1 Turingmaschinen

Def. 1.1 Eine deterministische Turingmaschine $\tau = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup)$ besteht aus folgenden Komponenten:

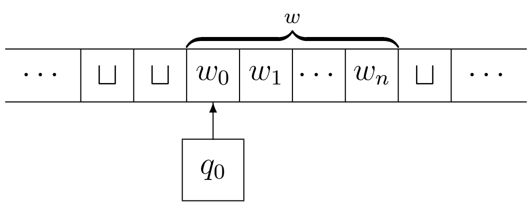
- Q ist eine endliche, nichtleere Menge von Zuständen,
- $\Sigma \subseteq \Gamma$ ist das Eingabealphabet,
- Γ ist eine endliche nichtleere Menge, das Bandalphabet, mit $Q \cap \Gamma = \emptyset$,
- $q_0 \in Q$ ist der Anfangszustand,
- $\sqcup \in \Gamma \setminus \Sigma$ ist das Leerzeichen oder Blank,
- $\delta : Q \times \Gamma \rightharpoonup Q \times \Gamma \times \{L, S, R\}$ ist die (partielle) Transitionsfunktion

unendliches/unbeschränktes Band, Lese-/Schreibkopf frei bewegbar, Eingabe auf Band; TM stoppt, wenn kein Nachfolger definiert ist.

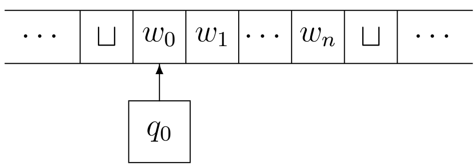
1.1 Arbeitsweise einer TM: informell

Konfigurationen der TM beschreiben den momentanen Zustand, den Bandinhalt und das betrachtete Feld.

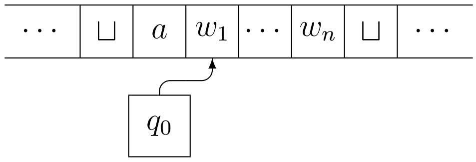
- Anfangskonfiguration:



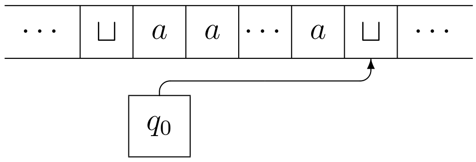
- Ausführen eines Schrittes:



z.B. $\delta(q_0, w_0) = (q_0, a, R)$ führt zu

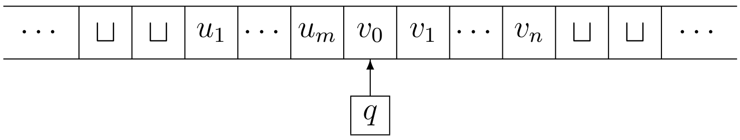


- Wiederholung dieses Schrittes liefert das Ergebnis:



  wobei $\delta(q_0, \sqcup)$ undefiniert sei.
  Das Ergebnis der Berechnung ist dann das Wort $aa \dots a$, d.h. der Bandinhalt ohne die Blanks.

Nur ein endlicher Ausschnitt des Bandes ist verschieden von $\sqcup$. Abstraktion von Blanks und Feldnummern. Die Situation



lässt sich eindeutig darstellen als Wort $\overbrace{u_1 \dots u_m}^{u} q \overbrace{v_0 v_1 \dots v_n}^{v}$ mit $u_i \in \Gamma, v_j \in \Gamma, m, n \geq 0$. Beachte: $Q \cap \Gamma = \emptyset$

1.2 Arbeitsweise einer TM: formal

Def. 1.2 Die Menge $\mathcal{K}_\tau$ der *Konfigurationen* einer TM $\tau = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup)$ ist durch

$$\mathcal{K}_\tau = \Gamma^* \cdot Q \cdot \Gamma^+$$

gegeben. Eine gegebene Konfiguration uqv bedeutet, dass sich die TM im Zustand q befindet, der Bandinhalt $\sqcup^\infty uv \sqcup^\infty$ ist und das erste (linkeste) Symbol des Wortes v gelesen wird.

Def. 1.3

(1) Die *Anfangskonfiguration* $\alpha(v)$ zu einem Wort $v \in \Sigma^*$ lautet

$$\alpha(v) = \begin{cases} q_0 v & \text{falls } v \neq \varepsilon \\ q_0 \sqcup & \text{sonst} \end{cases}$$

(2) Die *Transitionsrelation* $\vdash_\tau \subseteq \mathcal{K}_\tau \times \mathcal{K}_\tau$ ist wie folgt definiert:

$$K \vdash_\tau K' \qquad (K' \text{ heißt } \textit{Folgekonfiguration} \text{ von } K)$$

falls $\exists u, v \in \Gamma^* \; \exists a, b \in \Gamma \; \exists q, q' \in Q$:

$$\begin{array}{rllll} & (K = uqav & \wedge\ \delta(q,a) = (q', a', S) & \wedge\ K' = uq'a'v &) \\ \vee & (K = uqabv & \wedge\ \delta(q,a) = (q', a', R) & \wedge\ K' = ua'q'bv &) \\ \vee & (K = uqa & \wedge\ \delta(q,a) = (q', a', R) & \wedge\ K' = ua'q'\sqcup &) \\ \vee & (K = ubqav & \wedge\ \delta(q,a) = (q', a', L) & \wedge\ K' = uq'ba'v &) \\ \vee & (K = qav & \wedge\ \delta(q,a) = (q', a', L) & \wedge\ K' = q'\sqcup a'v &) \end{array}$$

(3) Eine *Endkonfiguration* ist eine Konfiguration $K \in \mathcal{K}_\tau$, die keine Folgekonfiguration besitzt.

