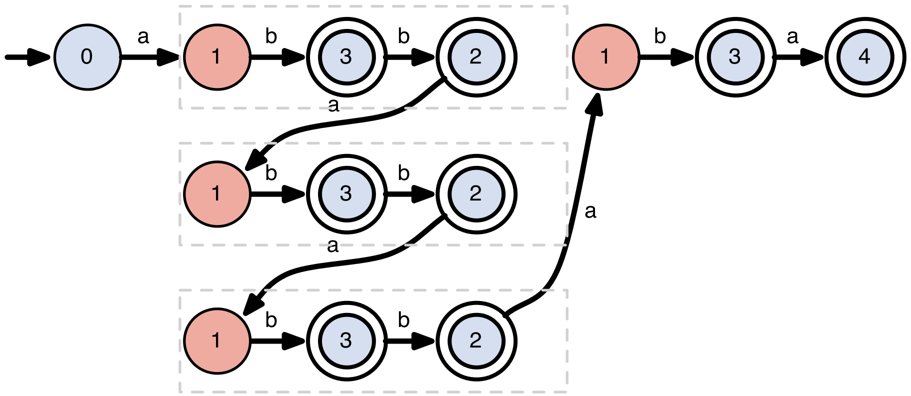
Berechenbarkeitstheorie

12. Vorlesung



Dr. Franziska Jahnke
Institut für Mathematische Logik und Grundlagenforschung
WWU Münster

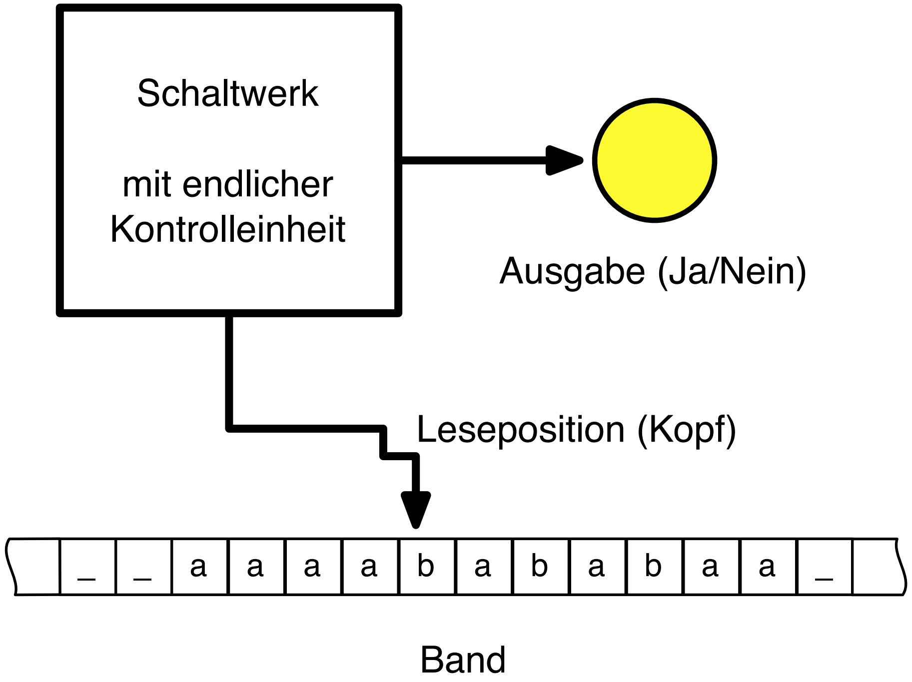
Turingmaschinen

Turingmaschinen

- **Ziel:** starkes Berechnungsmodell, welches die Arbeitsweise eines idealisierten *Computers* nachbildet
- eingeführt 1936 von Alan Turing

Turingmaschinen

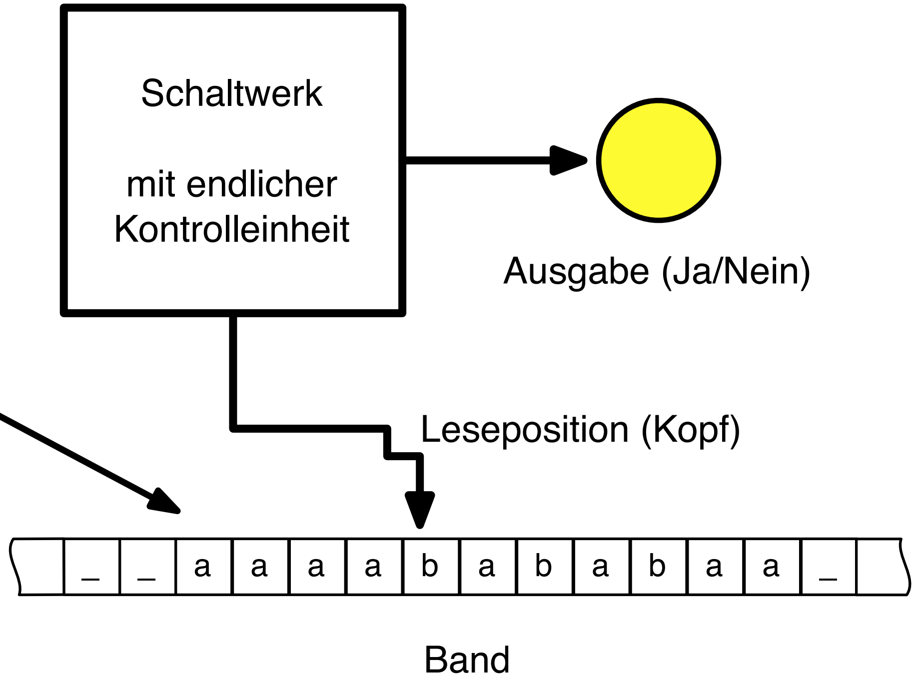
- **Ziel:** starkes Berechnungsmodell, welches die Arbeitsweise eines idealisierten *Computers* nachbildet
- eingeführt 1936 von Alan Turing



Turingmaschinen

- **Ziel:** starkes Berechnungsmodell, welches die Arbeitsweise eines idealisierten *Computers* nachbildet
- eingeführt 1936 von Alan Turing

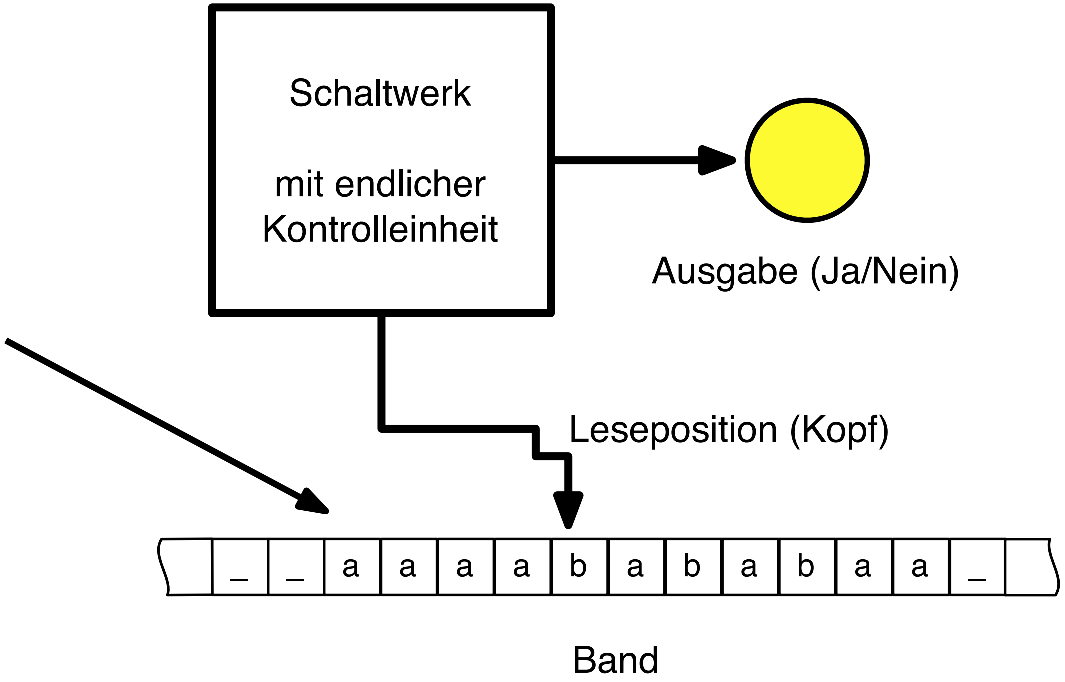
Kopf kann auf dem unbeschränktem Schreib-Lese-Band frei nach links und rechts bewegt werden werden (in 1er Schritten)



Turingmaschinen

- **Ziel:** starkes Berechnungsmodell, welches die Arbeitsweise eines idealisierten *Computers* nachbildet
- eingeführt 1936 von Alan Turing

Kopf kann auf dem unbeschränktem Schreib-Lese-Band frei nach links und rechts bewegt werden werden (in 1er Schritten)



- Übergänge vom Zeichen an Kopfposition abhängig

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

- Konfiguration $=$ Bandinhalt $+$ Kopfposition $+$ aktueller Zustand

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

- Konfiguration $=$ Bandinhalt $+$ Kopfposition $+$ aktueller Zustand
- Bandinhalt $=$ **beschriebene Zeichen** des Bandes$=$ endliches Wort

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

- Konfiguration = Bandinhalt + Kopfposition + aktueller Zustand
- Bandinhalt = **beschriebene Zeichen** des Bandes= endliches Wort
- Default Symbol des Bandes heißt Blank-Symbol, $\square$

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

- Konfiguration $=$ Bandinhalt $+$ Kopfposition $+$ aktueller Zustand
- Bandinhalt $=$ **beschriebene Zeichen** des Bandes$=$ endliches Wort
- Default Symbol des Bandes heißt Blank-Symbol, $\square$
- Schreibweise für Konfigurationen: $w_1 q w_2$

Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

- Konfiguration $=$ Bandinhalt $+$ Kopfposition $+$ aktueller Zustand
- Bandinhalt $=$ **beschriebene Zeichen** des Bandes$=$ endliches Wort
- Default Symbol des Bandes heißt Blank-Symbol, $\square$
- Schreibweise für Konfigurationen: $w_1 q w_2$
  $\rightarrow$ Bandinhalt $w_1 w_2$, Kopf hinter w_1, Zustand q

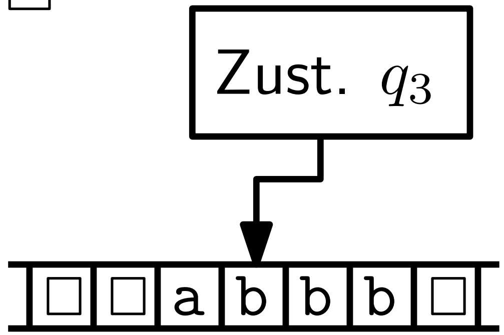
Definition

Eine **Turingmachine (TM)** ist ein 7-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$, mit:

- der Zustandsmenge Q,
- dem Eingabealphabet Σ, dem Arbeitsalphabet Γ,
- der Übergangsfunktion $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- $q_0, q_A, q_V \in Q$ mit $q_A \neq q_V$.

Konfiguration einer TM

- Konfiguration $=$ Bandinhalt $+$ Kopfposition $+$ aktueller Zustand
- Bandinhalt $=$ **beschriebene Zeichen** des Bandes= endliches Wort
- Default Symbol des Bandes heißt Blank-Symbol, $\square$
- Schreibweise für Konfigurationen: $w_1 q w_2$
  $\to$ Bandinhalt $w_1 w_2$, Kopf hinter w_1, Zustand q



Beispiel: a$q_3$bbb

Akzeptanz bei Turingmaschinen

Akzeptanz bei Turingmaschinen

Definition

Eine Turingmaschine M **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann wenn eine Folge von nicht-verwerfenden Konfigurationen $C_1, \ldots, C_n$ existiert mit:

1. $C_1 = q_0\, w$ (Startkonfiguration)
2. $\forall i = 1, \ldots, n-1$: C_i geht über nach C_{i+1}
3. C_n akzeptierend

Akzeptanz bei Turingmaschinen

Definition

Eine Turingmaschine M **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann wenn eine Folge von nicht-verwerfenden Konfigurationen $C_1, \ldots, C_n$ existiert mit:

Lauf der TM

1. $C_1 = q_0\, w$ (Startkonfiguration)
2. $\forall i = 1, \ldots, n-1$: C_i geht über nach C_{i+1}
3. C_n akzeptierend

Akzeptanz bei Turingmaschinen

Definition

Eine Turingmaschine M **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann wenn eine Folge von nicht-verwerfenden Konfigurationen $\underbrace{C_1, \ldots, C_n}_{}$ existiert mit:

Lauf der TM

1. $C_1 = q_0\, w$ (Startkonfiguration)
2. $\forall i = 1, \ldots, n-1$: C_i geht über nach C_{i+1}
3. C_n akzeptierend

Def.:

Eine TM M **erkennt** die Sprache $\{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$

Akzeptanz bei Turingmaschinen

Definition

Eine Turingmaschine M **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann wenn eine Folge von nicht-verwerfenden Konfigurationen $C_1, \ldots, C_n$ existiert mit:

Lauf der TM

1. $C_1 = q_0\, w$ (Startkonfiguration)
2. $\forall i = 1, \ldots, n-1$: C_i geht über nach C_{i+1}
3. C_n akzeptierend

Def.:

Eine TM M **erkennt** die Sprache $\{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$

2 Möglichkeiten für Nichtakzeptanz

Akzeptanz bei Turingmaschinen

Definition

Eine Turingmaschine M **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann wenn eine Folge von nicht-verwerfenden Konfigurationen $C_1, \dots, C_n$ existiert mit:

Lauf der TM

1. $C_1 = q_0\, w$ (Startkonfiguration)
2. $\forall i = 1, \dots, n-1$: C_i geht über nach C_{i+1}
3. C_n akzeptierend

Def.:

Eine TM M **erkennt** die Sprache $\{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$

2 Möglichkeiten für Nichtakzeptanz

① Lauf führt zu einer Konfiguration mit q_V

Akzeptanz bei Turingmaschinen

Definition

Eine Turingmaschine M **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann wenn eine Folge von nicht-verwerfenden Konfigurationen $C_1, \ldots, C_n$ existiert mit:

Lauf der TM

1. $C_1 = q_0\, w$ (Startkonfiguration)
2. $\forall i = 1, \ldots, n-1:\ C_i$ geht über nach C_{i+1}
3. C_n akzeptierend

Def.:

Eine TM M **erkennt** die Sprache $\{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$

