eines Papiers mit einem Stift. Wollen wir etwas aufschreiben, dann benötigen wir dafür ein Medium, das sich beschreiben und wieder lesen lässt. Da die Form dieses Mediums gleichgültig ist, benutzen wir das schon bekannte Band eines Automaten. Der Einfachheit halber kann auch die Anfangsbeschriftung der Maschine auf diesem Band untergebracht werden. Die schon weiter oben skizzierte Turingmaschine benötigt also nur ein einzelnes *Arbeitsband*. Auch die benötigten Zeichen lassen sich beliebig codieren, z. B. als Dualzahlen. Deshalb wollen wir für die Darstellung der Zeichen auf dem Band Dualziffern wählen.

Betrachten wir den Rechenvorgang selbst. Bei uns beschränkt sich dieser auf die Operationen des wiederholten Einsetzens von Funktionswerten in andere und die Ermittlung der Werte der elementaren Funktionen. Da die Beschriftung selbst aus Einsen und Nullen gebildet wird, erfordert Rechnen die Möglichkeit, das Rechenband zu bewegen, zu lesen und wieder zu beschreiben. In Abhängigkeit vom letzten und eventuell den vorher gelesenen Zeichen werden dann weitere solcher Operationen veranlasst. Diese Möglichkeit, unterschiedliche Aktionen auszulösen, kennzeichnet gerade die endlichen Automaten. Reicht deren „Gedächtnis" in Form von Zuständen nicht aus, dann steht das Arbeitsband ebenso wie bei den Kellerautomaten als „Notizbuch" zur Verfügung.

Eine Turingmaschine wird aus einem endlichen Automaten gebildet, der um die Fähigkeit zur Manipulation des Arbeitsbandes erweitert wurde, auf dem anfangs nur die Eingabedaten stehen.

Da für die Arbeit bestimmt sein muss, an welcher Stelle des Arbeitsbandes begonnen wird, definieren wir eine *Standardbeschriftung* des Bandes und eine *Standardlage* des Schreib-/Lese-(SL)-Kopfes des Automaten.

Das n*-te Zeichen wird durch* n+1 *Einsen verschlüsselt (unäre Darstellung).*

Damit ist es möglich, auch das Zeichen 0 darzustellen: durch eine Eins. Mehrere Zeichen werden durch genau eine Null getrennt. Die Anfangsbeschriftung beginnt mit zwei Nullen, gefolgt von den Eingabezeichen, und wird mit wenigstens zwei Nullen beendet. Das Arbeitsband ist nach rechts hin unendlich lang.

In der Standardlage befindet sich der SL-Kopf der Maschine über der am weitesten rechts stehenden Eins des Bandes.

Damit könnte eine mögliche Ausgangssituation auf dem Arbeitsband wie folgt aussehen. (Der SL-Kopf der Turingmaschine wird durch den Pfeil dargestellt.)

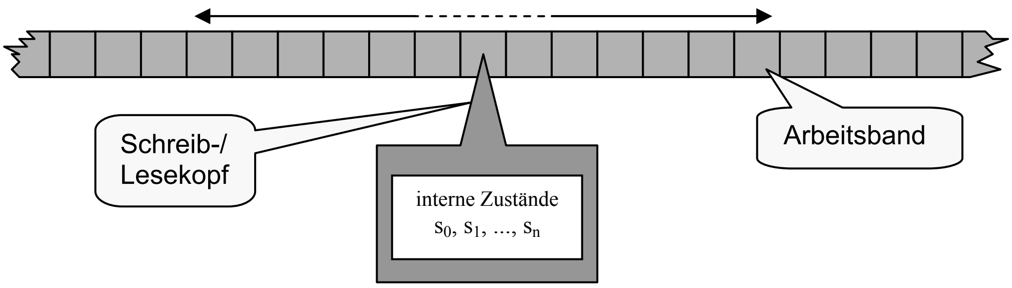
0	0	**1**	**1**	**1**	0	**1**	**1**	0	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0
														↑									

Eine Turingmaschine T mit den beschriebenen Eigenschaften wird beschrieben wie ein erkennender Automat mit einer zusätzlichen Menge B von Bandbewegungen. (Dabei entspricht natürlich eine Kopfbewegung nach links einer Bandbewegung nach rechts usw.)

