Theoretische Informatik

Turingmaschine (TM)

E. Mehofer

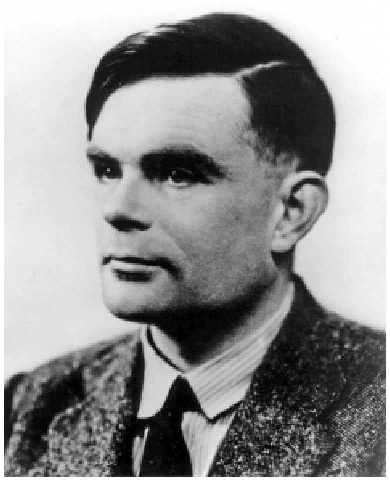
Scientific Computing

Fakultät für Informatik

Universität Wien

Gliederung

- Einführung.
- Komponenten einer TM.
- Arbeitsweise einer TM.
- Formale Definition einer TM.
  - Struktur.
  - Konfiguration.
  - Spezifikation:
    - Turingtafel (Tabelle)
    - Übergangsdiagramm.
- Akzeptierte Sprachen.
- Berechnungen mit einer TM.



Alan Turing

(1912-1954)

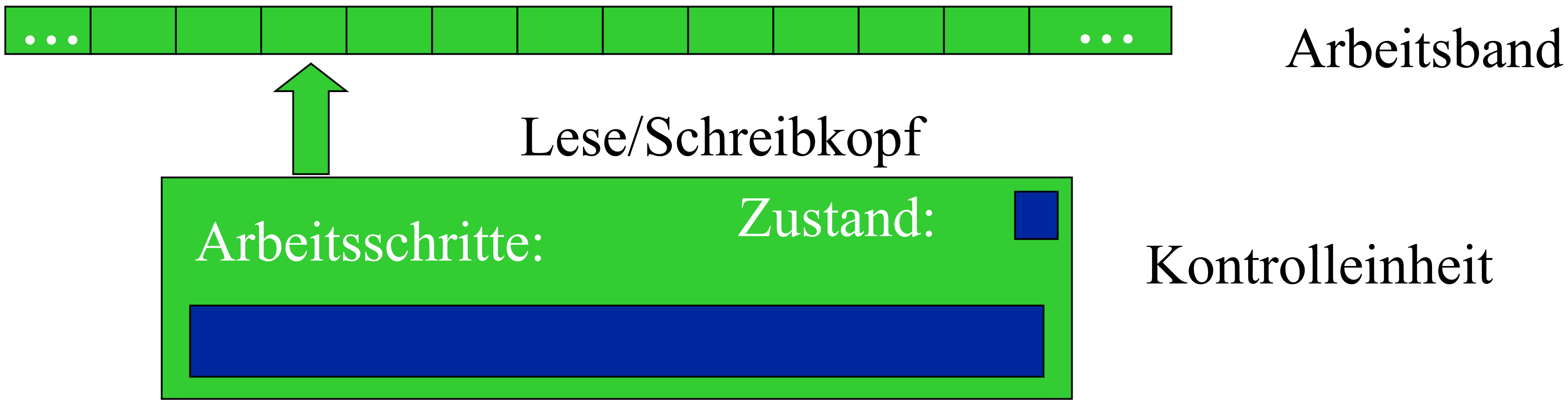
Turing Award

Einführung

- Eine **Turingmaschine** (TM) ist ein einfaches mathematisches Modell, um Berechnungen durchzuführen. Wurde 1936 von Alan Turing publiziert.
- Turingmaschinen haben eine große Bedeutung für Überlegungen zu **Berechenbarkeit** und **Entscheidbarkeit**.
- Mann kann zeigen, dass eine TM jede Funktion berechnen kann, die in einer beliebigen Programmiersprache oder in einem anderen Berechenbarkeitsmodell (z.B. while-Programme, Lambda-Kalkül, etc.) formuliert werden kann.
- Es gibt viele leicht unterschiedliche Versionen bzw. Erweiterungen der klassischen TM. Die Grundlage für Prüfungen ist jene, die hier eingeführt wird.

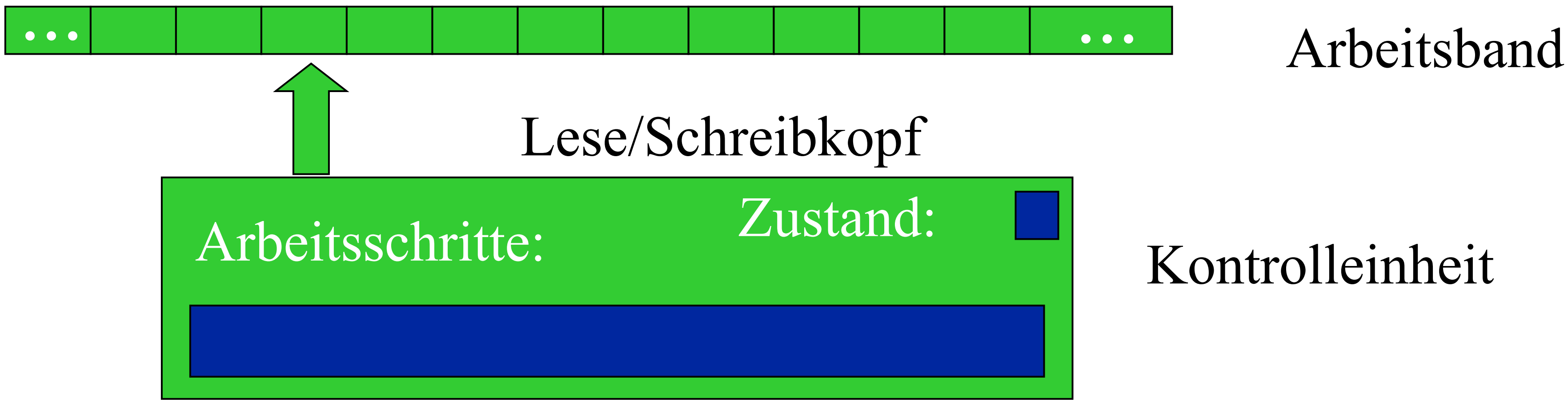
  Eine oft zitierte Erweiterung ist die Mehrband-Turingmaschine, die aber nicht mächtiger als die klassische Einband-TM ist, die in dieser VO behandelt wird.

