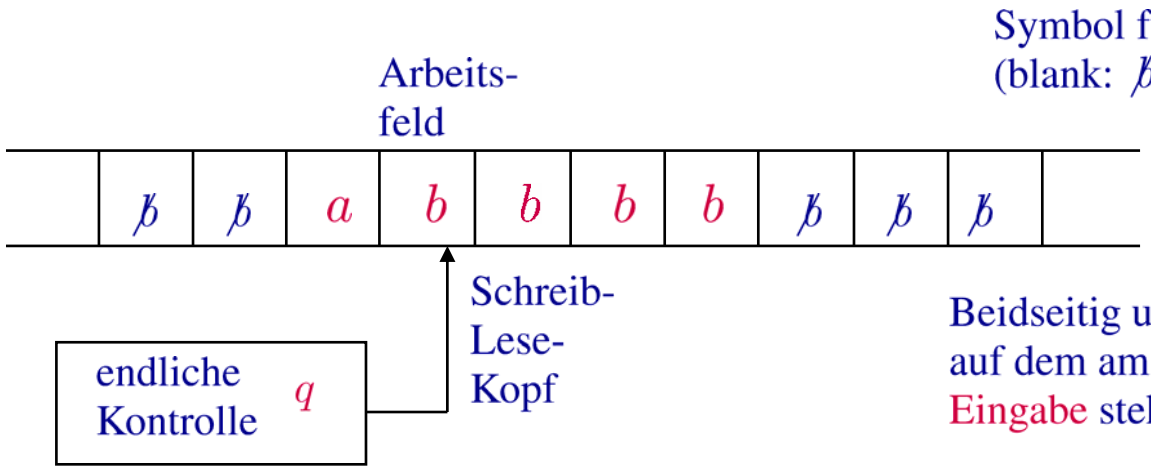
Inhaltsangabe

Teil III: Turingmaschinen und Grammatiken

Turingmaschinen liefern Automatenmodelle für Typ-0- und Typ-1-Sprachen.

§ 11. Turingmaschinen



Zu jedem Zeitpunkt sind nur endlich viele Symbole auf dem Band verschieden von ß.

Definition 11.1 (Turingmaschine)

Eine Turingmaschine über dem Eingabealphabet Σ hat die Form $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, \Delta, F)$, wobei

- Q endliche Zustandsmenge ist,
- Σ das Eingabealphabet ist,
- Γ das Arbeitsalphabet ist mit $\Sigma \subseteq \Gamma$, $\not b \in \Gamma \setminus \Sigma$,
- $q_0 \in Q$ der Anfangszustand ist,
- $F \subseteq Q$ die Endzustandsmenge ist und
- $\Delta \subseteq Q \times \Gamma \times \Gamma \times \{r, l, n\} \times Q$ die Übergangsrelation ist.

Bedeutung der Übergangsrelation $\Delta \subseteq Q \times \Gamma \times \Gamma \times \{r, l, n\} \times Q$

$(q, a, a', \begin{matrix} r \\ l \\ n \end{matrix}, q') \in \Delta$ sagt:

- Im Zustand q
- mit a auf dem gerade gelesenen Feld (Arbeitsfeld)

kann die Turingmaschine $\mathcal{A}$

- das Symbol a durch a' ersetzen,
- in den Zustand q' gehen und
- den Schreib-Lesekopf entweder um ein Feld nach rechts (r), links (l) oder nicht (n) bewegen.

Deterministische und nicht-deterministische Turingmaschinen:

Die Maschine $\mathcal{A}$ heißt deterministisch, falls es für jedes Tupel $(q, a) \in Q \times \Gamma$ höchstens ein Tupel der Form $(q, a, \ldots, \ldots, \ldots) \in \Delta$ gibt.

NTM steht im folgenden für (möglicherweise nicht-deterministische) Turingmaschinen und DTM für deterministische.

Bei einer DTM gibt es zu jedem Berechnungszustand höchstens einen Folgezustand, während es bei einer NTM mehrere geben kann.