2 Möglichkeiten für Nichtakzeptanz

① Lauf führt zu einer Konfiguration mit q_V

② Lauf erreicht weder q_V noch q_A
(M stoppt nicht)

Angabe von TM Programmen

Angabe von TM Programmen

① Diagrammform

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben
- Fehlende Übergänge im Diagramm würden nach q_V überführen

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben
- Fehlende Übergänge im Diagramm würden nach q_V überführen

② Modulbeschreibung

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben
- Fehlende Übergänge im Diagramm würden nach q_V überführen

② Modulbeschreibung

- Programm wird verbal beschrieben

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben
- Fehlende Übergänge im Diagramm würden nach q_V überführen

② Modulbeschreibung

- Programm wird verbal beschrieben
- Abfolge von Modulen (=Befehlssequenzen)

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben
- Fehlende Übergänge im Diagramm würden nach q_V überführen

② Modulbeschreibung

- Programm wird verbal beschrieben
- Abfolge von Modulen (=Befehlssequenzen)
- Module sind problemlos als TM-Teilprogramm formulierbar
- Bsp.:
  Gehe ans Ende des Bandes links und ersetze jede zweite 1 durch eine 0

Angabe von TM Programmen

① Diagrammform

- Programm wird präzise angegben
- Spezifikation von Q und Γ
- Übergangsfunktion δ wird in Diagrammform angegeben
- Fehlende Übergänge im Diagramm würden nach q_V überführen

② Modulbeschreibung

- Programm wird verbal beschrieben
- Abfolge von Modulen (=Befehlssequenzen)
- Module sind problemlos als TM-Teilprogramm formulierbar
- Bsp.:
  Gehe ans Ende des Bandes links und ersetze jede zweite 1 durch eine 0
- Bsp.:
  Merke das aktuelle Zeichen im Zustand, gehe bis zum ersten $ nach rechts und ersetze $ durch das gemerkte Zeichen

Beispiel $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 1\}$

1. Test ob die Eingabe die Form $\mathtt{a}^+\mathtt{b}^+\mathtt{c}^+$ hat
2. Laufe zum ersten a links, ersetze a durch x
3. Laufe zum ersten nicht-a nach rechts (ignoriere x)
   3.1. Ist dieses Zeichen kein b verwerfe,
   3.2. ansonsten ersetzte das b durch x
4. Laufe zum ersten nicht-b nach rechts (ignoriere x)
   4.1. Ist dieses Zeichen kein c verwerfe,
   4.2. ansonsten ersetzte das c durch x
5. Laufe zum ersten Blank links
6. Laufe zum ersten nicht-x nach rechts
   6.1. Ist dieses Zeichen ein Blank, dann akzeptiere,
   6.2. ist es ein a ersetze es durch x und wiederhole ab 3.,
   6.3. ansonsten verwerfe

Definitionen

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt
- Eine Sprache L heißt **entscheidbar**, gdw. eine Entscheider existiert, der L erkennt

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt
- Eine Sprache L heißt **entscheidbar**, gdw. eine Entscheider existiert, der L erkennt
- Menge der erkennbaren Sprachen $\mathbb{A}$

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt
- Eine Sprache L heißt **entscheidbar**, gdw. eine Entscheider existiert, der L erkennt
- Menge der erkennbaren Sprachen $\mathbb{A}$
- Menge der entscheidbaren Sprachen $\mathbb{E}$

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt
- Eine Sprache L heißt **entscheidbar**, gdw. eine Entscheider existiert, der L erkennt
- Menge der erkennbaren Sprachen $\mathbb{A}$
- Menge der entscheidbaren Sprachen $\mathbb{E}$
- Schreibweise: Turingmaschine M mit Eingabe w : $M(w)$

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt
- Eine Sprache L heißt **entscheidbar**, gdw. eine Entscheider existiert, der L erkennt
- Menge der erkennbaren Sprachen $\mathbb{A}$
- Menge der entscheidbaren Sprachen $\mathbb{E}$
- Schreibweise: Turingmaschine M mit Eingabe w : $M(w)$

  Turingmaschine M mit Eingabe w stoppt: $M(w)\downarrow$

Definitionen

- Eine Turingmaschine heißt **Entscheider**, wenn sie auf allen Eingaben stoppt.
- Eine Sprache L heißt **erkennbar**, gdw. eine Turingmaschine existiert, die L erkennt
- Eine Sprache L heißt **entscheidbar**, gdw. eine Entscheider existiert, der L erkennt
- Menge der erkennbaren Sprachen $\mathbb{A}$
- Menge der entscheidbaren Sprachen $\mathbb{E}$
- Schreibweise: Turingmaschine M mit Eingabe w : $M(w)$

Turingmaschine M mit Eingabe w stoppt: $M(w)\downarrow$

Turingmaschine M mit Eingabe w stoppt nicht: $M(w)\uparrow$

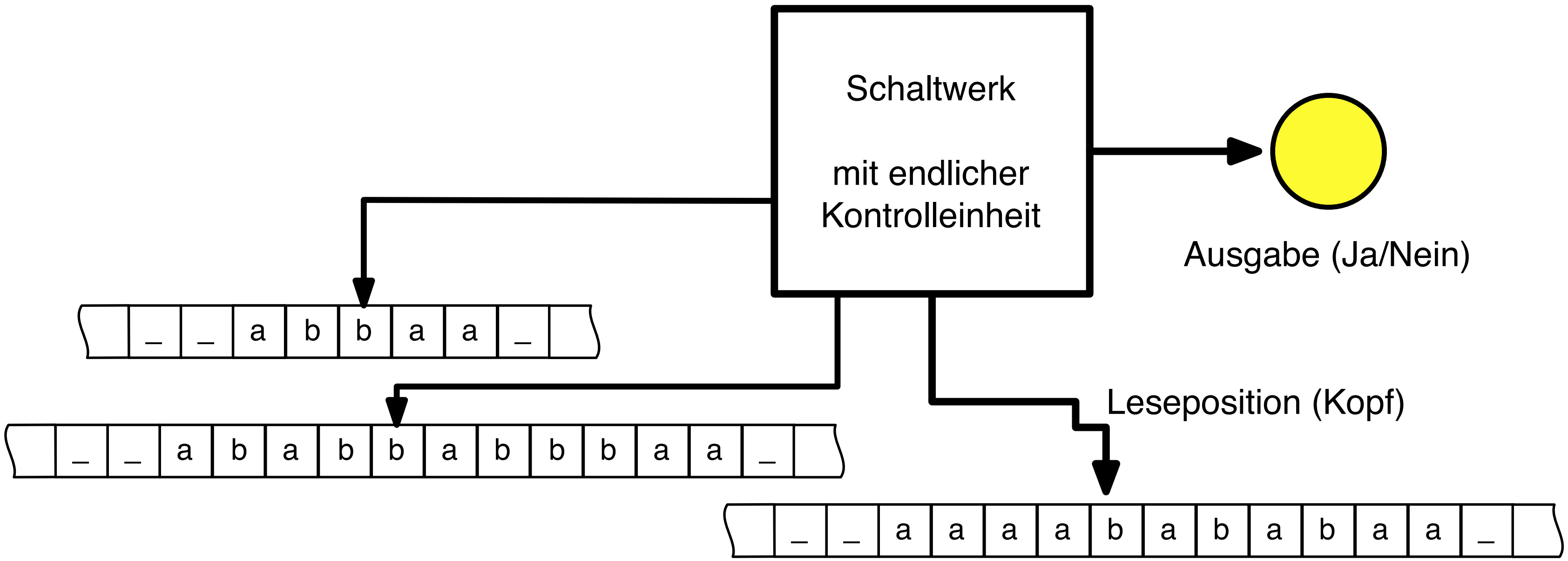
Mehrband-Turingmaschinen

Mehrband-Turingmaschinen

- Idee: TM hat k Bänder zur Verfügung

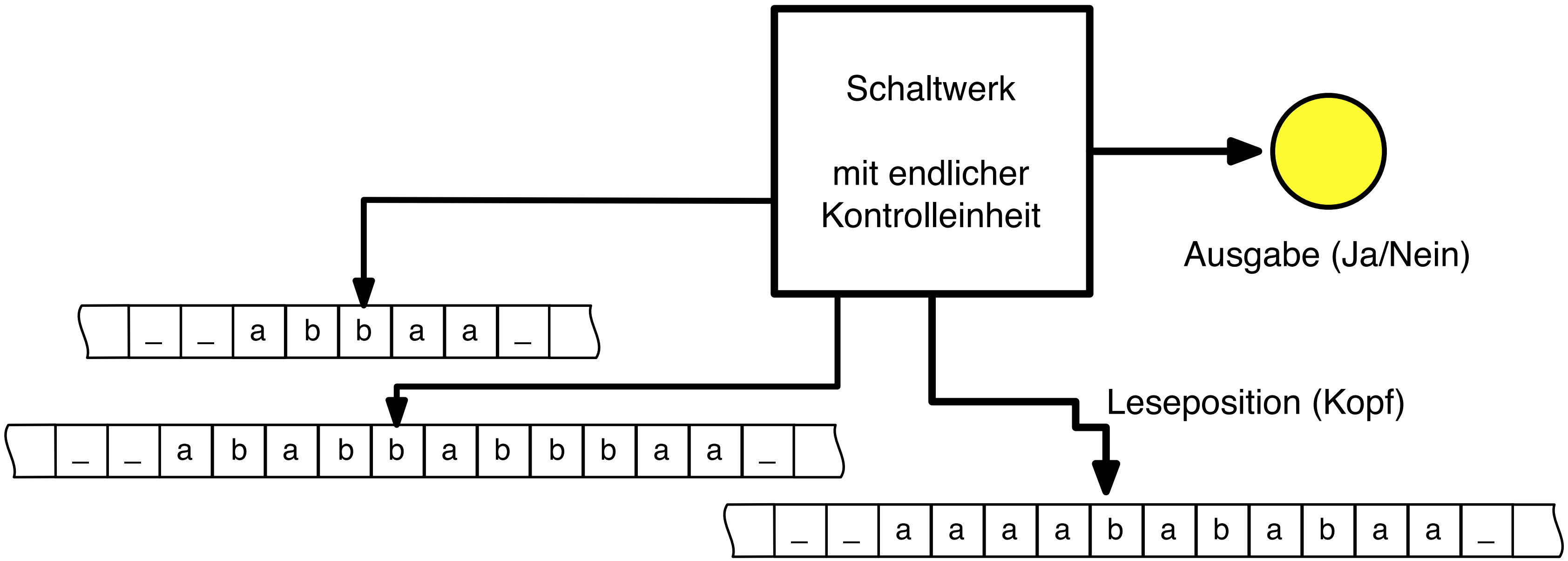
Mehrband-Turingmaschinen

- Idee: TM hat k Bänder zur Verfügung



Mehrband-Turingmaschinen

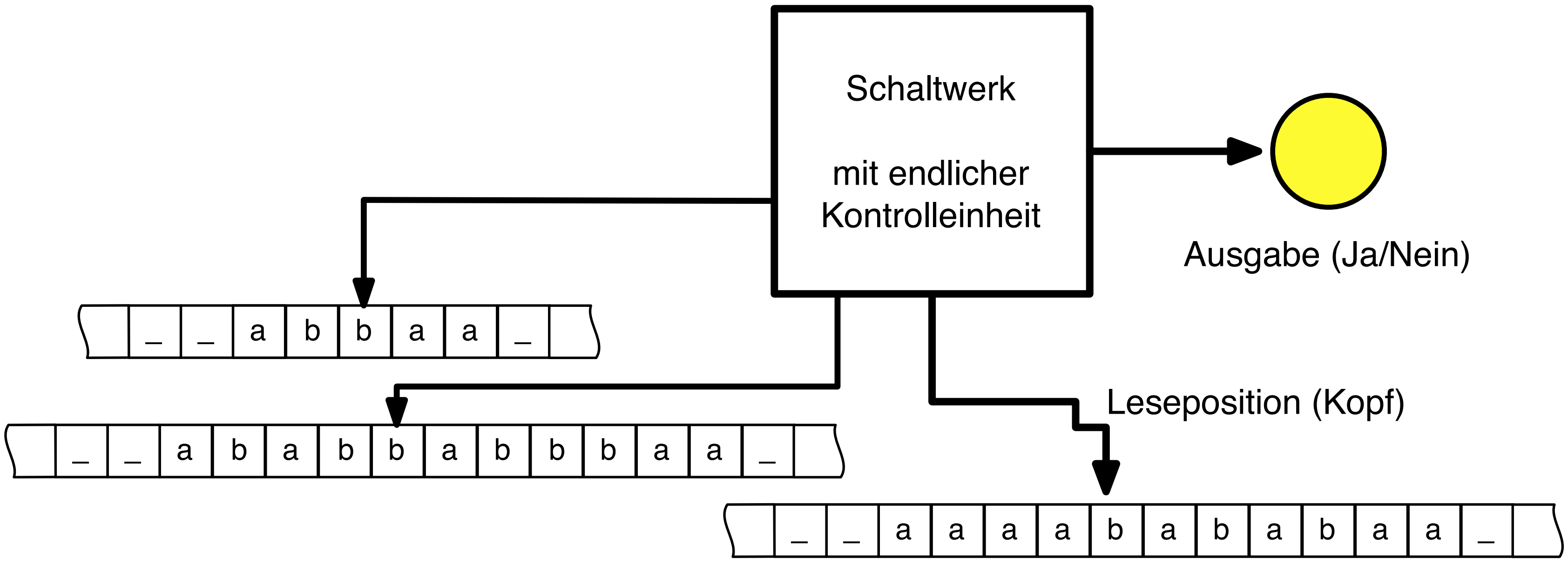
- Idee: TM hat k Bänder zur Verfügung



- Kopfpositionen und Bewegungen unabhängig

Mehrband-Turingmaschinen

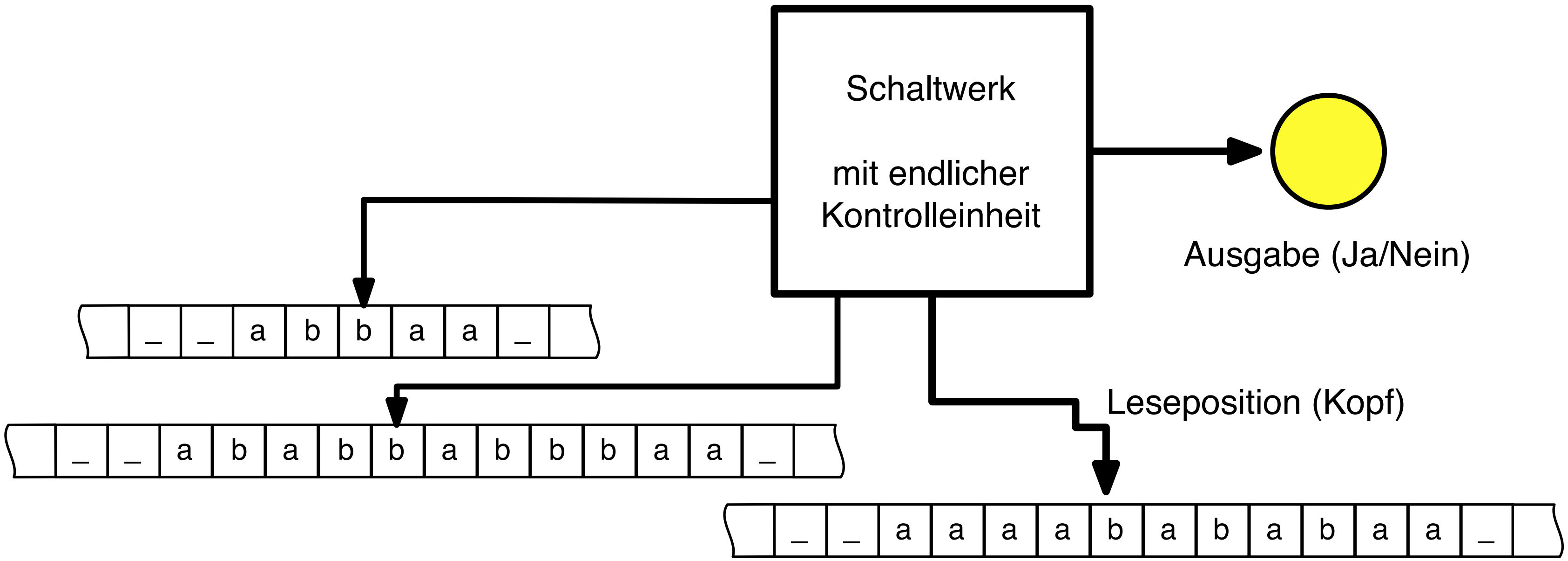
- Idee: TM hat k Bänder zur Verfügung



- Kopfpositionen und Bewegungen unabhängig
- Übergang hängt von allen Zeichen auf Kopfpositionen ab