Aufbau einer Turingmaschine



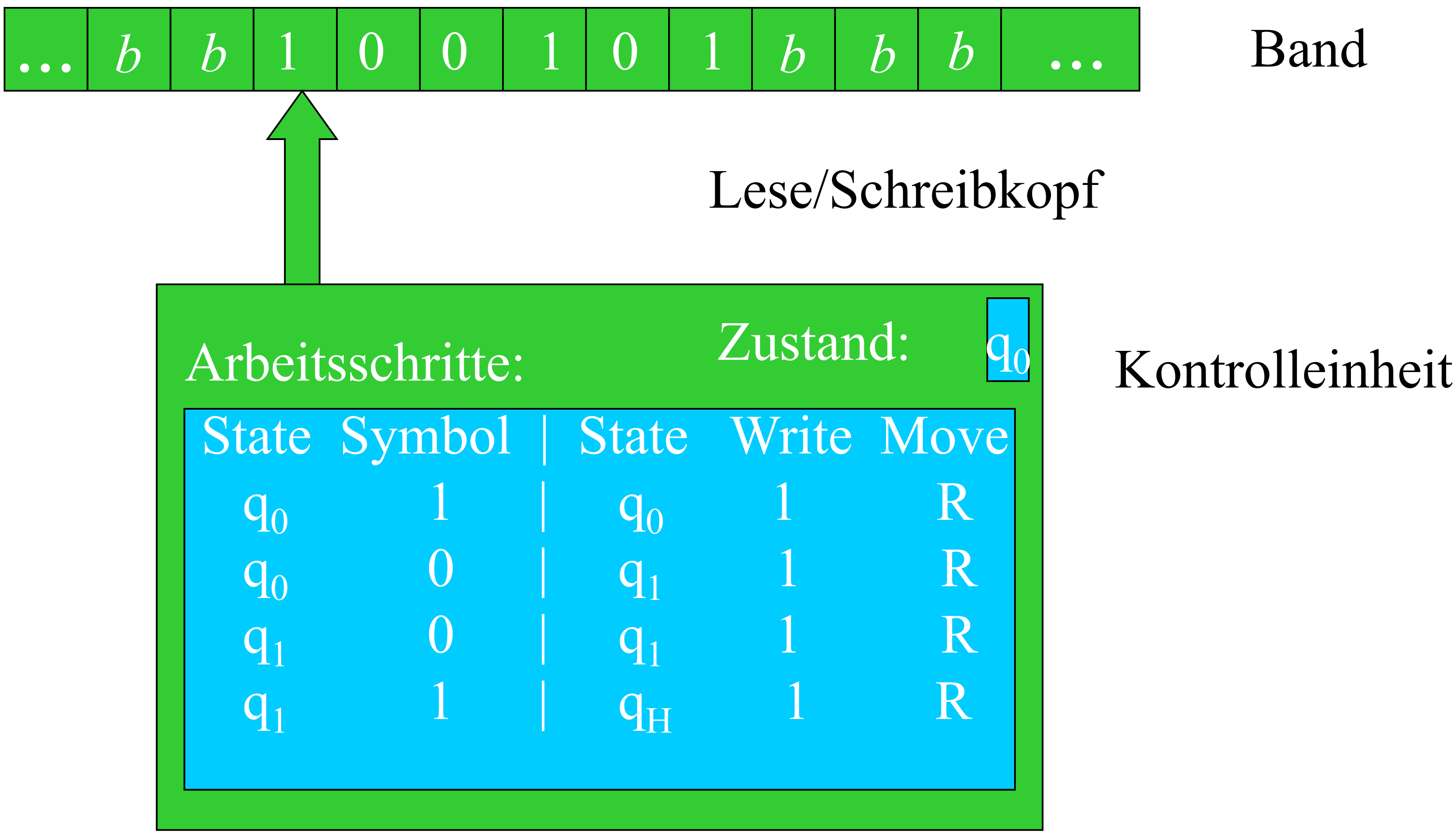
- Eine TM enthält eine **Kontrolleinheit** mit endlich vielen **Zuständen**.
- Die Kontrolleinheit ist über einen **Lese/Schreibkopf** (LS-Kopf) mit einem Speicher verbunden, der durch ein eindimensionales beidseitig unbegrenztes **Arbeitsband** repräsentiert wird.
- Das Arbeitsband ist in Felder unterteilt; in jedes Feld kann höchstens ein Symbol geschrieben werden.

Arbeitsweise einer Turingmaschine



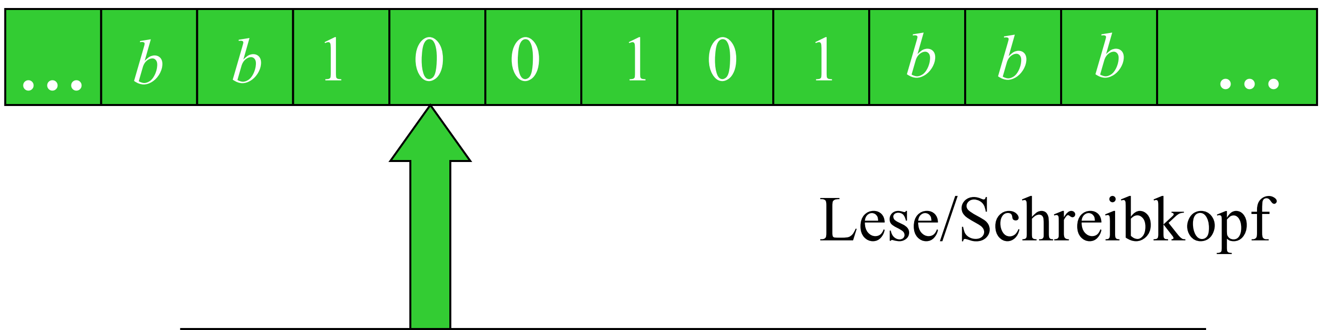
- Ausführung der angegebenen **Arbeitsschritte**.
- Arbeitsschritt ist abhängig vom Inhalt des Feldes unter Lese/Schreibkopf und vom aktuellen Zustand.
- Ein Arbeitsschritt besteht aus:
  - Schreibe ein Symbol in das Feld unter dem LS-Kopf.
  - Bewege LS-Kopf um ein Feld nach links oder rechts.
  - Gehe in einen neuen Zustand.