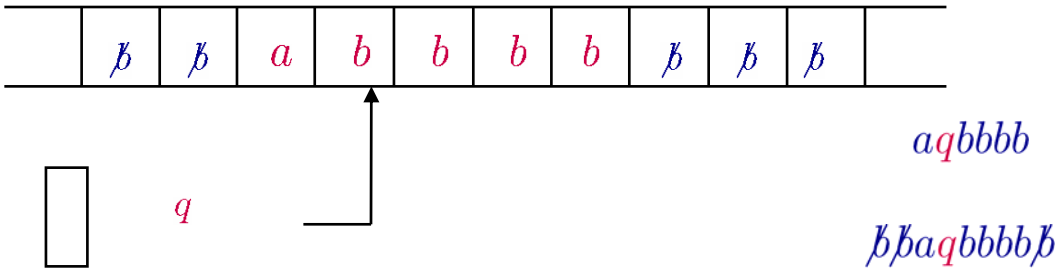
Berechnungszustand (Konfiguration) einer Turingmaschine:

kann man beschreiben durch ein Wort $\alpha q \beta$ mit $\alpha, \beta \in \Gamma^+$, $q \in Q$:

- q ist der momentane Zustand
- α ist die Beschriftung des Bandes links vom Arbeitsfeld
- β ist die Beschriftung des Bandes beginnend beim Arbeitsfeld nach rechts

Dabei werden (um endliche Wörter α, β zu erhalten) unendlich viele Blanks weggelassen, d.h. α, β umfassen mindestens den Bandabschnitt, auf dem Symbole $\neq \not b$ stehen.

Beispiel:



Die Übergangsrelation Δ ermöglicht die folgenden Konfigurationsübergänge:

es seien $\alpha, \beta \in \Gamma^+,\ \beta' \in \Gamma^*,\ a, b, a' \in \Gamma,\ q, q' \in Q$

- $\left.\begin{array}{rcl} \alpha q a \beta & \vdash_{\mathcal{A}} & \alpha a' q' \beta \\ \alpha q a & \vdash_{\mathcal{A}} & \alpha a' q' \not{b} \end{array}\right\}$ falls $(q, a, a', r, q') \in \Delta$

- $\left.\begin{array}{rcl} \alpha b q a \beta' & \vdash_{\mathcal{A}} & \alpha q' b a' \beta' \\ b q a \beta' & \vdash_{\mathcal{A}} & \not{b}\, q' b a' \beta' \end{array}\right\}$ falls $(q, a, a', l, q') \in \Delta$

- $\alpha q a \beta' \vdash_{\mathcal{A}} \alpha q' a' \beta'$ falls $(q, a, a', n, q') \in \Delta$

Dresden

Folgekonfiguration:

Gilt $k \vdash_{\mathcal{A}} k'$, so heißt k' Folgekonfiguration von k.

Akzeptierende Konfiguration:

Die Konfiguration $\alpha q \beta$ heißt akzeptierend, falls $q \in F$.

Stoppkonfiguration:

Die Konfiguration $\alpha q \beta$ heißt Stoppkonfiguration, falls sie keine Folgekonfiguration hat.

Die von $\mathcal{A}$ akzeptierte Sprache:

$$L(\mathcal{A}) := \{w \in \Sigma^* \mid \not{b}\, q_0 w \not{b} \vdash_{\mathcal{A}}^* k, \text{ wobei } k \text{ akzeptierende Stoppkonfiguration ist}\}$$

Definition 11.2 (Turing-akzeptierbar)

Die Sprache $L \subseteq \Sigma^*$ heißt Turing-akzeptierbar, falls es eine NTM $\mathcal{A}$ gibt mit $L = L(\mathcal{A})$.

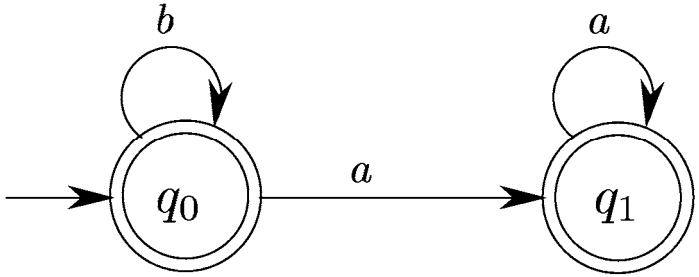
Beispiel

Die Sprache

$$L = \{w \in \{a, b\}^* \mid w \text{ enthält nicht den Infix } ab\}$$

ist Turing-akzeptierbar.

Die Turingmaschine, welche L akzeptiert, verhält sich wie der endliche Automat für diese Sprache, wobei sie das Eingabewort von links nach rechts liest, bis sie das erste Blank sieht (Eingabe ganz gelesen).