Mehrband-Turingmaschinen

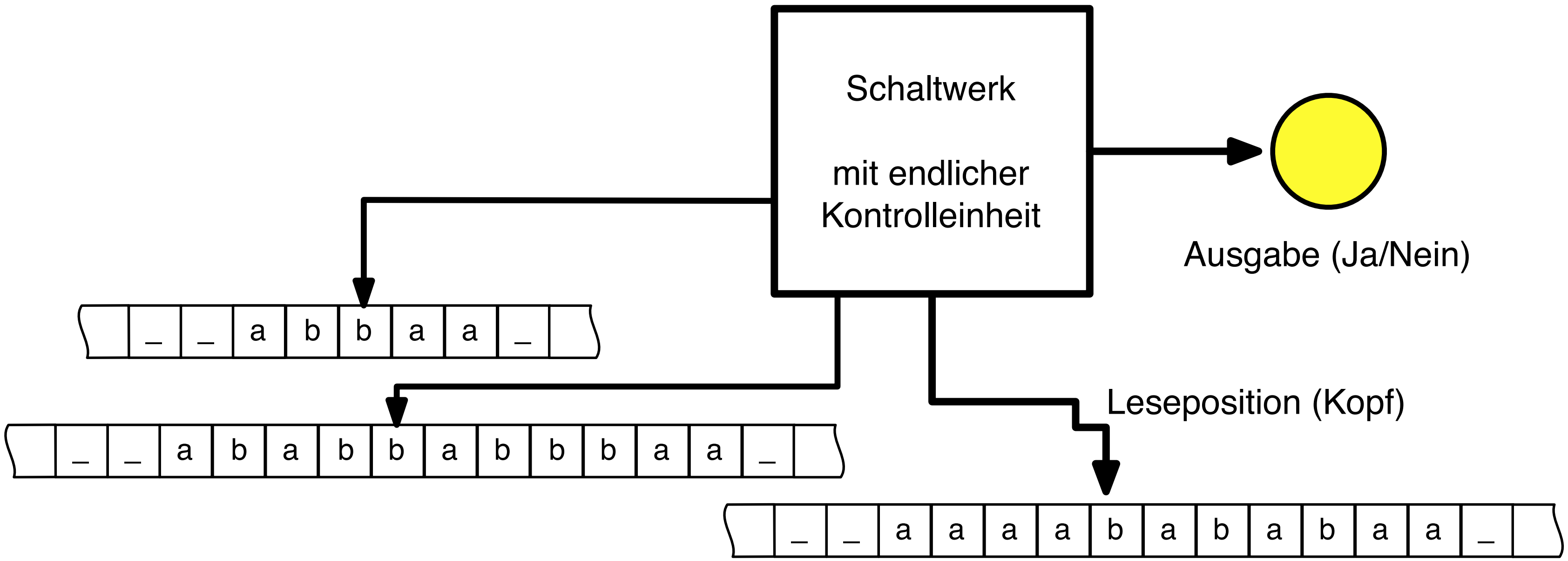
- Idee: TM hat k Bänder zur Verfügung



- Kopfpositionen und Bewegungen unabhängig
- Übergang hängt von allen Zeichen auf Kopfpositionen ab
  $\rightarrow \delta\colon Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$

Mehrband-Turingmaschinen

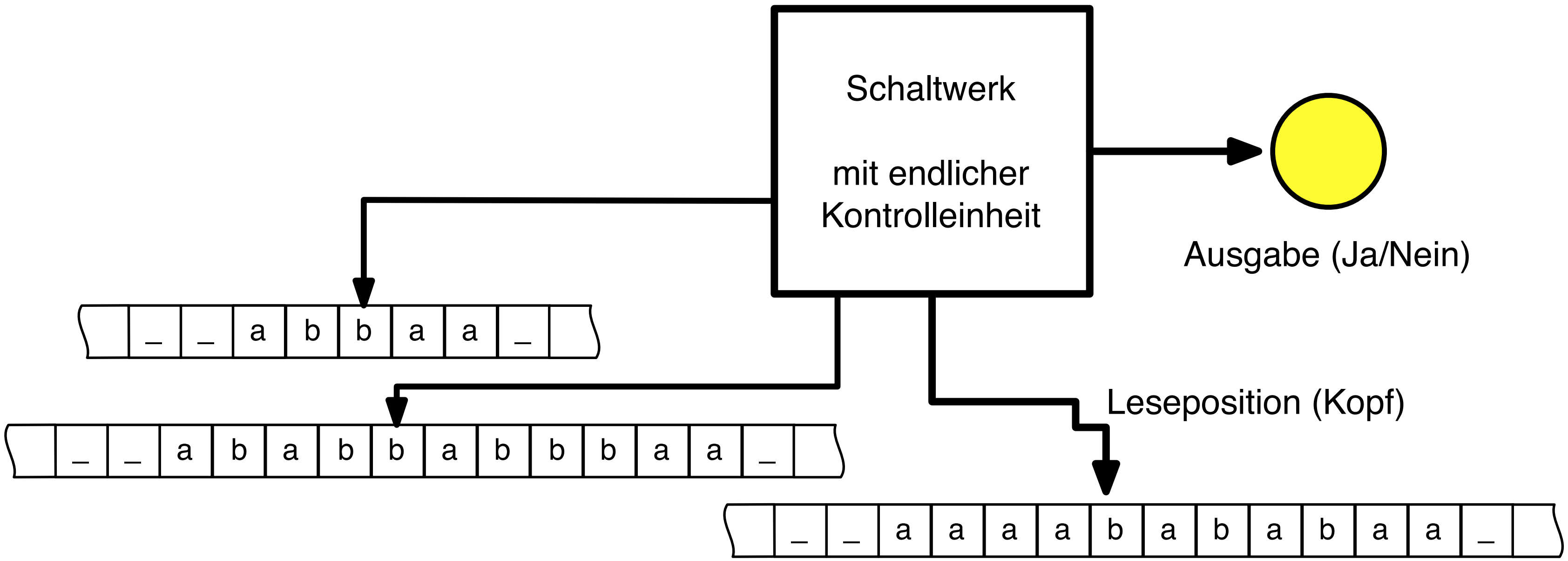
- Idee: TM hat k Bänder zur Verfügung



- Kopfpositionen und Bewegungen unabhängig
- Übergang hängt von allen Zeichen auf Kopfpositionen ab
  $\to \delta\colon Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$
- Bsp.: $\delta(q_i, (\mathsf{a}, \mathtt{x})) = (q_\ell, (\mathtt{y}, \mathsf{c}), (R, L))$

Mehrband-Turingmaschinen

- Idee: TM hat k Bänder zur Verfügung



- Kopfpositionen und Bewegungen unabhängig
- Übergang hängt von allen Zeichen auf Kopfpositionen ab
  $\to \delta\colon Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$
- Bsp.: $\delta(q_i, (\mathtt{a}, \mathtt{x})) = (q_\ell, (\mathtt{y}, \mathtt{c}), (R, L))$
- Initial: Eingabe steht auf 1. Band, sonst überall Blanks

Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*

Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k

Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k
- Spur i ist def. durch die i-ten Komponenten der Bandsymbole

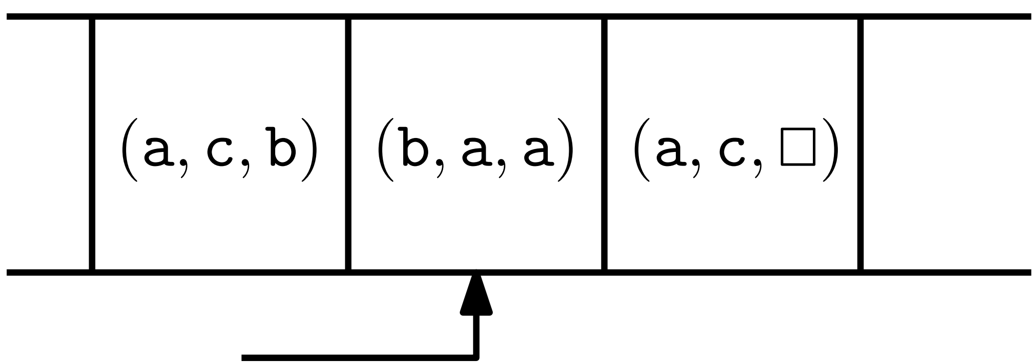
Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k
- Spur i ist def. durch die i-ten Komponenten der Bandsymbole

Bsp. :



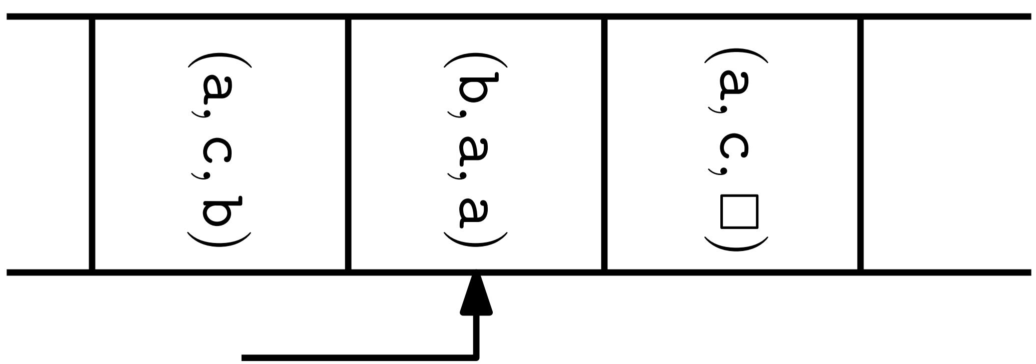
Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k
- Spur i ist def. durch die i-ten Komponenten der Bandsymbole

Bsp. :



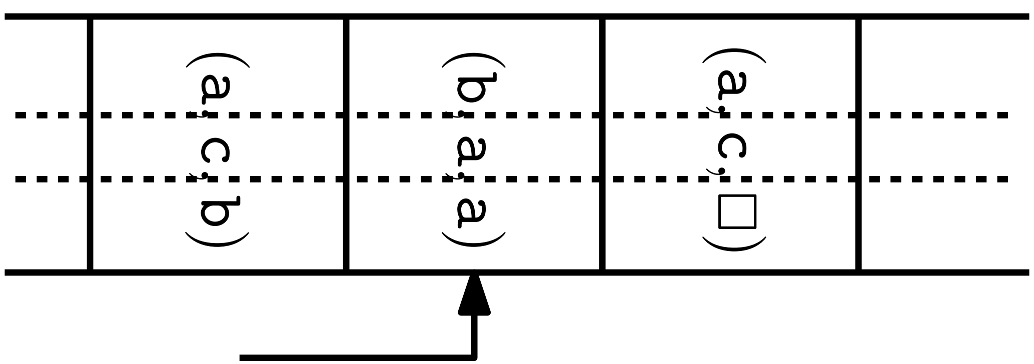
Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k
- Spur i ist def. durch die i-ten Komponenten der Bandsymbole

Bsp. :



1. Spur ..aba..
2. Spur ..cac..
3. Spur ..ba □ ..

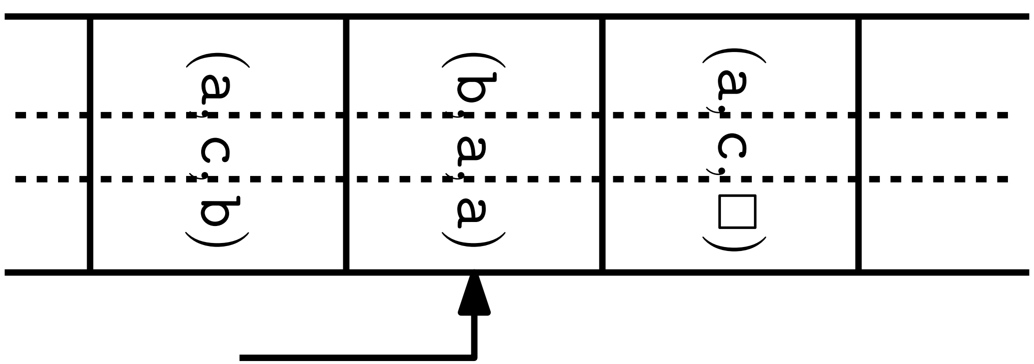
Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k
- Spur i ist def. durch die i-ten Komponenten der Bandsymbole

Bsp. :



1. Spur ..aba..
2. Spur ..cac..
3. Spur ..ba □ ..

- Ziel: Identifikation Spur = Band

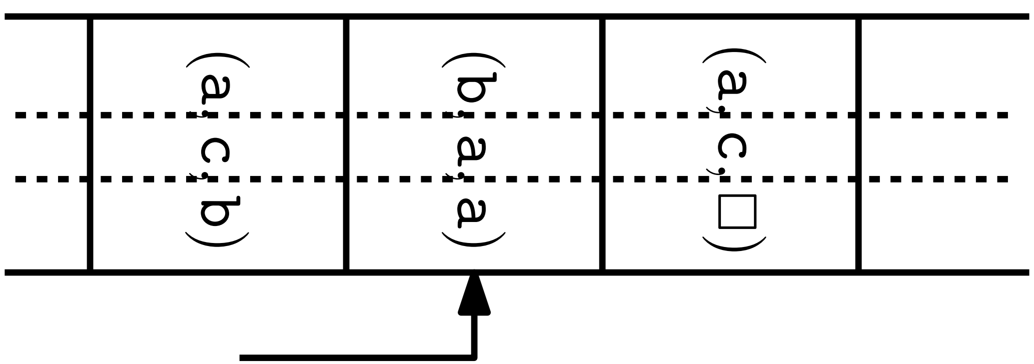
Satz 15

Jede Mehrband TM kann durch eine herkömmliche TM simuliert werden.