Beispiel: TM (1)



State	Symbol	|	State	Write	Move
q_0	1	|	q_0	1	R
q_0	0	|	q_1	1	R
q_1	0	|	q_1	1	R
q_1	1	|	q_H	1	R

Beispiel: TM (2)

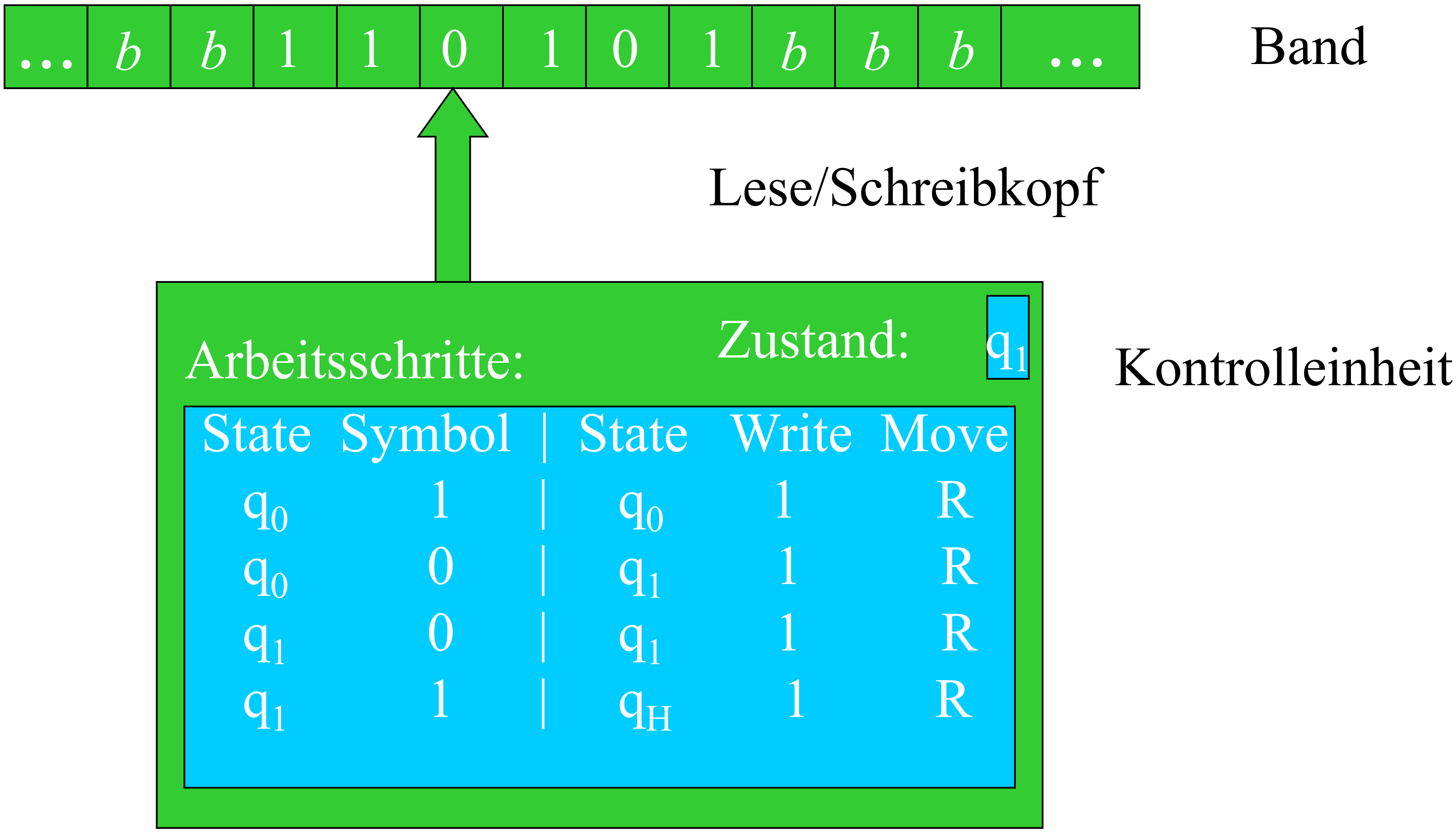


Arbeitsschritte:

Zustand: q_0

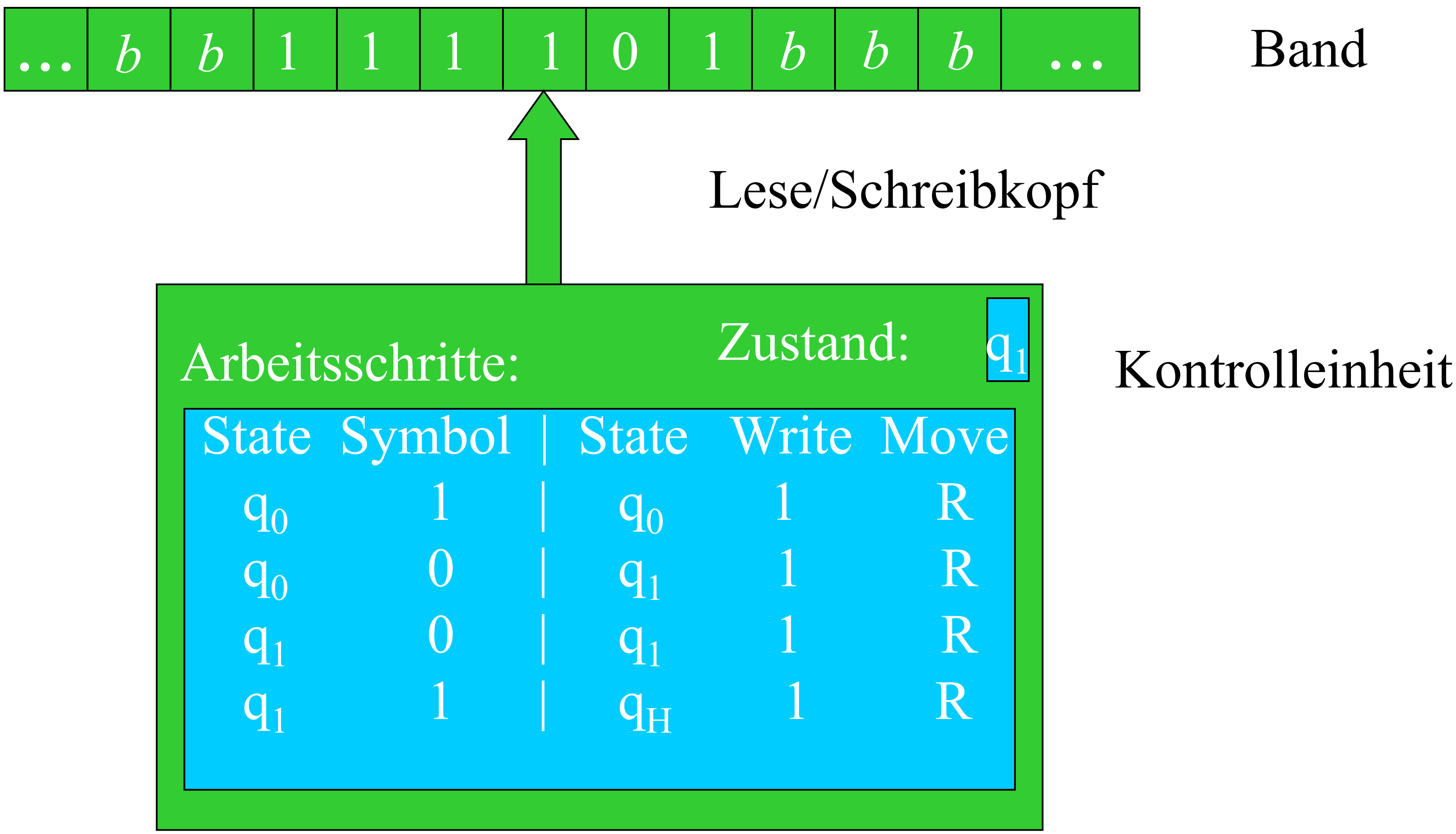
State	Symbol	State	Write	Move
q_0	1	q_0	1	R
q_0	0	q_1	1	R
q_1	0	q_1	1	R
q_1	1	q_H	1	R

Beispiel: TM (3)



State	Symbol	State	Write	Move
q_0	1	q_0	1	R
q_0	0	q_1	1	R
q_1	0	q_1	1	R
q_1	1	q_H	1	R

Beispiel: TM (4)



Beispiel: TM (5)

…	*b*	*b*	1	1	1	1	0	1	*b*	*b*	*b*	…

Band

Lese/Schreibkopf

Kontrolleinheit

Zustand: q_H

Arbeitsschritte:

State	Symbol	State	Write	Move
q_0	1	q_0	1	R
q_0	0	q_1	1	R
q_1	0	q_1	1	R
q_1	1	q_H	1	R

TM: Formale Definition

Def.: Eine Turingmaschine ist ein 7-Tupel $M=(Q,\Sigma,\Gamma,b,\delta,q_0,F)$, wobei

- Q eine endliche Menge von **Zuständen**,
- Σ das **Eingabealphabet** mit $\Sigma \subseteq \Gamma$,
- Γ eine endliche Menge von **Bandzeichen**,
- Blank b, ein Symbol aus Γ, um leere Felder zu markieren,
- die **Überführungsfunktion** δ (partiell),

  δ: $(Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{ L, R, N \}$ mit

  L: ein Feld nach links, R: ein Feld nach rechts, N: keine Bewegung,
- q_0 der Startzustand und
- $F \subseteq Q$ die Menge der Endzustände ist.

TM: Formale Definition - Struktur

Die **Struktur** einer TM $M=(Q,\Sigma,\Gamma,b,\delta,q_0,F)$ kann man sich bestehend aus einer Kontrolleinheit, einem Arbeitsspeicher und einem Lese/Schreibkopf vorstellen.

- Die Kontrolleinheit kann Zustände aus Q annehmen.
- Das Arbeitsband ist in Felder unterteilt, die Symbole aus dem Alphabet Γ enthalten. Zu jedem Zeitpunkt sind höchstens endlich viele Felder beschriftet, die die **Bandinschrift** ergeben.
- Der L/S-Kopf verbindet die Kontrolleinheit mit dem Arbeitsband

TM: Formale Definition - Funktionsweise

Ein **Arbeitsschritt** führt eine Konfiguration in eine Nachfolgekonfiguration wie folgt über:
Gegeben sei eine Konfiguration von M mit einer bestimmten Bandinschrift, sei q der Zustand der Kontrolleinheit, y das Zeichen unter dem L/S-Kopf, und sei (q,y) im Definitionsbereich der Überführungsfunktion mit

- $\delta(q,y) = (q_1,y_1,\mathrm{d})$,

wobei $q_1 \in Q$, $y_1 \in \Gamma$ und $\mathrm{d} \in \{ L,R,N \}$. Dann lässt sich die neue Konfiguration von M wie folgt beschreiben:

1. die Kontrolleinheit geht in den Zustand q_1 über
2. in das Feld unter dem L/S-Kopf wird y_1 geschrieben
3. der L/S-Kopf bewegt sich um ein Feld nach links falls d=L, nach rechts falls d=R und bewegt sich nicht falls d=N
4. die Bandinschrift bleibt bis auf Änderung durch Schritt 2 gleich

TM: Anhalten

Eine TM **<u>hält</u>**, wenn:

1. es für eine Konfiguration keine Nachfolgekonfiguration gibt (man beachte, dass die δ-Funktion partiell ist, d.h. nicht für alle Eingabezeichen definiert sein muss)

2. die TM in einen Endzustand gelangt (man beachte, dass ein Endzustand keine Nachfolgezustände hat, d.h. ein <u>Endzustand kann nicht mehr verlassen werden</u>)

Um ein Wort zu akzeptieren, muss die TM in einen Endzustand gelangen. Ob dabei das gesamte Wort abgearbeitet bzw. gelesen wurde, ist unerheblich (siehe folg. Folien). Man beachte die Unterschiede gegenüber Automaten.