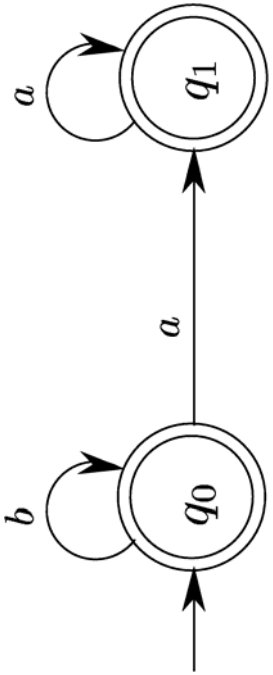
Beispiel

Die Sprache

$$L = \{w \in \{a,b\}^* \mid w \text{ enthält nicht den Infix } ab\}$$

ist Turing-akzeptierbar.

Die Turingmaschine, welche L akzeptiert, verhält sich wie der endliche Automat für diese Sprache, wobei sie das Eingabewort von links nach rechts liest, bis sie das erste Blank sieht (Eingabe ganz gelesen).



q_0	$\not b$	$\not b$	n	q_{akz}
q_0	b	b	r	q_0
q_0	a	a	r	q_1
q_1	a	a	r	q_1
q_1	$\not b$	$\not b$	n	q_{akz}

q_0 Anfangszustand

q_{akz} Endzustand

Beispiel

Die Sprache

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

ist Turing-akzeptierbar.

Die Turingmaschine, welche L akzeptiert, geht wie folgt vor:

- Sie ersetzt das erste a durch a', das erste b durch b' und das erste c durch c';
- läuft zurück zum zweiten a, ersetzt es durch a', das zweite b durch b' und das zweite c durch c' etc.
- Dies wird solange gemacht, bis nach erzeugtem c' ein $\not b$ steht.
- Danach wird von rechts nach links geprüft, ob – in dieser Reihenfolge – nur noch ein c'-Block, dann ein b'-Block und dann ein a'-Block (abgeschlossen durch $\not b$) vorhanden ist.
- Die Maschine blockiert, falls das nicht so ist. Dies kann auch bereits vorher geschehen, wenn ein erwartetes a, b, oder c nicht gefunden wird.

q_0	$\not b$	$\not b$	n	q_{akz}
q_0	a	a'	r	$finde\_b$
$finde\_b$	a	a	r	$finde\_b$
$finde\_b$	b'	b'	r	$finde\_b$
$finde\_b$	b	b'	r	$finde\_c$
$finde\_c$	b	b	r	$finde\_c$
$finde\_c$	c'	c'	r	$finde\_c$
$finde\_c$	c	c'	r	$zu\_Ende\_?$
$zu\_Ende\_?$	c	c	l	$zurück$
$zurück$	c'	c'	l	$zurück$
$zurück$	b	b	l	$zurück$
$zurück$	b'	b'	l	$zurück$
$zurück$	a	a	l	$zurück$
$zurück$	a'	a'	r	q_0
$zu\_Ende\_?$	$\not b$	$\not b$	l	$zu\_Ende\_!$
$zu\_Ende\_!$	c'	c'	l	$zu\_Ende\_!$
$zu\_Ende\_!$	b'	b'	l	$zu\_Ende\_!$
$zu\_Ende\_!$	a'	a'	l	$zu\_Ende\_!$
$zu\_Ende\_!$	$\not b$	$\not b$	n	q_{akz}

Varianten von Turingmaschinen:

In der Literatur werden verschiedene Versionen der Definition der Turingmaschine angegeben, die aber alle äquivalent zueinander sind, d.h. dieselben Sprachen akzeptieren und dieselben Funktionen berechnen.

Zwei Beispiele dafür sind:

- Turingmaschinen mit nach links begrenztem und nur nach rechts unendlichem Arbeitsband
- Turingmaschinen mit mehreren Bändern und Schreib-Leseköpfen

Wir betrachten das zweite Beispiel genauer und zeigen Äquivalenz zur in Definition 11.1 eingeführten 1-Band-TM.

Definition 11.3 (k-Band-TM)

Eine k-Band-NTM hat die Form $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, \Delta, F)$ mit

- $Q, \Sigma, \Gamma, q_0, F$ wie in Definition 11.1 und
- $\Delta \subseteq Q \times \Gamma^k \times \Gamma^k \times \{r, l, n\}^k \times Q$.

Bedeutung von $(q, (a_1, \ldots, a_k), (b_1, \ldots, b_k), (d_1, \ldots, d_k), q') \in \Delta$:

- Vom Zustand q aus
- mit $a_1, \ldots, a_k$ auf den Arbeitsfeldern der k Bänder

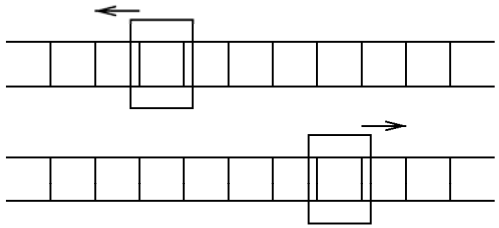
kann $\mathcal{A}$ das folgende tun:

- das Symbol a_i auf dem i-ten Band durch b_i ersetzen,
- in den Zustand q' gehen und
- die Schreib-Leseköpfe der Bänder entsprechend d_i bewegen.