Simulation durch Mehrspurtechnik

- **Idee:** Unterteile das Band durch Erweiterung des Arbeitsalphabetes in *Spuren*
- neues Arbeitsalphabet: Γ^k
- Spur i ist def. durch die i-ten Komponenten der Bandsymbole

Bsp. :



1. Spur ..aba..
2. Spur ..cac..
3. Spur ..ba □ ..

- Ziel: Identifikation Spur = Band
- Wie kann ich unabhängige Kopfbewegungen simulieren?

Simulation der Kopfbewegungen

Simulation der Kopfbewegungen

- Extraband mit Markierung für Kopfposition

Simulation der Kopfbewegungen

- Extraband mit Markierung für Kopfposition

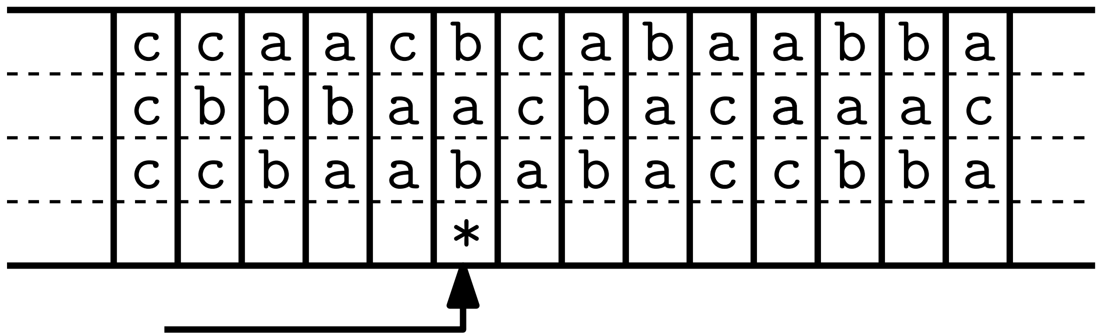
Bsp.:

c	c	a	a	c	b	c	a	b	a	a	b	b	a
c	b	b	b	a	a	c	b	a	c	a	a	a	c
c	c	b	a	a	b	a	b	a	c	c	b	b	a
					*								

Simulation der Kopfbewegungen

- Extraband mit Markierung für Kopfposition

Bsp.:



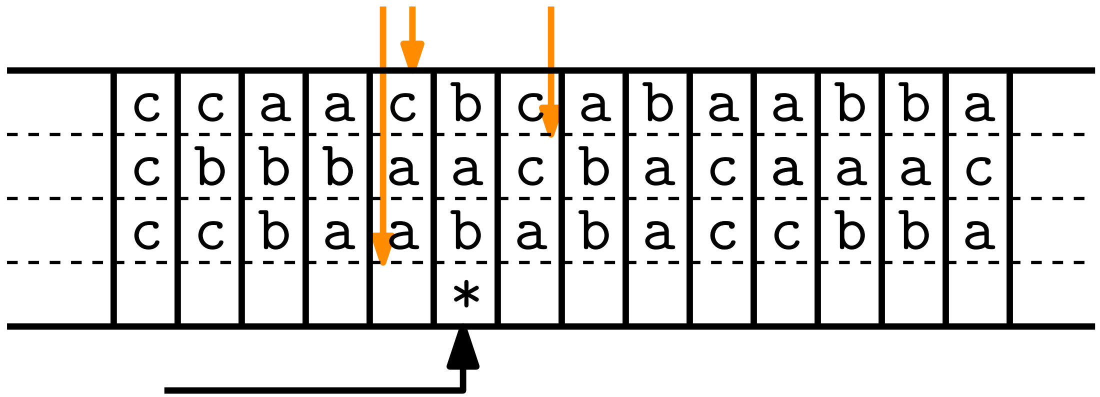
Kopfbewegung

(L, R, L)

Simulation der Kopfbewegungen

- Extraband mit Markierung für Kopfposition

Bsp.:



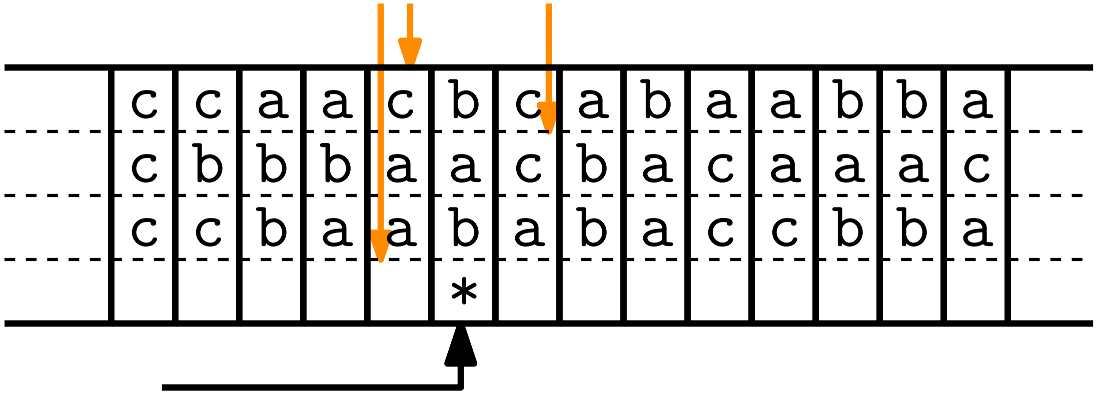
Kopfbewegung

(L, R, L)

Simulation der Kopfbewegungen

- Extraband mit Markierung für Kopfposition

Bsp.:



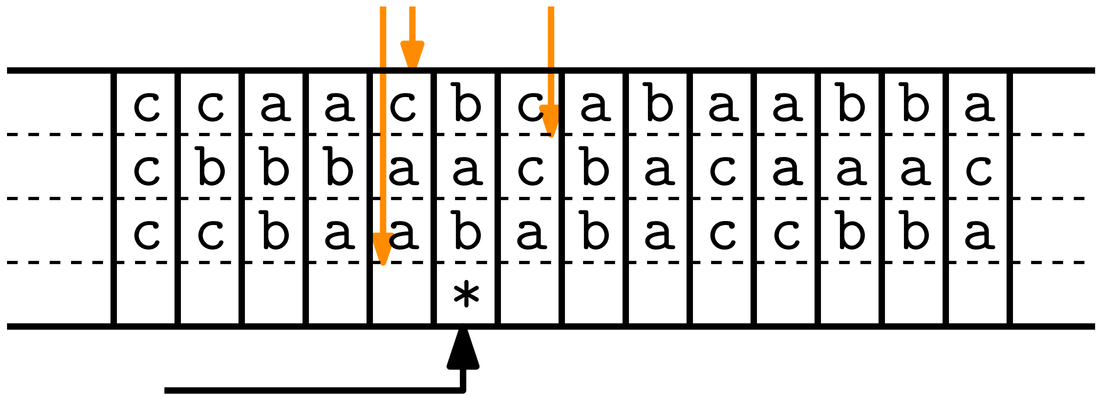
Kopfbewegung
(L, R, L)

- Verschieben der Spuren zur Ausrichtung der Kopfpositionen an $*$

Simulation der Kopfbewegungen

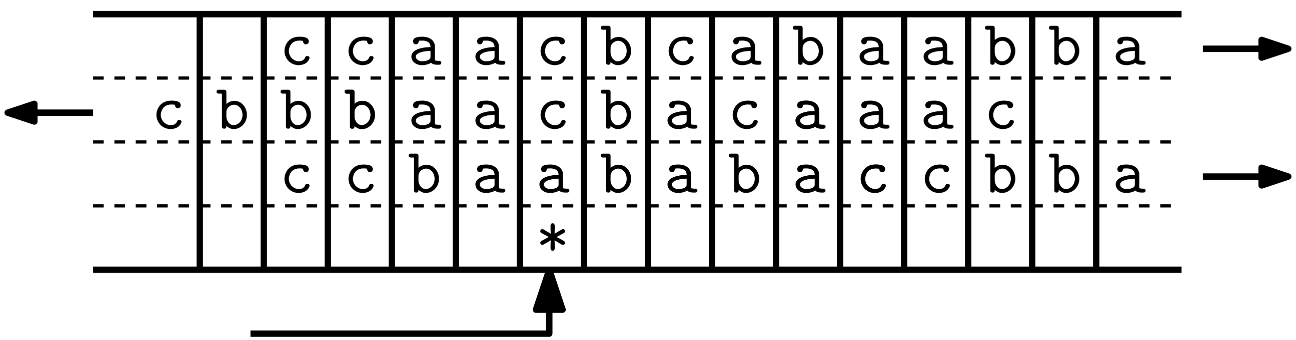
- Extraband mit Markierung für Kopfposition

Bsp.:



Kopfbewegung
(L, R, L)

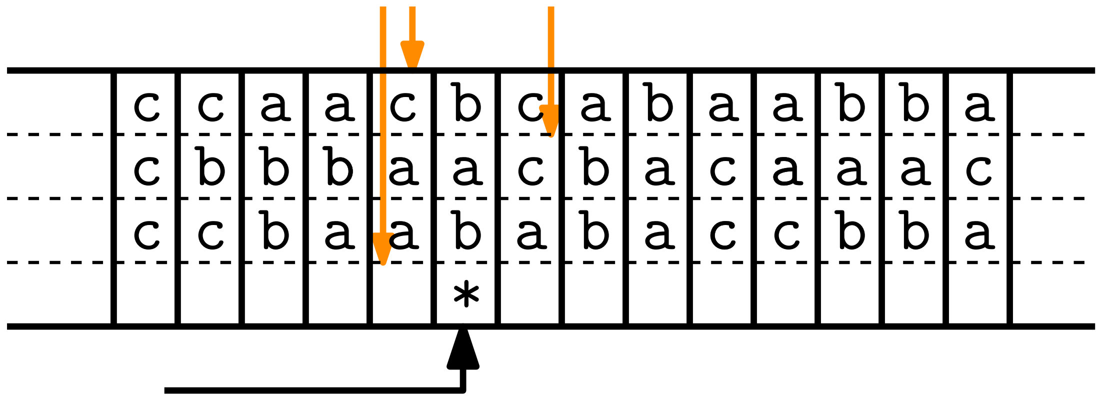
- Verschieben der Spuren zur Ausrichtung der Kopfpositionen an $*$



Simulation der Kopfbewegungen

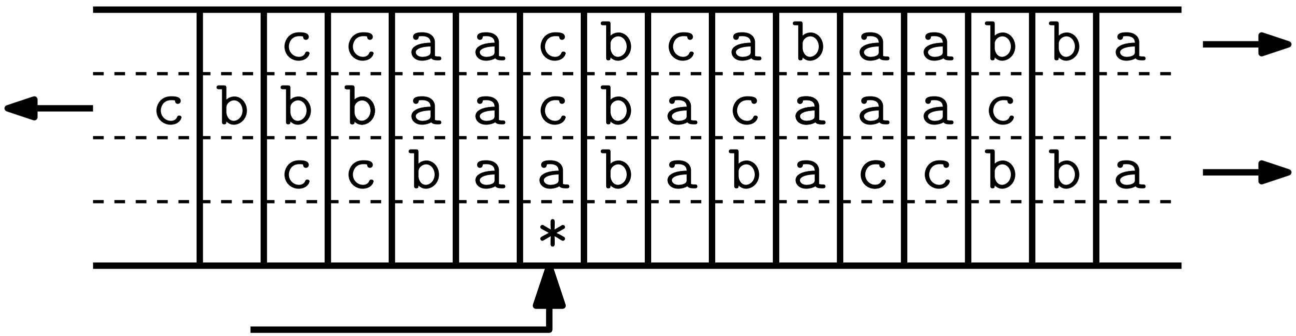
- Extraband mit Markierung für Kopfposition

Bsp.:



Kopfbewegung
(L, R, L)

- Verschieben der Spuren zur Ausrichtung der Kopfpositionen an $*$

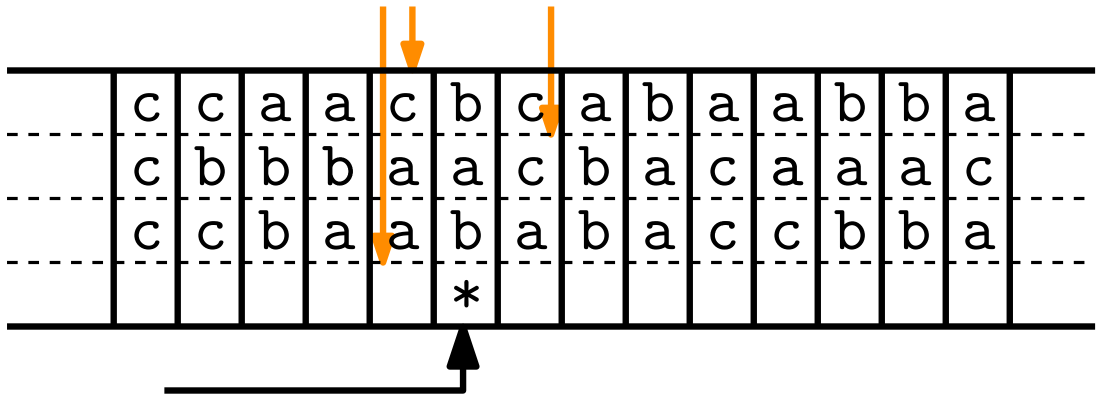


- Erweitern des ursprünglichen Befehls mit k Spur-Verschiebungen

Simulation der Kopfbewegungen

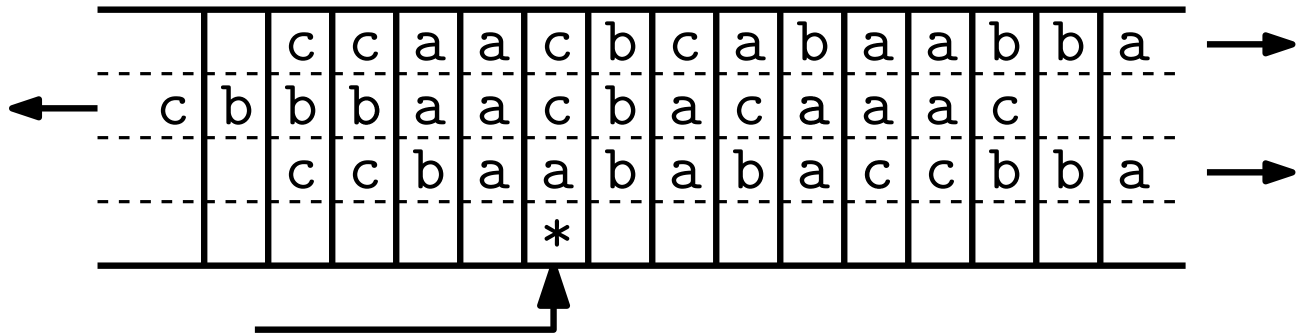
- Extraband mit Markierung für Kopfposition