$T = (E, S, B, s_0, u, \Sigma)$ mit

$E = \{ e_1, e_2,\ldots\ldots\ldots,e_r \}$	dem Alphabet für Ein- und Ausgabe
$B = \{ L,R,H \}$	dem Bandbewegungsalphabet L: Lesekopf um ein Feld nach links bewegen R: Lesekopf um ein Feld nach rechts bewegen H: Lesekopf nicht bewegen
$S = \{ s_0, s_1,\ldots\ldots\ldots,s_n \}$	der Zustandsmenge
$s_0 \,\varepsilon\, S$	dem Anfangszustand
$\Sigma \subset S$	der Menge der Endzustände
$u: (e_i,s_j) \longrightarrow (e_k,s_l,b_m)$	die Überführungsfunktion ($1 \le i \le r$, $0 \le j \le n$, $1 \le l \le n$, $b_m \varepsilon B$)

Als Modell ergibt sich eine ähnliche Anordnung wie bei den endlichen Automaten. Das Eingabeband kann jetzt aber in beiden Richtungen bewegt werden.



Die Arbeitsweise der Turingmaschine wird in Form eines Struktogramms beschrieben:

T in die Standardlage bringen	
$s \leftarrow s_0$	
wdh	lies(e)
	Zeichen unter dem SL-Kopf gemäß u(s,e) neu beschreiben
	Kopfbewegung gemäß u(s,e) durchführen
	Folgezustand einnehmen
bis ein Endzustand erreicht wird	

Wir wollen jetzt konkrete Turingmaschinen entwickeln: zuerst die **Kopiermaschine K_1**. Sie soll eine Gruppe aus Einsen kopieren, also rechts neben dieselbe Gruppe schreiben, getrennt von der ersten durch eine Null. Der SL-Kopf wird durch einen Pfeil markiert.

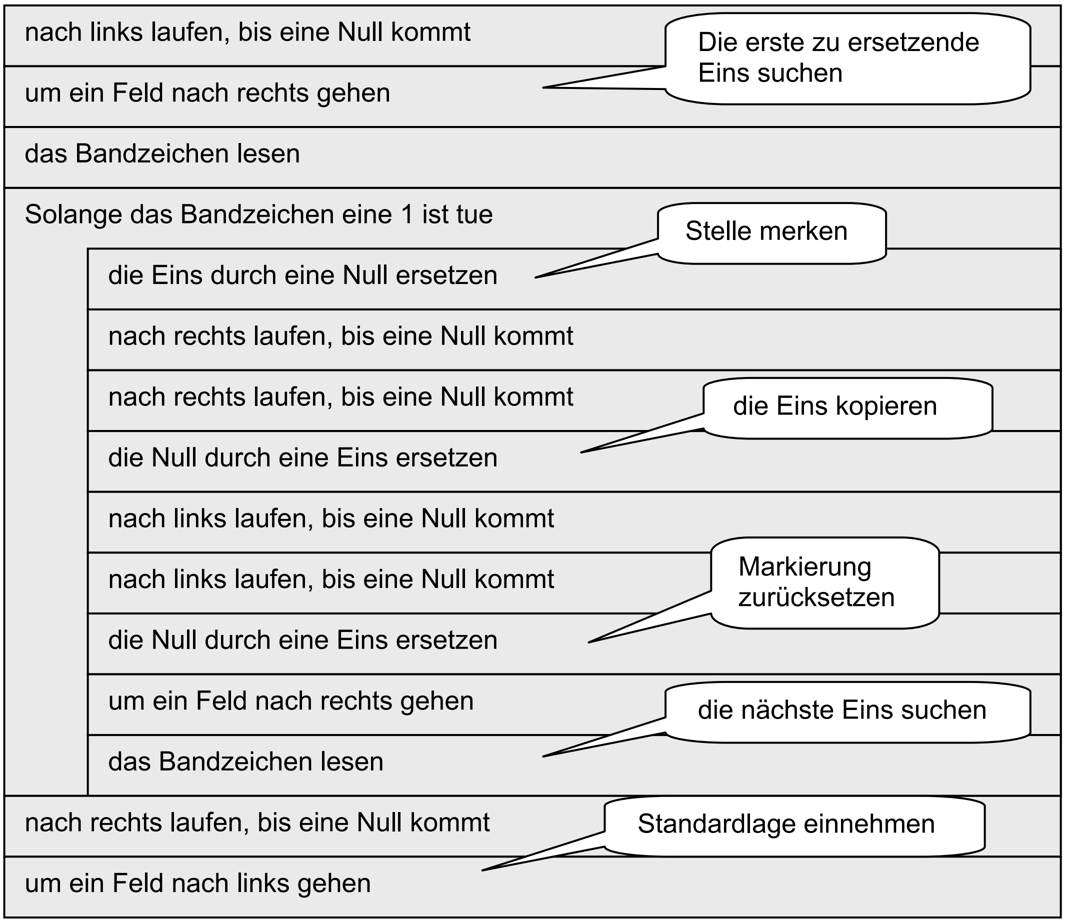
Ausgangslage:

0	0	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
							↑													

Endlage:

0	0	**1**	**1**	**1**	**1**	**1**	**1**	0	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0
														↑						

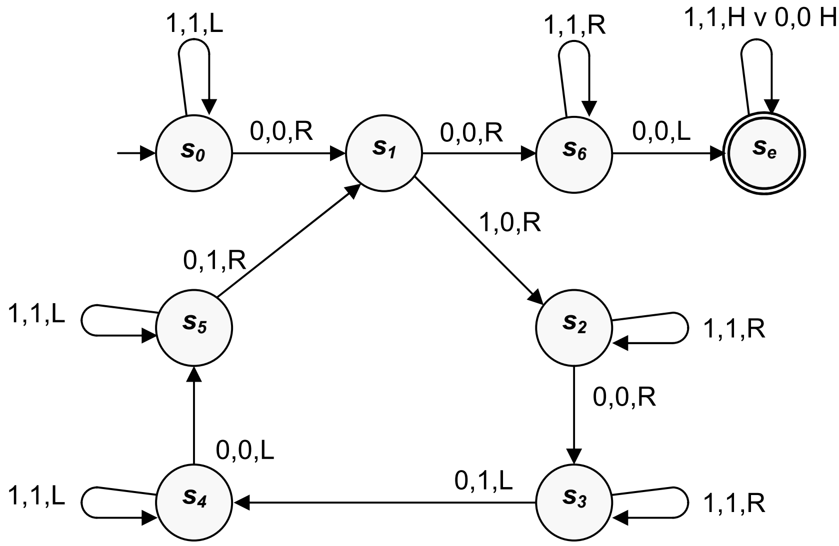
Da die Einsengruppe beliebig lang sein kann, muss sich K_1 irgendwie merken, welche Eins sie schon kopiert hat. Dazu ersetzt K_1 die gerade zu kopierende Eins durch eine Null, kopiert diese und macht danach die Ersetzung rückgängig.



Mit einem Transitionsgraphen können wir dieses Verhalten wie folgt beschreiben. Dabei werden die Kanten durch Zeichenfolgen

<gelesenes Zeichen><geschriebenes Zeichen><Kopfbewegung>

beschriftet. **1,0,L** z. B. bedeutet, dass beim Lesen einer Eins eine Null zurück geschrieben wird und der Kopf um ein Feld nach links wandert.



Natürlich kann auch unser Simulator SIAM die Arbeit der Kopiermaschine visualisieren.

Als weiteres Beispiel folgt die **Addiermaschine A**. Die Maschine soll zwei Zahlen a und b, die durch eine Null getrennt auf dem Band stehen, addieren. Da a durch a+1 Einsen und b durch b+1 Einsen dargestellt werden, ergibt eine Addition der Zeichen a+b+2 Einsen, also eine zuviel. A muss also die Null zwischen den Zahlen durch eine Eins ersetzen (damit stehen a+b+3 Einsen auf dem Band), nach rechts laufen und dort zwei Einsen löschen.

Ausgangslage:	0	0	**1**	**1**	**1**	**1**	0	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0
Endlage:	0	0	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0