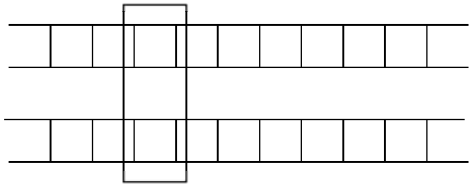
Das erste Band wird (o.B.d.A.) als Ein- und Ausgabeband verwendet.

Beachte:

Wichtig ist hier, daß sich die Köpfe der verschiedenen Bänder auch verschieden bewegen können:



Wären die Köpfe gekoppelt, so hätte man im Prinzip nicht mehrere Bänder, sondern ein Band mit mehreren Spuren:



k Spuren erhält man einfach, indem man eine normale NTM (nach Definition 11.1) verwendet, die als Bandalphabet Γ^k statt Γ hat.

Offenbar kann man jede 1-Band-NTM durch eine k-Band-NTM ($k > 1$) simulieren, indem man nur das erste Band wirklich verwendet.

Umkehrung?

Satz 11.4

> Wird die Sprache L durch eine k-Band-NTM akzeptiert, so auch durch eine 1-Band-NTM.

Beweis: Es sei $\mathcal{A}$ eine k-Band-NTM.

Gesucht:

- eine 1-Band-NTM $\mathcal{A}'$ und
- eine Kodierung $k \mapsto k'$ der Konfigurationen k von $\mathcal{A}$ durch Konfigurationen k' von $\mathcal{A}'$,

so daß gilt:

$$k_1 \vdash_{\mathcal{A}} k_2 \text{ gdw } k_1' \vdash_{\mathcal{A}'}^{*} k_2'$$

mehrere Schritte nötig zur Simulation eines Schrittes von $\mathcal{A}$

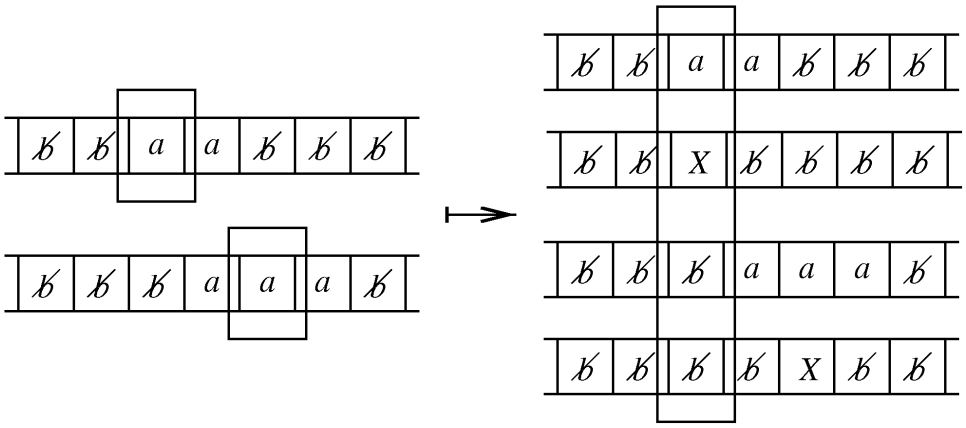
Arbeitsalphabet von $\mathcal{A}'$: $\Gamma^{2k} \cup \Sigma \cup \{\not b\}$

- $\Sigma \cup \{\not b\}$ wird für Eingabe benötigt;
- Γ^{2k} sorgt dafür, daß sich $\mathcal{A}'$ wie eine 1-Band-NTM mit $2k$ Spuren verhält.