Bsp.:



Kopfbewegung
(L, R, L)

- Verschieben der Spuren zur Ausrichtung der Kopfpositionen an $*$



- Erweitern des ursprünglichen Befehls mit k Spur-Verschiebungen
- Spur Verschiebung als Modul

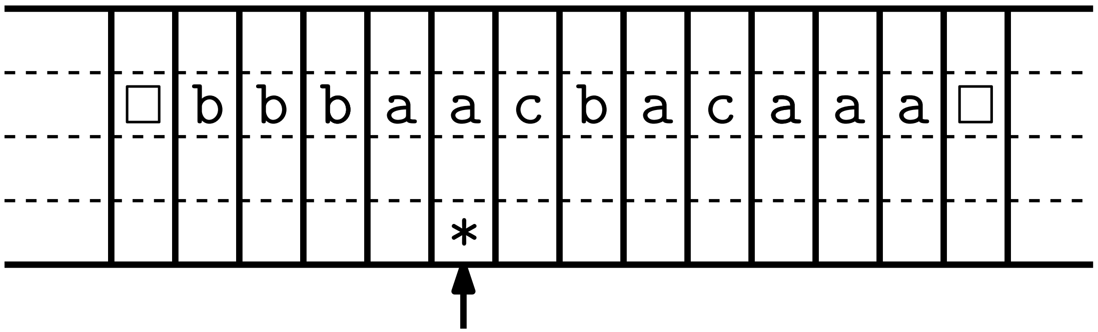
Modul: Spur-i-nach-links

Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k+1$

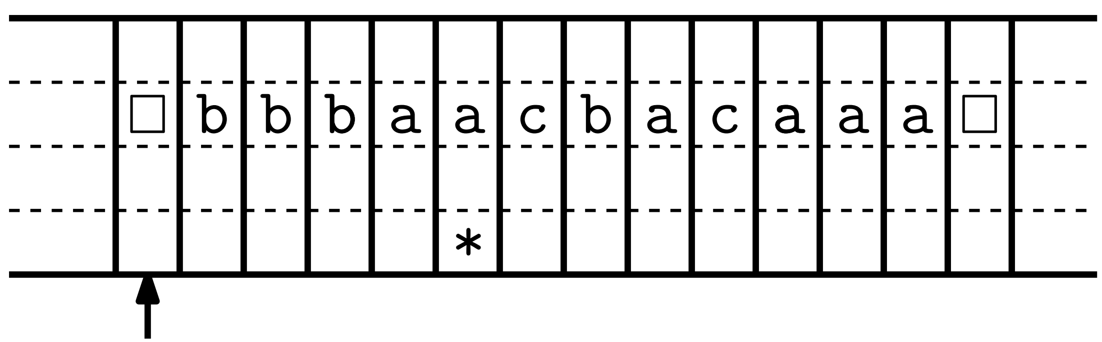
Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k + 1$



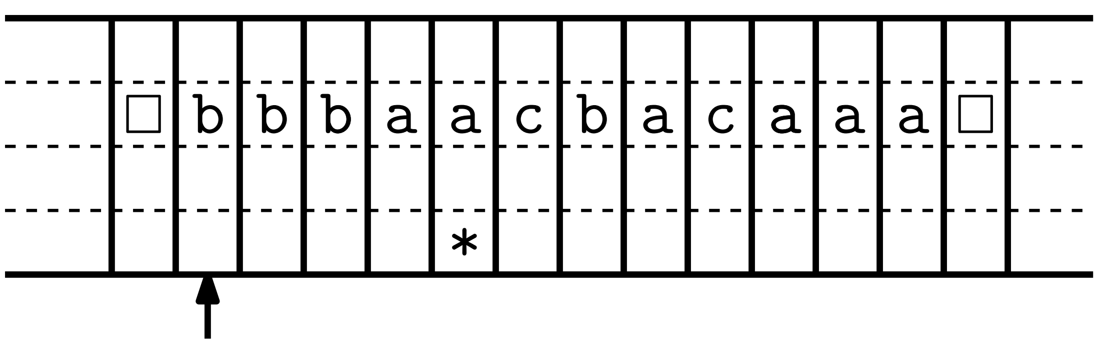
Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k+1$



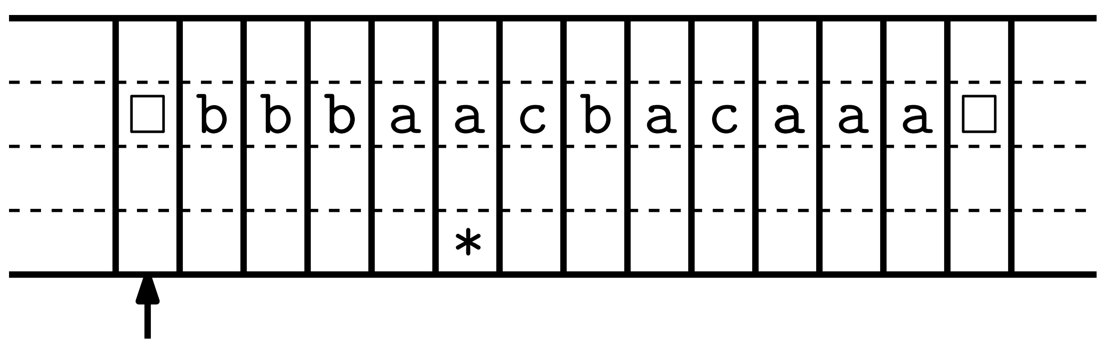
Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k + 1$



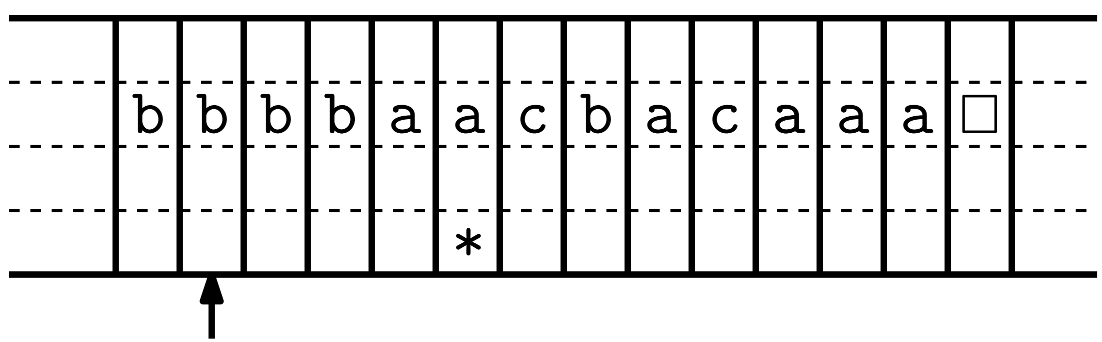
Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k + 1$



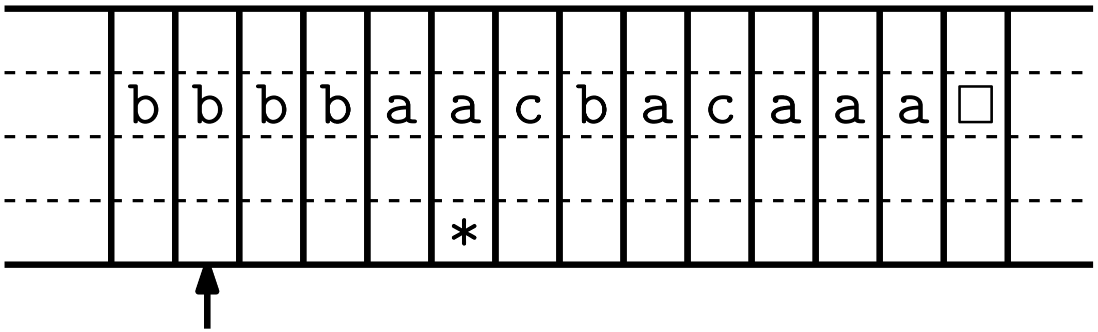
Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k+1$



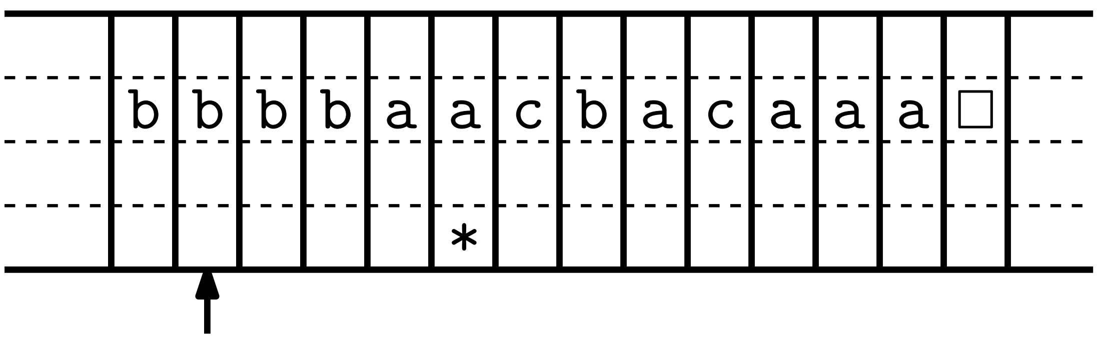
Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k+1$



Modul: Spur-i-nach-links

1. Gehe zum ersten Blank nach links
2. Gehe einen Schritt nach rechts
3. Merke das aktuelle Zeichen im Zustand und gehe nach links
4. Schreibe das gemerkte Zeichen aufs Band und gehe nach rechts
5. Stehe ich auf keinem Blank wiedehole ab 2.
6. Bewege Kopf zur markierten Position auf Spur $k + 1$



- andere Spuren und Richtungen analog

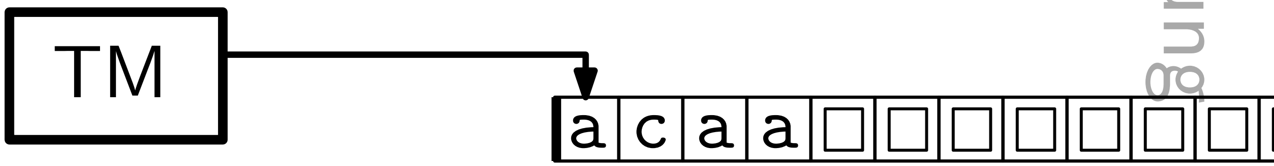
Halbband-Turingmaschinen

Halbband-Turingmaschinen

- Statt eines beidseitig unbeschränkten Bandes jetzt nur noch ein einseitig unbeschränktes Band

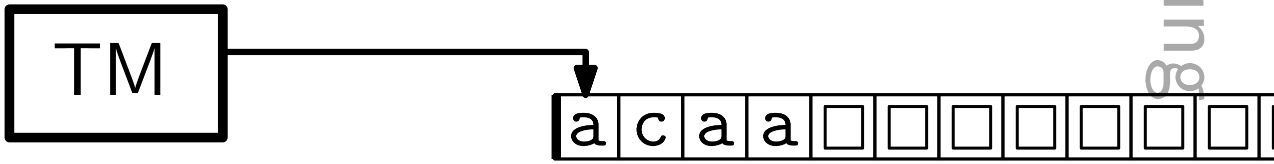
Halbband-Turingmaschinen

- Statt eines beidseitig unbeschränkten Bandes jetzt nur noch ein einseitig unbeschränktes Band
- Startkonfiguration: Kopf auf Zelle 0, Eingabe beginnt ab Kopfposition



Halbband-Turingmaschinen

- Statt eines beidseitig unbeschränkten Bandes jetzt nur noch ein einseitig unbeschränktes Band
- Startkonfiguration: Kopf auf Zelle 0, Eingabe beginnt ab Kopfposition

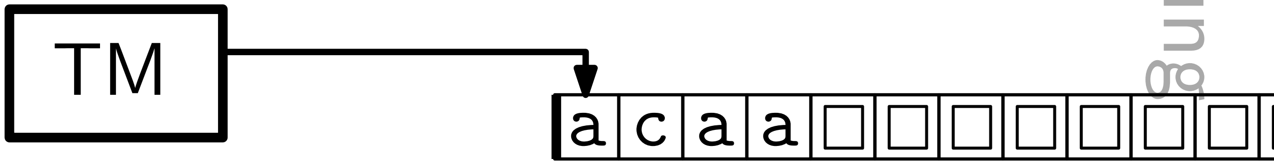


Satz 16

Jede TM kann durch eine Halbband-TM simuliert werden.

Halbband-Turingmaschinen

- Statt eines beidseitig unbeschränkten Bandes jetzt nur noch ein einseitig unbeschränktes Band
- Startkonfiguration: Kopf auf Zelle 0, Eingabe beginnt ab Kopfposition



Satz 16

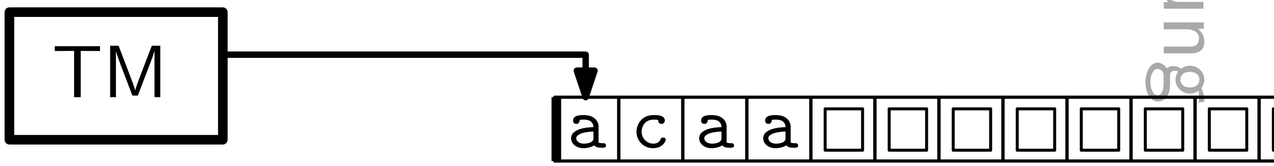
Jede TM kann durch eine Halbband-TM simuliert werden.