TM: Formale Definition - Spezifikation (1)

- **Tabelle** (Turingtafel): Beschreibung von Überführungsfunktion δ

  Beispiel:

δ	0	1	$\boldsymbol{b}$
q_0	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_H, \boldsymbol{b}, N)$
q_1	$(q_2, 0, R)$	$(q_2, 1, R)$	-
q_2	$(q_0, 0, R)$	$(q_0, 1, R)$	-

- **Quintupel:**
  Die Spezifikation der TM erfolgt durch eine Menge von Quintupeln (q,y,q_1,y_1,d), wobei $\delta(q,y) = (q_1,y_1,d)$.

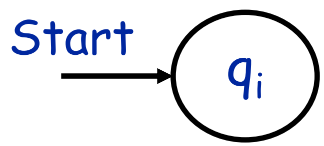
- Für eine **deterministische** Turingmaschine gilt:
  $|\ \delta(q,y)\ | \leq 1$ für alle $(q,y) \in Q \times \Gamma$.

TM: Formale Definition - Spezifikation (2)

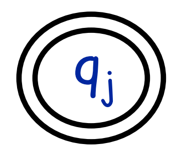
- **Übergangsdiagramm**
  - Knoten: Zustände der TM
  - Kanten: Übergänge von q nach p werden markiert mit

wobei $\delta(s,X)=(t,Y,D)$ mit $D = L \vee R \vee N$.

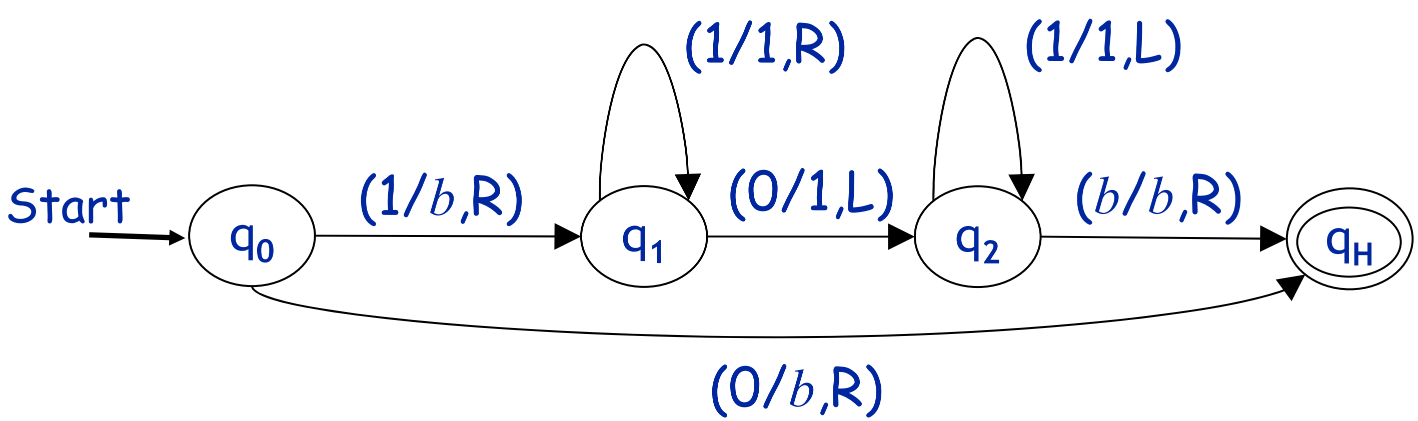
  - Startzustand:



Endzustand:



- **Beispiel:**



Konfigurationen einer TM

Eine Konfiguration einer TM beschreibt eine Momentaufnahme einer TM während der Abarbeitung, d.h. alle aktuellen Werte.

Die **Konfiguration** einer TM M ist daher durch folgende Angaben bestimmt:

- die Bandinschrift, den Zustand der Kontrolleinheit und die Position des L/S-Kopfs.

Notation:

- Sequenz $\boxed{A_1 A_2 \ldots A_{i-1} q\ A_i \ldots A_n}$ bezeichnet folgende Konfiguration:
- Bandinschrift $A_1 \ldots A_n$ und M befindet sich in q und liest A_i (Symbol rechts von q).

Startkonfiguration: $\boxed{q_0\ w}$, unter der Annahme, dass q_0 der Startzustand ist und Eingabewort **w** am Arbeitsband steht.

Konfigurationsrelation $\vdash$

1. $\delta(q,A_i) = (p,B,\mathrm{L})$

   $A_1\, A_2 \ldots A_{i-1}\, q\, A_i\, A_{i+1} \ldots A_n \vdash A_1\, A_2 \ldots p\, A_{i-1}\, B\, A_{i+1} \ldots A_n$

2. $\delta(q,A_i) = (p,B,\mathrm{R})$

   $A_1\, A_2 \ldots A_{i-1}\, q\, A_i\, A_{i+1} \ldots A_n \vdash A_1\, A_2 \ldots A_{i-1}\, B\, p\, A_{i+1} \ldots A_n$

3. Wie üblich, $\vdash^*$ bezeichnet 0,1, oder mehrere Schritte.

Akzeptierte Sprachen

- Die von M **akzeptierte Sprache** ist die Menge aller Wörter $w \in \Sigma^*$, die bei Eingabe von *w* die Turingmaschine M in einen Endzustand überführt. Eingabe von *w* bedeutet, daß *w* auf das Arbeitsband geschrieben wird, der L/S-Kopf das am weitesten links stehende Symbol von *w* liest und M sich im Zustand q_0 (links und rechts von *w* sind unendl. viele Blanks).
- Formal ist die durch $M = (Q, \Sigma, \Gamma, b, \delta, q_0, F)$ **akzeptierte Sprache** definiert wie folgt:

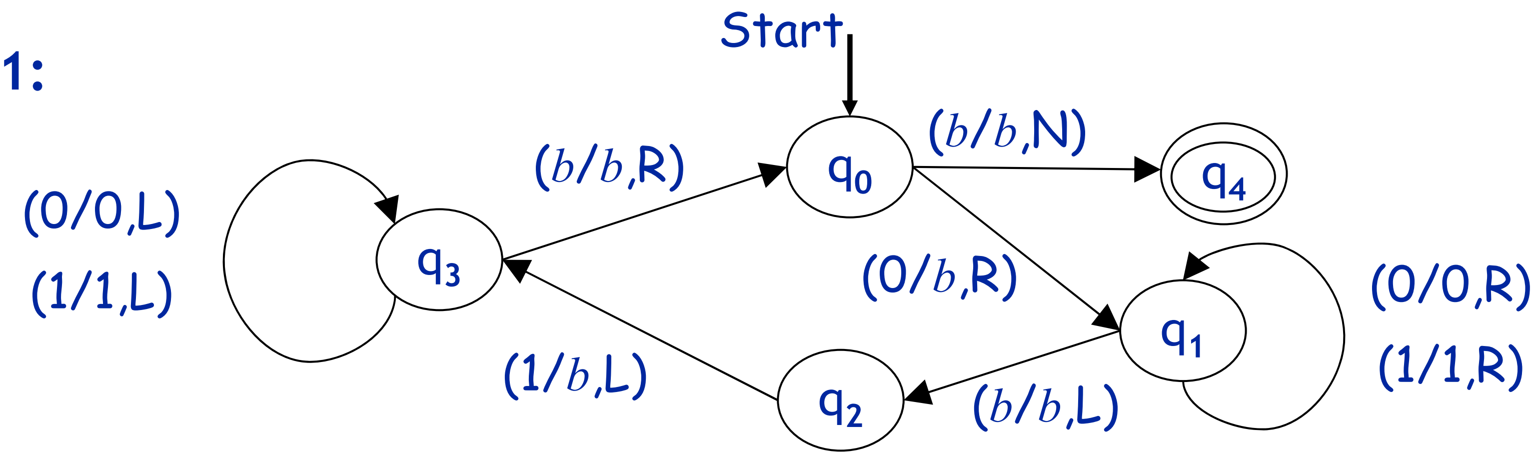
  $$L(M) = \{ w \mid w \in \Sigma^* \text{ und } q_0 w \overset{*}{\underset{M}{\vdash}} a_1 q\, a_2 \text{ mit } q \in F \text{ und } a_1, a_2 \in \Gamma^* \}.$$

- Die Bandinschrift ist nach Akzeptanz unerheblich.

Beispiel Akzeptieren

Man konstruiere eine Turingmaschine M, die die Sprache
$L = \{ 0^n 1^n \mid n \geq 0 \}$ akzeptiert. Hilfsskizze Band: $\ldots\ b\ \cancel{0}\ \cancel{0}\ 0\ 1\ \cancel{1}\ \cancel{1}\ b \ldots$

Lösung 1:



$M = (Q, \Sigma, \Gamma, b, \delta, q_0, F)$
$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$, $\Sigma = \{ 0, 1 \}$, $\Gamma = \Sigma \cup \{ b \}$
$F = \{ q_4 \}$,
$\delta = \{(q_0, 0, q_1, b, R)$, *// lösche erste 0;*
$(q_1, 0, q_1, 0, R)$, $(q_1, 1, q_1, 1, R)$, *// laufe nach rechts;*
(q_1, b, q_2, b, L), $(q_2, 1, q_3, b, L)$, *// lösche letzte 1;*
$(q_3, 0, q_3, 0, L)$, $(q_3, 1, q_3, 1, L)$, *// laufe nach links;*
(q_3, b, q_0, b, R), *// beginne wieder von vorne;*
$(q_0, b, q_4, b, N)\}$ *// keine 0 oder 1 vorhanden - akzeptiere;*

Beispiel Akzeptieren - Fortsetzung

Lösung 1: Abarbeitung bei Eingabe **000111**

- $q_0000111 \vdash q_100111 \vdash 0q_10111 \vdash 00q_1111 \vdash 001q_111 \vdash 0011q_11 \vdash$
  $\vdash 00111q_1b \vdash 0011q_21b \vdash 001q_31bb \vdash 00q_311 \vdash 0q_3011 \vdash q_30011 \vdash$
  $\vdash q_3b0011 \vdash$
- $q_00011 \vdash q_1011 \vdash 0q_111 \vdash 01q_11 \vdash 011q_1b \vdash 01q_21b \vdash 0q_31bb \vdash q_301 \vdash$
  $\vdash q_3b01 \vdash$
- $q_001 \vdash q_11 \vdash 1q_1b \vdash q_21b \vdash q_3bbb \vdash bq_0bb \vdash q_4b$

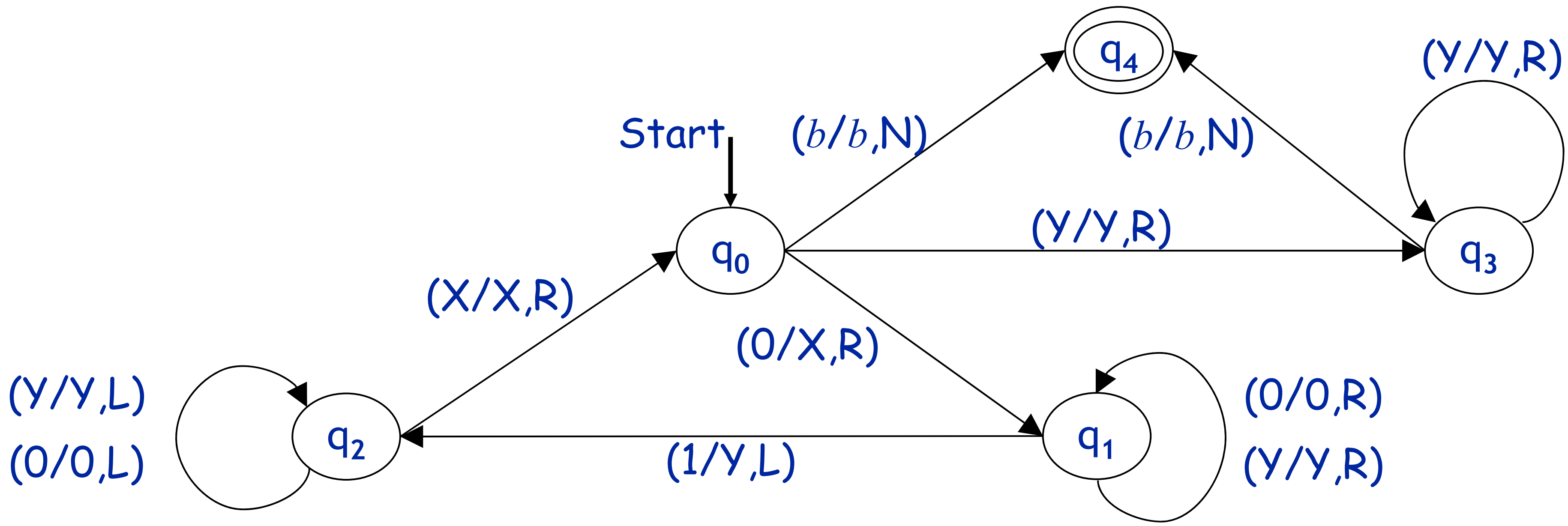
Nochmals Überführungsfunktion:

$\delta=\{(q_0,0, q_1,b,R),$
$(q_1,0, q_1,0,R), (q_1,1, q_1,1,R),$
$(q_1,b, q_2,b,L), (q_2,1, q_3,b,L),$
$(q_3,0, q_3,0,L), (q_3,1, q_3,1,L),$
$(q_3,b, q_0,b,R),$
$(q_0,b, q_4,b,N)\}$

Beispiel Akzeptieren - Fortsetzung

Lösung 2a: Es werden die Symbole X,Y zusätzlich zu Σ verwendet, es wird nicht am Ende des Wortes ersetzt,

Hilfsskizze Band: … b X X 0 0 Y Y 1 1 b …



$M = (Q, \Sigma, \Gamma, b, \delta, q_0, F)$
$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$, $\Sigma = \{ 0, 1 \}$, $\Gamma = \Sigma \cup \{ X, Y, b \}$
$F = \{ q_4 \}$,

Beispiel Akzeptieren - Fortsetzung

Lösung 2b: Wie Lösung 2a, mit Quintupel spezifiziert.

Hilfsskizze Band: … b X X 0 0 Y Y 1 1 b …

$M = (Q, \Sigma, \Gamma, b, \delta, q_0, F)$
$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$, $\Sigma = \{ 0, 1 \}$, $\Gamma = \Sigma \cup \{ X, Y, b \}$
$F = \{ q_4 \}$,
$\delta = \{(q_0,0, q_1,X,R)$, *// Beginn der Schleife: Ersetze 0 durch X;*
$(q_1,0, q_1,0,R)$, (q_1,Y, q_1,Y,R), *// laufe bis zur ersten 1*
$(q_1,1, q_2,Y,L)$, *// und ersetze 1 durch Y;*
(q_2,Y, q_2,Y,L), $(q_2,0, q_2,0,L)$, *// laufe zurück zum ersten*
(q_2,X, q_0,X,R) *// X u. wiederhole Schleife;*
(q_0,Y, q_3,Y,R), *// keine 0 mehr vorhanden, akzeptiere*
(q_3,Y, q_3,Y,R), *// falls auch keine 1 mehr vorhanden ist;*
(q_3,b, q_4,b,N),
$(q_0,b, q_4,b,N)\}$ *// akzeptiere auch falls keine 0 und 1;*

Berechnungen mit Turingmaschinen

- Turingmaschinen können zur Berechnung von Funktionen benutzt werden.
- Zu Beginn einer Berechnung wird das Argument für die zu berechnende Funktion auf dem Band notiert. Dies ist die **Eingabe** für die TM. Wenn die TM nach endlich vielen Schritten im Endzustand hält, dann ist die Bandinschrift zu diesem Zeitpunkt die **Ausgabe** der TM und bezeichnet den Funktionswert.
- Codierung von natürlichen Zahlen *nat*: $\mathbb{N}_0 \to \{ 1 \}^*$
  d.h. eine natürliche Zahl $i \geq 0$ wird durch die Zeichenfolge 1^i repräsentiert – eine Aufeinanderfolge von 1er.

Berechnungen mit Turingmaschinen: Beispiel 1

- Sei f: $\mathbb{N}_0 \rightarrow \mathbb{N}$ definiert durch $f(x) = \begin{cases} 1, & \text{falls x durch 3 teilbar} \\ \text{undefiniert sonst} & \end{cases}$

  Man konstruiere eine TM, die f berechnet (Codierung *nat*).

- **Lösung:**

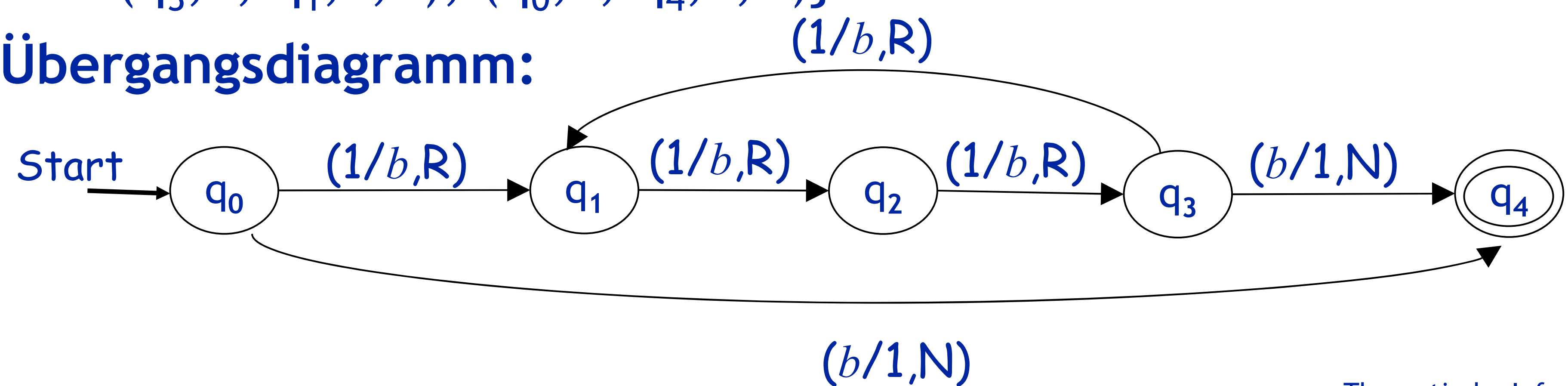
  $M = (Q, \Sigma, \Gamma, b, \delta, q_0, F)$

  $Q = \{ q_0, q_1, q_2, q_3, q_4 \}$, $\Sigma = \{ 1 \}$, $\Gamma = \Sigma \cup \{ b \}$