Kodierung:

Jeweils 2 Spuren kodieren ein Band von $\mathcal{A}$:

- erste Spur enthält die Bandbeschriftung
- zweite Spur enthält eine Markierung X (und sonst Blanks), die zeigt, wo das Arbeitsfeld des Bandes ist



Das Arbeitsfeld von $\mathcal{A}'$ liegt stets bei dem am weitesten links stehenden X

Dresden

Initialisierung:

Zunächst wird die Anfangskonfiguration

$$\not b\, q_0 a_1 \ldots a_m \not b$$

von $\mathcal{A}'$ in die Kodierung der entsprechenden Anfangskonfiguration von $\mathcal{A}$ umgewandelt

durch entsprechende Übergänge

$\not b$	$\not b$	a_1	a_2	$\ldots$	a_m	$\not b$	Spur 1 und 2 kodieren
$\not b$	$\not b$	X	$\not b$	$\ldots$	$\not b$	$\not b$	Band 1
$\not b$	$\not b$	$\not b$	$\not b$	$\ldots$	$\not b$	$\not b$	Spur 3 und 4 kodieren
$\not b$	$\not b$	X	$\not b$	$\ldots$	$\not b$	$\not b$	Band 2

$\vdots$

Simulation der Übergänge:

Betrachte $(q, (a_1, \ldots, a_k), (b_1, \ldots, b_k), (d_1, \ldots, d_k), q') \in \Delta$.

- Von links nach rechts suche die mit X markierten Felder.

  Dabei merke man sich (in dem Zustand der TM) die Symbole, welche jeweils über dem X stehen. Außerdem zählt man (durch Zustand der TM) mit, wieviele X man schon gelesen hat, um festzustellen, wann das k-te erreicht ist.

  Man bestimmt so, ob das aktuelle Tupel tatsächlich $(a_1, \ldots, a_k)$ ist.

- Von rechts nach links gehend überdrucke man die a_i bei den X-Marken jeweils durch das entsprechende b_i und

  verschiebt die X-Marken gemäß d_i.

- Bleibe bei der am weitesten links stehenden X-Markierung, lasse den Kopf dort stehen und gehe in Zustand q'.

Bemerkung 11.5

War $\mathcal{A}$ deterministisch, so liefert obige Konstruktion auch eine deterministische 1-Band-Turingmaschine.

Bei endlichen Automaten sind nichtdeterministische Automaten nicht stärker als deterministische, bei Kellerautomaten aber schon.

Bei Turingmaschinen: sind DTM schwächer als NTM?

Satz 11.6

Zu jeder NTM gibt es eine DTM, die dieselbe Sprache akzeptiert.

Beweis:

Wegen Satz 11.4 und Bemerkung 11.5 genügt es, zu jeder NTM eine deterministische 3-Band-Turingmaschine zu konstruieren.

Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, \Delta, F)$ eine NTM.

Die Maschine $\mathcal{A}'$ soll für wachsendes n auf dem dritten Band jeweils alle Konfigurationsfolgen

$$k_0 \vdash_{\mathcal{A}} k_1 \vdash_{\mathcal{A}} k_2 \vdash_{\mathcal{A}} \ldots \vdash_{\mathcal{A}} k_n$$

beginnend mit der Startkonfiguration $k_0 = \not b\, q_0 w \not b$ erzeugen.

Die Kontrolle, daß tatsächlich alle solchen Folgen erzeugt werden, wird auf dem zweiten Band vorgenommen.

Das erste Band speichert das Eingabewort w

(damit man stets weiß, was k_0 sein muß).

Genauer:

maximaler Verzweigungsgrad der nichtdeterministischen Berechnung

Es sei

$$r \quad = \quad \text{maximale Anzahl von Transitionen in } \Delta \text{ pro festem Paar } (q, a) \in Q \times \Gamma$$

Eine Indexfolge $i_1, \ldots, i_n$ mit $i_j \in \{1, \ldots, r\}$ bestimmt dann von k_0 aus für n Schritte die Auswahl der jeweiligen Transition

und somit von k_0 aus eine feste Konfigurationsfolge

$$k_0 \vdash_{\mathcal{A}} k_1 \vdash_{\mathcal{A}} \ldots \vdash_{\mathcal{A}} k_n.$$

Aufzählung aller endlichen Konfigurationsfolgen:

- Zähle alle endlichen Wörter über $\{1, \ldots, r\}$ auf und
- zu jedem solchen Wort $i_1 \ldots i_n$ die zugehörige Konfigurationsfolge

Die Maschine $\mathcal{A}'$ realisiert dies auf den drei Bändern wie folgt:

- Auf Band 1 bleibt die Eingabe gespeichert.
- Auf dem zweiten Band werden sukzessive alle Wörter $i_1 \dots i_n \in \{1, \dots, r\}^*$ erzeugt.
- Für jedes dieser Wörter wird auf dem dritten Band die zugehörige Konfigurationsfolge realisiert.

  Erreicht man hierbei eine akzeptierende Stoppkonfiguration von $\mathcal{A}$, so geht auch $\mathcal{A}'$ in eine akzeptierende Stoppkonfiguration.