Beweisidee

Band der TM wir durch zwei Spuren auf dem Halbband simuliert

Halbband-Turingmaschinen

- Statt eines beidseitig unbeschränkten Bandes jetzt nur noch ein einseitig unbeschränktes Band
- Startkonfiguration: Kopf auf Zelle 0, Eingabe beginnt ab Kopfposition

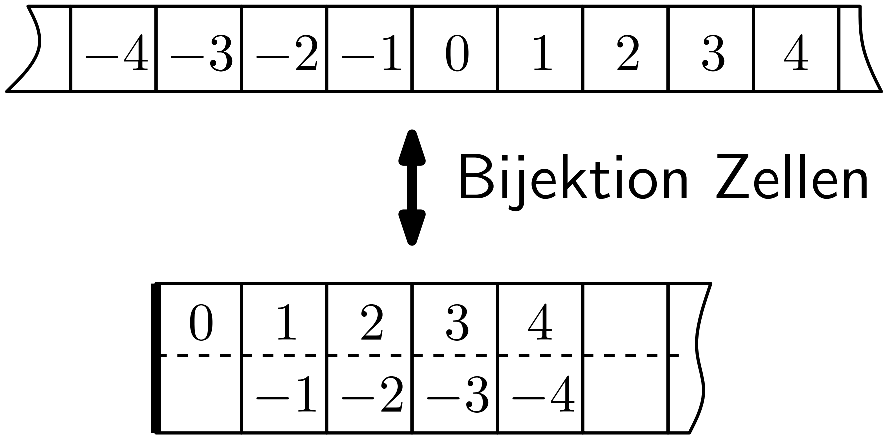


Satz 16

Jede TM kann durch eine Halbband-TM simuliert werden.

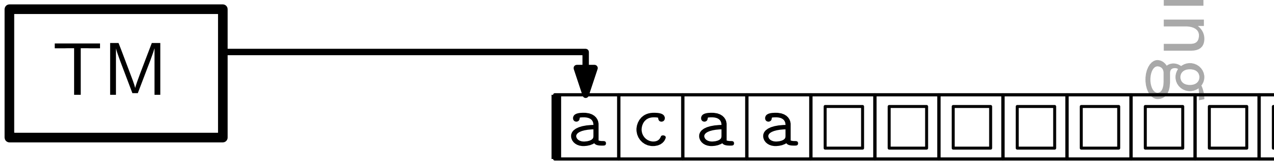
Beweisidee

Band der TM wir durch zwei Spuren auf dem Halbband simuliert



Halbband-Turingmaschinen

- Statt eines beidseitig unbeschränkten Bandes jetzt nur noch ein einseitig unbeschränktes Band
- Startkonfiguration: Kopf auf Zelle 0, Eingabe beginnt ab Kopfposition



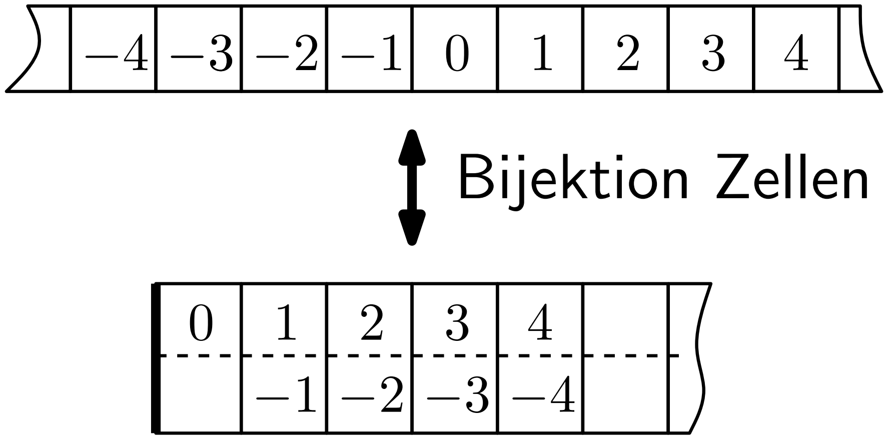
Satz 16

Jede TM kann durch eine Halbband-TM simuliert werden.

Beweisidee

Band der TM wir durch zwei Spuren auf dem Halbband simuliert

aktuelle Spur wird sich im Zustand "gemerkt"



Nichtdeterministische TM (NTM)

Nichtdeterministische TM (NTM)

- TM definiert wie bisher, bis auf
  $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$

Nichtdeterministische TM (NTM)

- TM definiert wie bisher, bis auf
  $\delta \colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet

Nichtdeterministische TM (NTM)

- TM definiert wie bisher, bis auf
  $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet
- Berechnungsbaum nicht notwendiger Weise endlich

Nichtdeterministische TM (NTM)

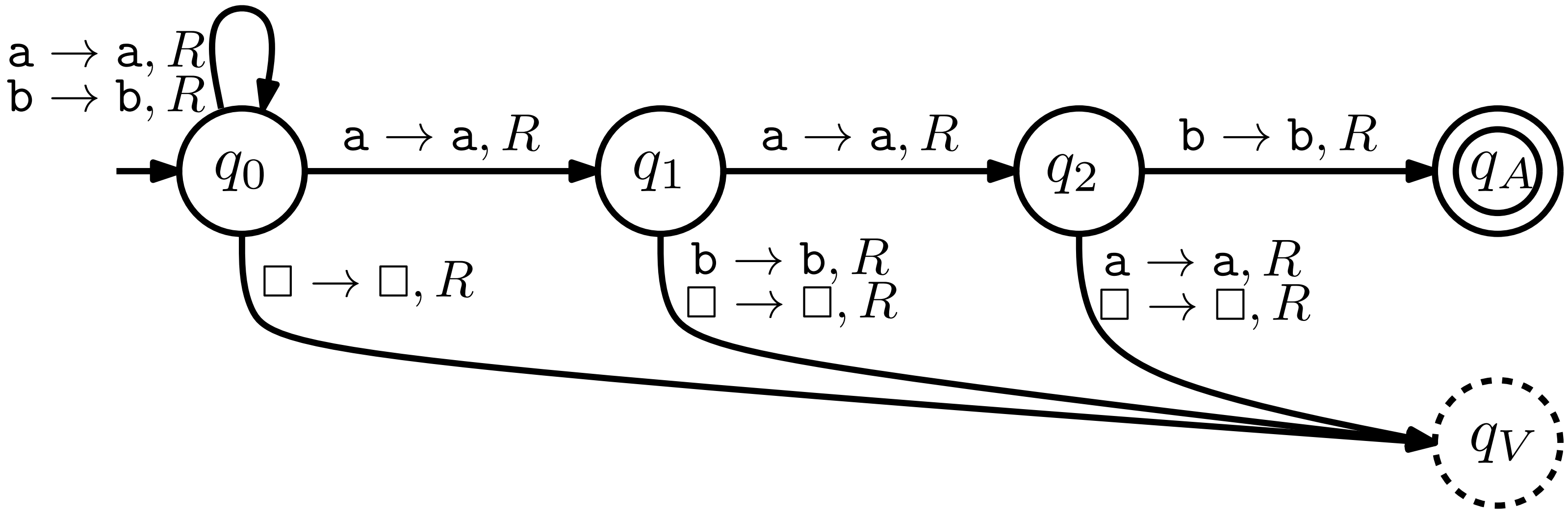
- TM definiert wie bisher, bis auf $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet
- Berechnungsbaum nicht notwendiger Weise endlich
- Berechnungsbaum hängt von TM **und** Eingabe ab

Nichtdeterministische TM (NTM)

- TM definiert wie bisher, bis auf
  $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet
- Berechnungsbaum nicht notwendiger Weise endlich
- Berechnungsbaum hängt von TM **und** Eingabe ab
- Lauf$=$Wurzel-Blatt Pfad im Berechnungsbaum
  bei (det.) TM sprechen wir von einem Berechnungspfad