  $F = \{ q_4 \}$,

  $\delta = \{(q_0, 1, q_1, b, R), (q_1, 1, q_2, b, R), (q_2, 1, q_3, b, R), (q_3, b, q_4, 1, R),$
  $(q_3, 1, q_1, b, R), (q_0, b, q_4, 1, N)\}$

- **Übergangsdiagramm:**



Berechnungen mit Turingmaschinen: Beispiel 2

- Gegeben sei $f: \mathbb{N}_0^k \to \mathbb{N}_0^l$ mit $f(m_1,m_2,\ldots,m_k)=(n_1,n_2,\ldots,n_l)$
- Dann wird TM_f mit folgendert initialen Konfiguration gestartet:
  - Bandinschrift: $\ldots b\ 1^{m_1}\ 0\ 1^{m_2}\ 0 \ldots 0\ 1^{m_k}\ b \ldots$ (Verwendung der Codierungsfunktion *nat*)
  - L/S-Kopf über am weitesten links stehenden Symbol $\neq b$
- Terminiert TM_f, dann soll folgende Konfiguration gegeben sein:
  - Bandinschrift: $\ldots b\ 1^{n_1}\ 0\ 1^{n_2}\ 0 \ldots 0\ 1^{n_l}\ b \ldots$ (Verwendung der Codierungsfunktion *nat*)
  - L/S-Kopf soll wieder über am weitesten links stehenden Symbol $\neq b$ positioniert werden.

Berechnungen mit Turingmaschinen: Beispiel 2

- Sei g: $\mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch g(m,n)=m+n.

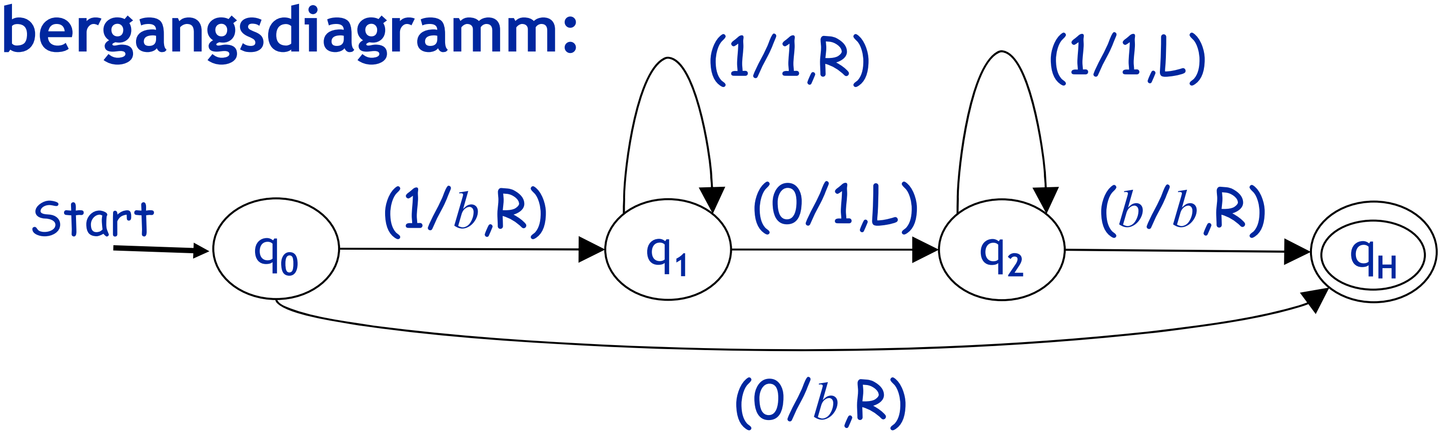
  Man konstruiere eine TM, die g berechnet.
- **Lösung:** $(\ldots b\ 1^m\ 0\ 1^n\ b \ldots)$

  $M = (Q, \Sigma, \Gamma, b, \delta, q_0, F)$

  $Q = \{ q_0, q_1, q_2, q_H \}$, $\Sigma = \{ 1,0 \}$, $\Gamma = \Sigma \cup \{ b \}$

  $F = \{ q_H \}$,

  $\delta = \{ (q_0,0, q_H,b,R), (q_0,1, q_1,b,R), (q_1,1, q_1,1,R), (q_1,0, q_2,1,L),$
  $(q_2,1, q_2,1,L), (q_2,b, q_H,b,R) \}$
- **Übergangsdiagramm:**



Beispiel 3: Akzeptieren der Sprache vv^R

- Es wurde bei Kellerautomaten gezeigt, dass es für die Sprache $L = \{v\,v^R \mid v \in \{0,1\}^*\}$, wobei v^R das gespiegelte Wort von v bezeichnet, keinen deterministischen Kellerautomaten zur Akzeptanz gibt, i.e. für Worte der Form 001100.
- Diese Sprache kann sehr einfach mit einer deterministischen TM akzeptiert werden. Die Strategie ist ähnlich wie bei der präsentierten TM für die Sprache $L = \{0^n1^n \mid n \geq 0\}$. Ausführlicher dazu in der VO.
- Wie sieht es aber mit Sprachen wie $L = \{v\,v \mid v \in \{0,1\}^*\}$ oder $L = \{v\_v \mid v \in \{0,1\}^*\}$ aus? D.h. Worte wie 001001 oder 001_001. Dieses wird näher in der Übungseinheit behandelt.

Linear beschränkter Automat

- Eine **linear beschränkter Automat** (LBA) ist eine nicht-deterministische Turingmaschine, die folgende zwei Bedingungen erfüllt:
  1. Das Eingabealphabet beinhaltet zwei Spezialsymbole $\$_{left}$ und $\$_{right}$ für linke und rechte Endmarkierungen des Eingabebandes.
  2. Der LBA darf sich nur innerhalb von $\$_{left}$ und $\$_{right}$ bewegen und darf $\$_{left}$ bzw. $\$_{right}$ nicht überschreiben.

- LBA sind Chomsky Typ-1 Akzeptoren.

Tools zur Simulation von Automaten

- **JFLAP**: Java Formal Languages and Automata Package
- Website: https://www.jflap.org

- unterstützt Automaten, Kellerautomaten und Turingmaschinen
- als Lernhilfe gedacht
- bei etwaigen Unterschieden gilt immer die in der VO gewählte Version
- weitere Informationen dazu im Tutorium