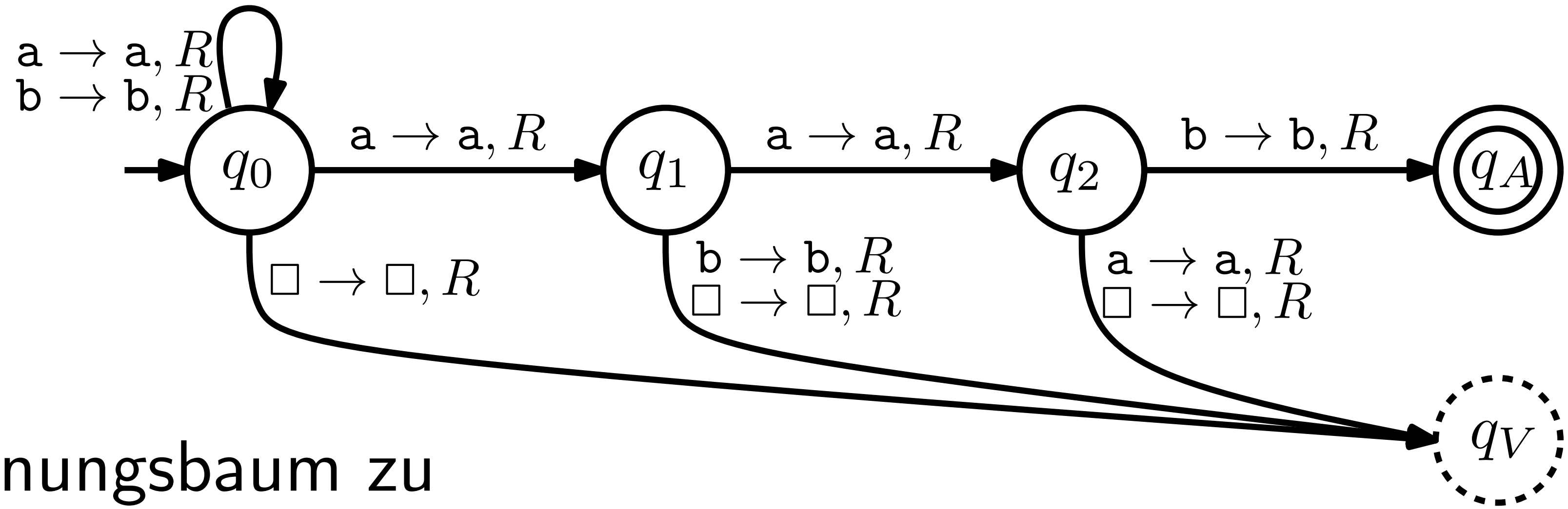
Berechnungsbaum am Beispiel

TM T

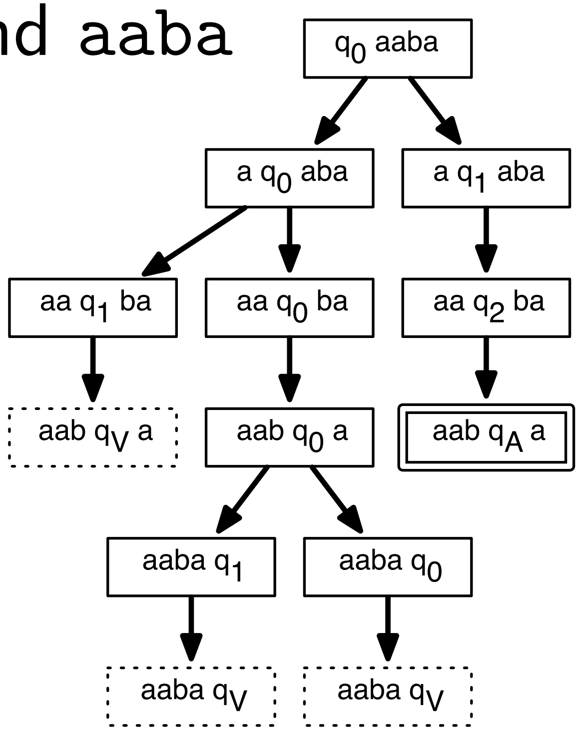


Berechnungsbaum am Beispiel

TM T

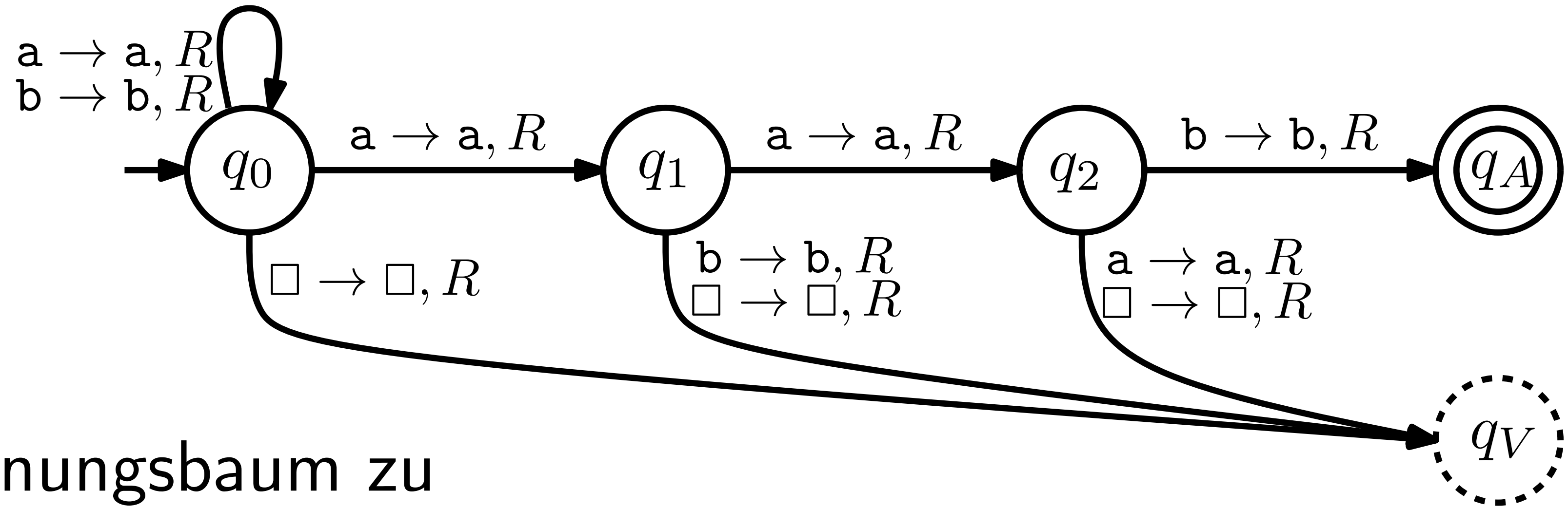


Berechnungsbaum zu T und aaba

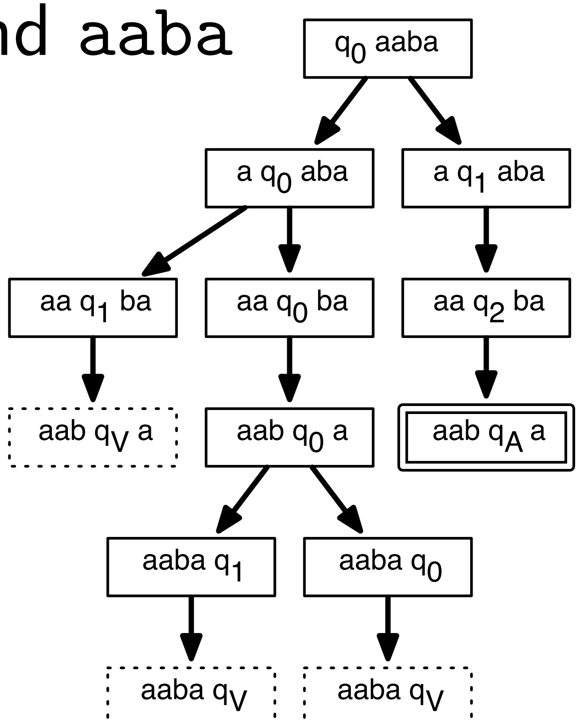


Berechnungsbaum am Beispiel

TM T



Berechnungsbaum zu T und aaba

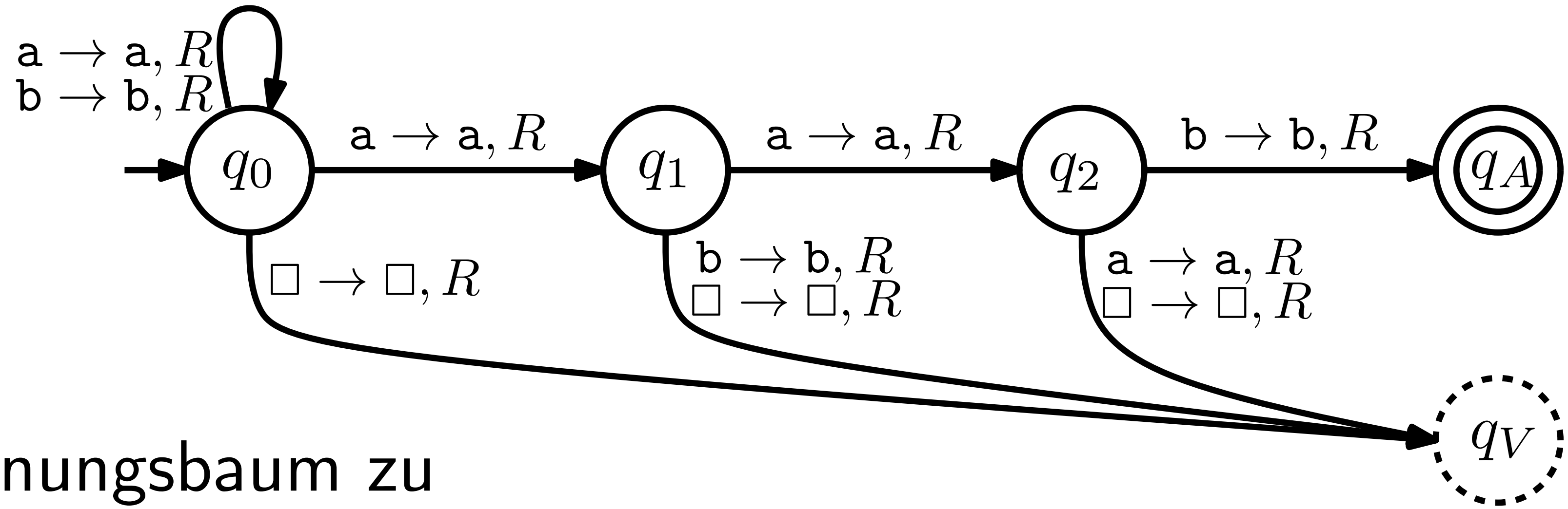


Definition

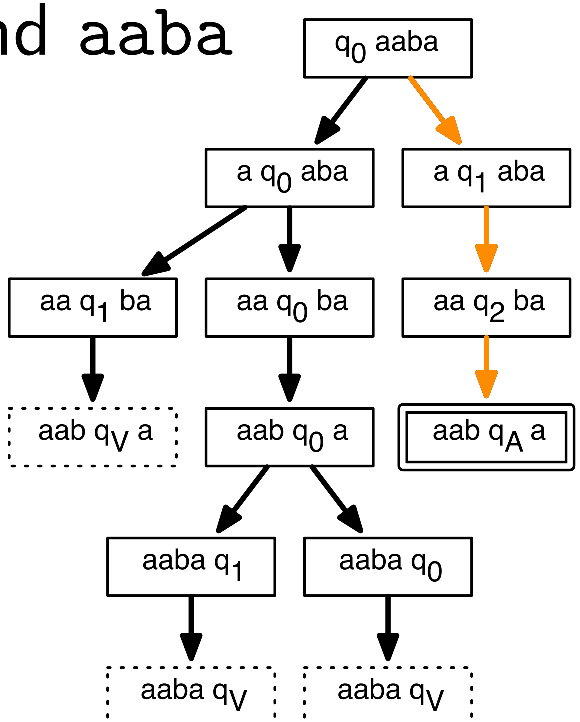
Ein Wort w wird von der NTM T akzeptiert, gdw. im Berechnungsbaum zu T und w es eine akzeptierende Konfiguration gibt.

Berechnungsbaum am Beispiel

TM T



Berechnungsbaum zu T und aaba

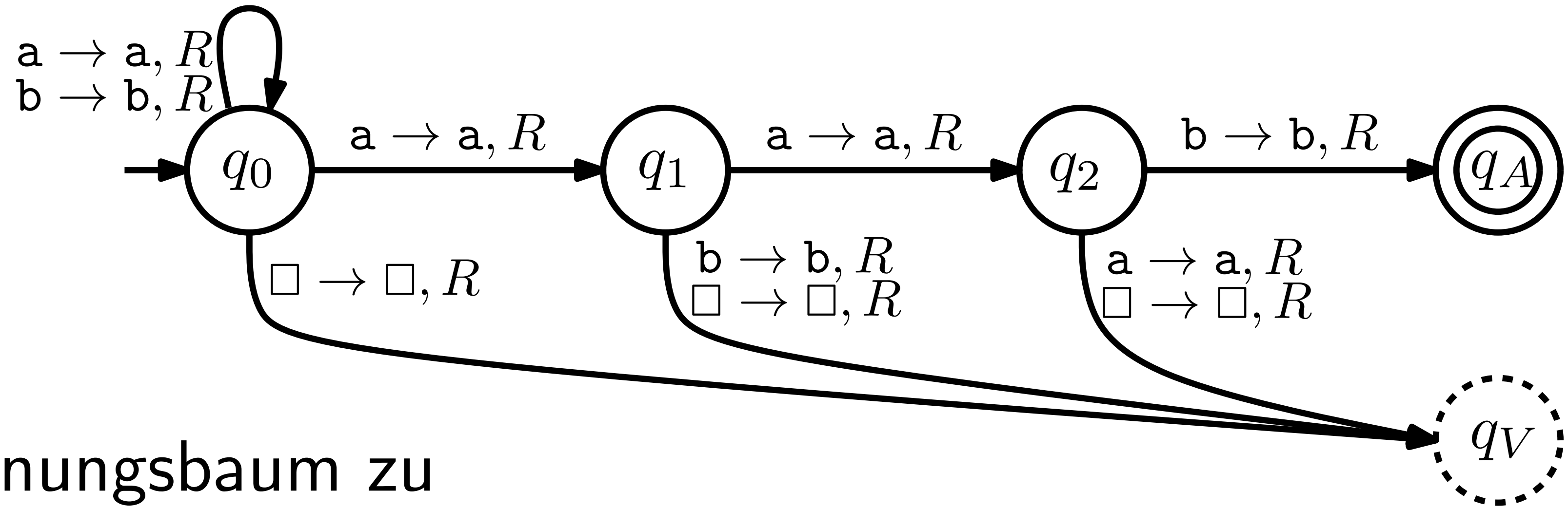


Definition

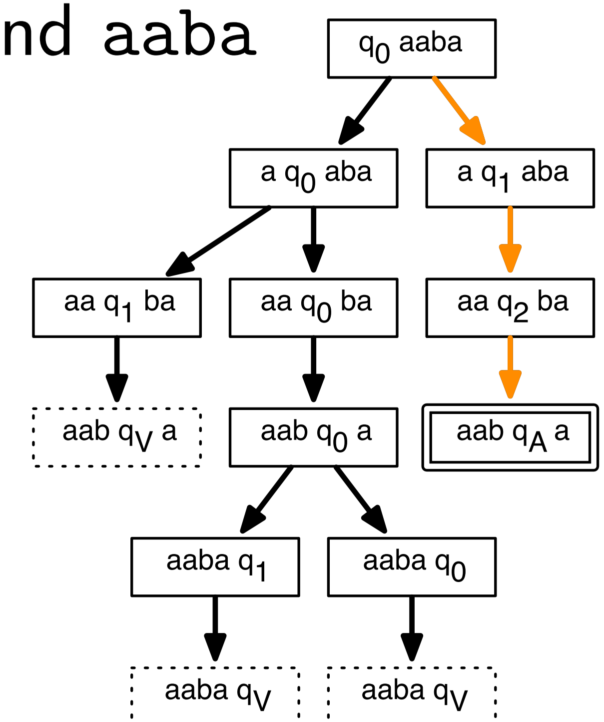
Ein Wort w wird von der NTM T akzeptiert, gdw. im Berechnungsbaum zu T und w es eine akzeptierende Konfiguration gibt.

Berechnungsbaum am Beispiel

TM T



Berechnungsbaum zu T und aaba



Definition

Ein Wort w wird von der NTM T akzeptiert, gdw. im Berechnungsbaum zu T und w es eine akzeptierende Konfiguration gibt.

$\rightarrow$ aaba wird akzeptiert

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

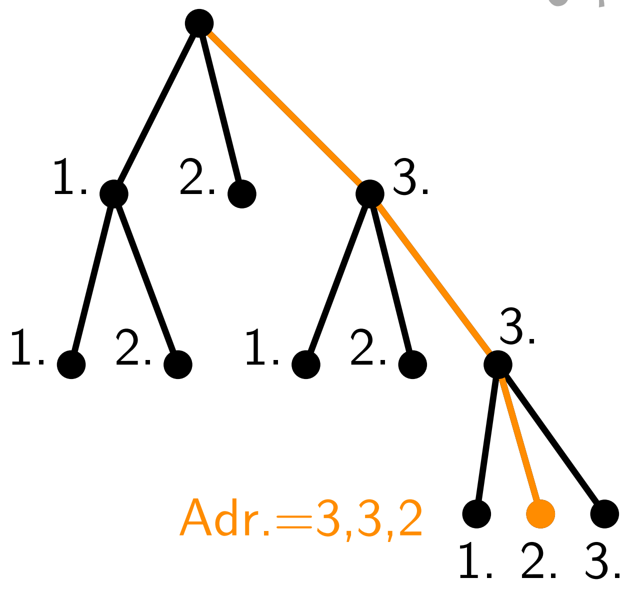
- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad

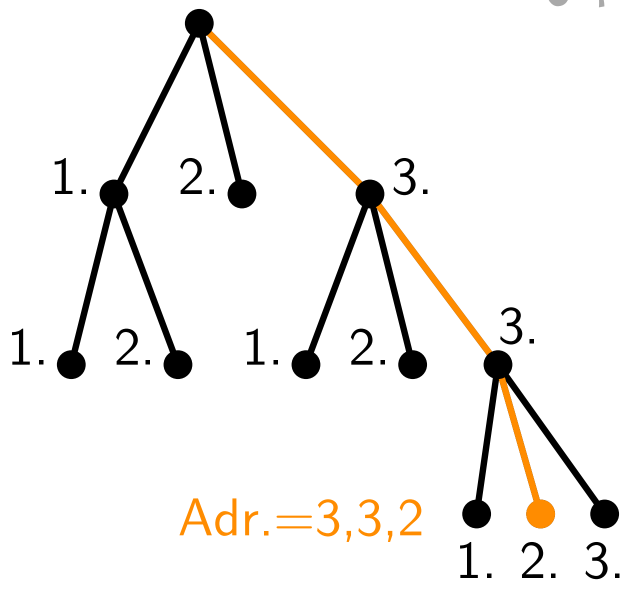


Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)

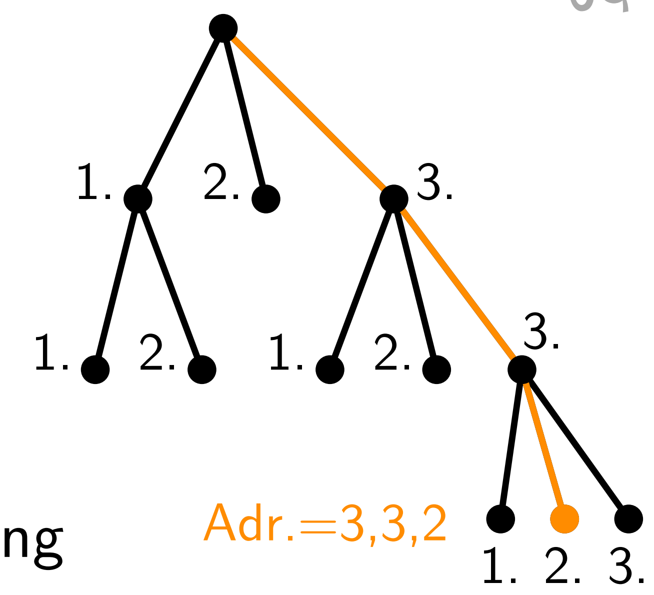


Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)
- Simuliere auf einem anderem Band den Übergang zur Konfiguration der angegebenen Adresse

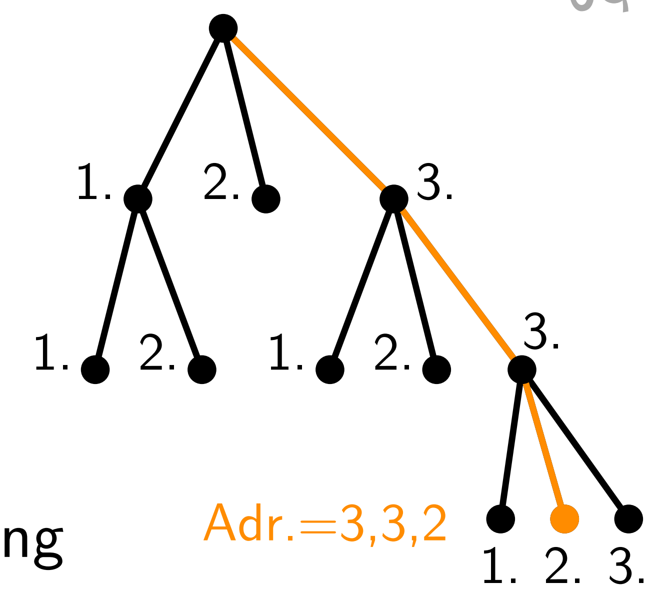


Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)
- Simuliere auf einem anderem Band den Übergang zur Konfiguration der angegebenen Adresse



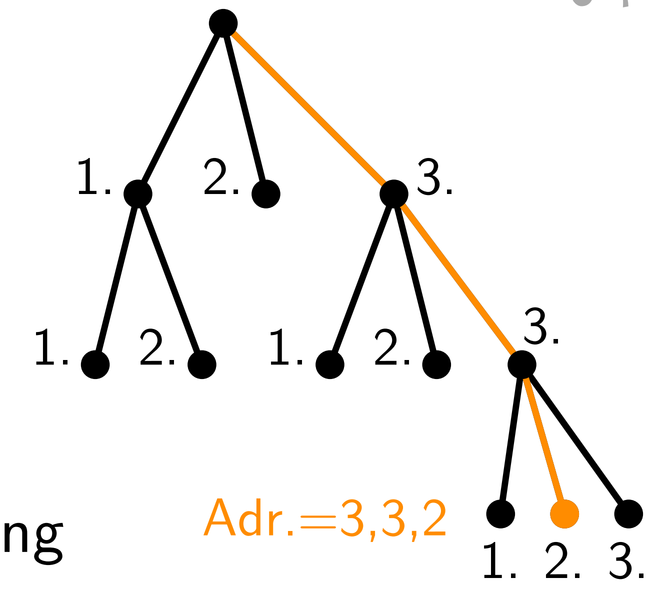
- Akzeptiere, wenn die berechnete Konf. akzeptierend, sonst weiter mit nächster Adresse

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)
- Simuliere auf einem anderem Band den Übergang zur Konfiguration der angegebenen Adresse



- Akzeptiere, wenn die berechnete Konf. akzeptierend, sonst weiter mit nächster Adresse

$\square$

TM mit Ausgabe

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschriebbar

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschriebbar
- Kopf des Ausgabebandes bewegt sich nur nach einem Schreibevorgang einen Schritt nach rechts, bleibt ansonsten stehen

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschreibbar
- Kopf des Ausgabebandes bewegt sich nur nach einem Schreibevorgang einen Schritt nach rechts, bleibt ansonsten stehen
- M beschreibt folgende Funktion:

$$\boxed{\begin{aligned} f_M(x) := y, \ & \text{mit } M(\langle x\rangle) \text{ stoppt akzeptierend} \\ & \text{und auf dem Ausgabeband steht } \langle y\rangle \end{aligned}}$$

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschreibbar
- Kopf des Ausgabebandes bewegt sich nur nach einem Schreibevorgang einen Schritt nach rechts, bleibt ansonsten stehen
- M beschreibt folgende Funktion:

$$\boxed{\begin{aligned} f_M(x) := y, \; & \text{ mit } M(\langle x \rangle) \text{ stoppt akzeptierend} \\ & \text{und auf dem Ausgabeband steht } \langle y \rangle \end{aligned}}$$

- verwirft M oder stoppt M nicht, dann ist die Funktion f_M für diese Eingabe undefiniert

Berechenbare Funktionen

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f : \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathtt{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f\colon \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathtt{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)
- Rechtseindeutige Relationen heißen auch **partielle** Funktionen.

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f\colon \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathtt{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)
- Rechtseindeutige Relationen heißen auch **partielle** Funktionen.
- In diesem Sinne beschreibt jede TM eine partielle Funktion.

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f \colon \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathtt{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)
- Rechtseindeutige Relationen heißen auch **partielle** Funktionen.
- In diesem Sinne beschreibt jede TM eine partielle Funktion.
- Um Hervorzuheben, dass eine partielle Funktion überall definiert ist, nennen wir solche Funktionen **total**.

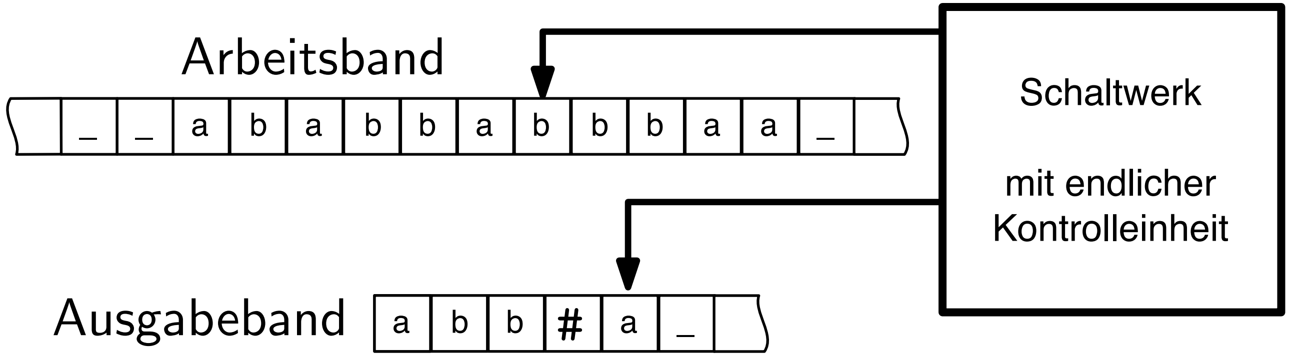
TM als Aufzähler

TM als Aufzähler

- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband

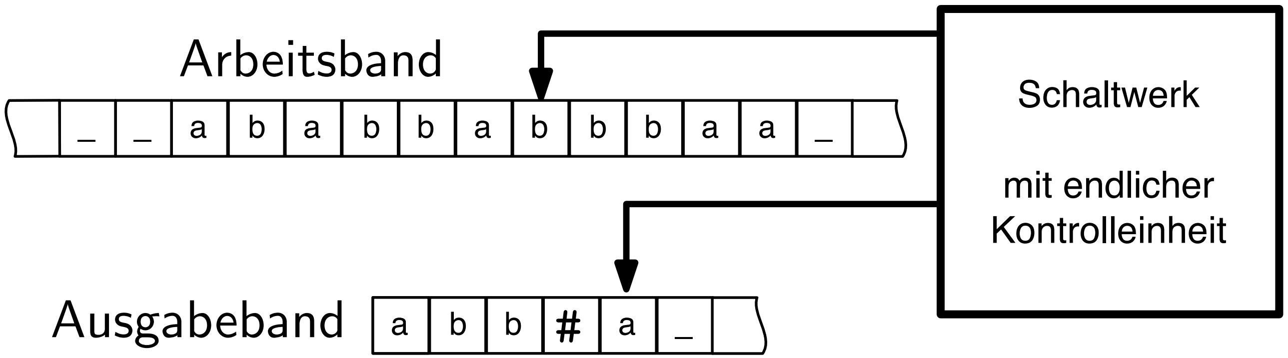
TM als Aufzähler

- ein **Aufzähler für** $L \subseteq \Sigma^*$ ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



TM als Aufzähler

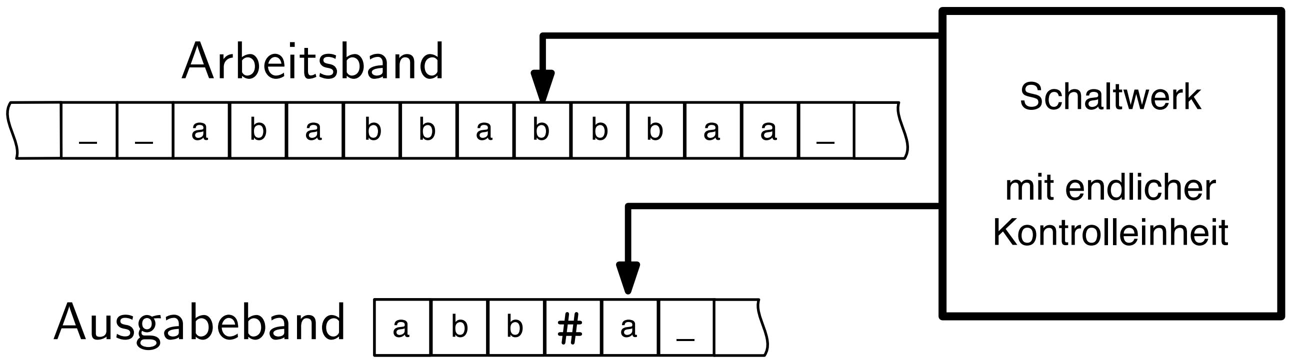
- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen

TM als Aufzähler

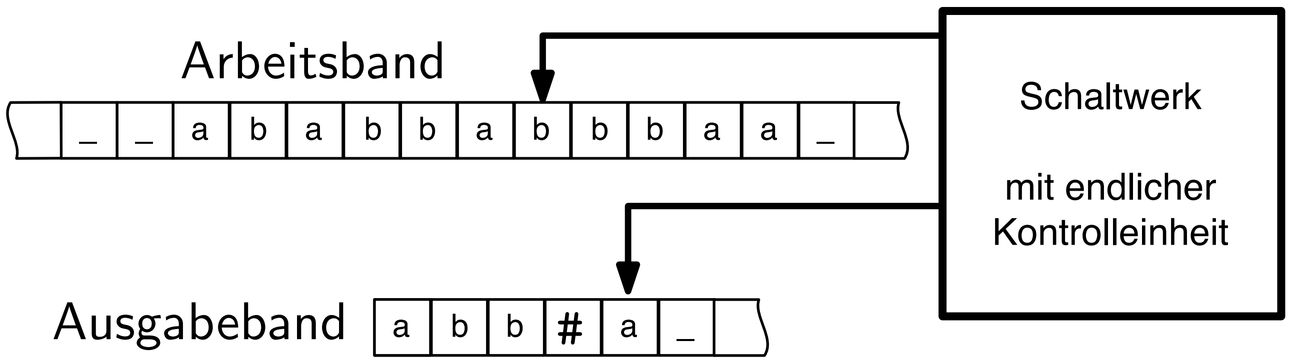
- ein **Aufzähler für** $L \subseteq \Sigma^*$ ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen
- TM schreibt auf das Ausgabeband $w_1\#w_2\#w_3\cdots$, wobei $\# \notin \Sigma$ und $L = \bigcup_i w_i$

TM als Aufzähler

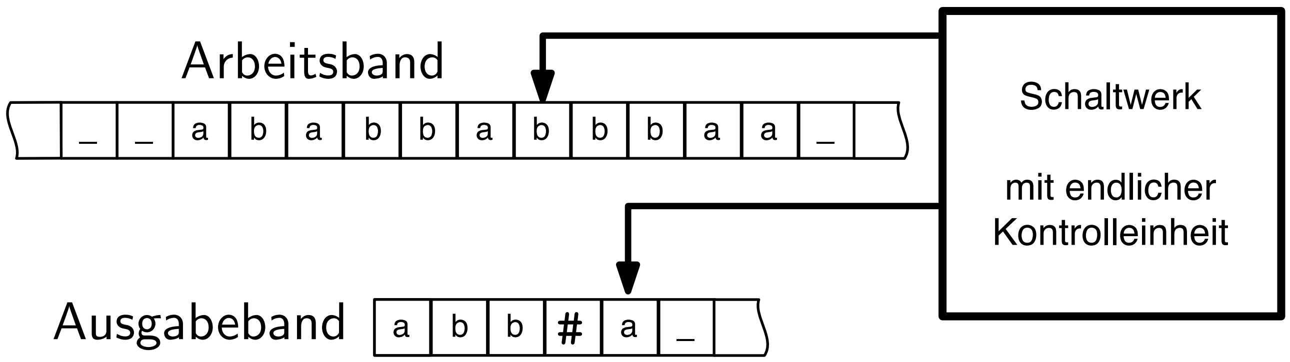
- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen
- TM schreibt auf das Ausgabeband $w_1\#w_2\#w_3\cdots$, wobei $\# \notin \Sigma$ und $L = \bigcup_i w_i$
- Achtung: Wiederholungen der Wörter aus L erlaubt

TM als Aufzähler

- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen
- TM schreibt auf das Ausgabeband $w_1\#w_2\#w_3\cdots$, wobei $\# \notin \Sigma$ und $L = \bigcup_i w_i$
- Achtung: Wiederholungen der Wörter aus L erlaubt
- Sprachen, für die es einen Aufzähler gibt, heißen **aufzählbar**

Beispiel $L = \Sigma^* = \{0, 1\}^*$

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.
- Modul A: Zähle Σ^i auf (Eingabe Arbeitsband 0^i)
  1. Kopiere Arbeitsband$+\ \#$ auf Ausgabe
  2. Wenn auf Arbeitsband 1^* steht, verlasse Modul
  3. Addiere$+1$ auf Binärzahl auf Arbeitsband
  4. Gehe zu 1.

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.
- Modul A: Zähle Σ^i auf (Eingabe Arbeitsband 0^i)
  1. Kopiere Arbeitsband$+$ $\#$ auf Ausgabe
  2. Wenn auf Arbeitsband 1^* steht, verlasse Modul
  3. Addiere$+1$ auf Binärzahl auf Arbeitsband
  4. Gehe zu 1.
- Modul: Addiere $+1$
  1. Gehe nach rechts
  2. Wenn Kopf auf 1 ersetze 1 durch 0, gehe einen Schritt nach links und wiederhole 2.
  3. Wenn 0 ersetze 0 durch 1 und stoppe Modul

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.
- Modul A: Zähle Σ^i auf (Eingabe Arbeitsband 0^i)
  1. Kopiere Arbeitsband$+$ $\#$ auf Ausgabe
  2. Wenn auf Arbeitsband 1^* steht, verlasse Modul
  3. Addiere$+1$ auf Binärzahl auf Arbeitsband
  4. Gehe zu 1.
- Modul: Addiere $+1$
  1. Gehe nach rechts
  2. Wenn Kopf auf 1 ersetze 1 durch 0, gehe einen Schritt nach links und wiederhole 2.
  3. Wenn 0 ersetze 0 durch 1 und stoppe Modul
- Hauptmodul:
  1. Schreibe $\#0\#1\#$ auf das Ausgabe- und 1 aufs Arbeitsband
  2. Ersetze alle 1en durch 0en plus Extra-0
  3. Führe Modul A aus, dann wiederhole ab 2.