(4) Das *Ergebnis* (oder die sichtbare Ausgabe) einer Konfiguration uqv ist

$$\omega(uqv) = \overline{uv},$$

wobei $\overline{u}$ das kürzeste Wort mit $u = \sqcup \ldots \sqcup \overline{u}$ und $\overline{v}$ das kürzeste Wort mit $v = \overline{v} \sqcup \ldots \sqcup$ ist. Man streicht also q, sowie führende und endende Blanks; Blanks, die sich zwischen Zeichen befinden, bleiben dagegen stehen.

Def. 1.4 Die von der TM $\tau = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup)$ *berechnete Funktion* ist

$$h_\tau : \quad \Sigma^* \xrightarrow{part} \Gamma^*$$

mit

$$h_\tau(v) = \begin{cases} w & \text{falls} \quad \exists \text{ Endkonfiguration } K \in \mathcal{K}_\tau : \\ & \qquad\quad \alpha(v) \vdash_\tau{}^* K \wedge w = \omega(K) \\ \text{undef.} & \text{sonst.} \end{cases}$$

Bemerkung Da δ eine (partielle) Funktion ist, ist $\vdash_\tau$ rechtseindeutig. Damit ist h_τ eine partielle Funktion.

Veranschaulichung der Resultatsfunktion:

$$\begin{tikzcd} \Sigma^* \ni v \arrow[rrr, "h_\tau"] \arrow[d, "\alpha"'] & & & w \in \Gamma^* \\ \mathcal{K}_\tau \ni \alpha(v) \vdash_\tau & \cdots & \vdash_\tau & K \in \mathcal{K}_\tau \arrow[u, "\omega"'] \end{tikzcd}$$

Def. 1.5 Eine Menge $M \subseteq \Sigma^*$ heißt *Haltebereich* oder *Definitionsbereich* von τ, falls gilt:

$$M = \{v \in \Sigma^* \mid h_\tau(v) \text{ ist definiert } \}$$

Eine Menge $N \subseteq \Gamma^*$ heißt *Ergebnisbereich* oder *Wertebereich* von τ, falls

$$N = \{w \in \Gamma^* \mid \exists v \in \Sigma^* : h_\tau(v) = w\}.$$

Def. 1.6 Es seien A, B Alphabete.

(i) Eine partiell definierte *Funktion* $f : A^* \overset{part}{\longrightarrow} B^*$ heißt *Turing-berechenbar*, falls es eine TM $\tau = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup)$ gibt mit $A = \Sigma$, $B \subseteq \Gamma$ und $f = h_\tau$, d.h. $f(v) = h_\tau(v)$ für alle $v \in A^*$.

(ii) $\mathcal{T}_{A,B} =_{def} \{f : A^* \overset{part}{\longrightarrow} B^* \mid \; f \text{ ist Turing-berechenbar } \}$

(iii) $\mathcal{T}$ sei die Klasse aller Turing-berechenbaren Funktionen (für beliebige Alphabete A, B).

Def. 1.7 Es sei A ein Alphabet.

(i) Eine *Menge* $L \subseteq A^*$ heißt *Turing-entscheidbar*, falls die *charakteristische Funktion von* L

$$\chi_L : \quad A^* \longrightarrow \{0, 1\}$$

Turing-berechenbar ist. Hierbei ist χ_L folgende totale Funktion:

$$\chi_L(v) = \begin{cases} 1 & \text{falls } v \in L \\ 0 & \text{sonst} \end{cases}$$

(ii) Analoge Definition für *Eigenschaften* $E : A^* \longrightarrow \{$ *wahr, falsch* $\}$.

Bemerkung Berechnung mehrstelliger Funktionen: Trennsymbol #.
Berechnung zahlentheoretischer Funktionen: Unärdarstellung mit Strichen |.

Konstruieren von Turingmaschinen: die Flussdiagrammschreibweise

Man definiert zunächst nützliche elementare TM. Sei $\Gamma = \{a_0, \ldots, a_n\}$ das Bandalphabet mit $a_0 = \sqcup$.

- Kleine Rechtsmaschine r:
  geht einen Schritt nach rechts und hält dann. Turingtafel:

r	q_0	a_0	q_e	a_0	R
	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
	q_0	a_n	q_e	a_n	R

- Kleine Linksmaschine l:
  geht einen Schritt nach links und hält dann. Turingtafel:

l	q_0	a_0	q_e	a_0	L
	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
	q_0	a_n	q_e	a_n	L

- Druckmaschine a für $a \in \Gamma$:
  druckt das Symbol a und hält dann. Turingtafel:

a	q_0	a_0	q_e	a	S
	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
	q_0	a_n	q_e	a	S

Wir nehmen im Folgenden an, dass alle konstruierten TM genau einen *Endzustand* besitzen, d.h. es gibt einen Zustand q_e, so dass für alle Endkonfigurationen uqv gilt:

$$q = q_e.$$

Offenbar erfüllen die TM r, l, a diese Bedingung. Solche TM können wir wie folgt zusammensetzen:

$\tau_1 \xrightarrow{a} \tau_2$ bedeutet anschaulich, dass zuerst τ_1 arbeitet. Hält τ_1 auf einem Feld mit dem Symbol a an, wird τ_2 gestartet.

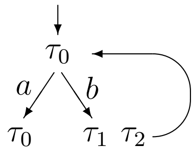
$\tau_1 \longrightarrow \tau_2$ bedeutet, dass zuerst τ_1 arbeitet. Sobald τ_1 anhält, wird τ_2 gestartet.

$\tau_1\tau_2$ ist eine Abkürzung für $\tau_1 \longrightarrow \tau_2$.

$\tau_1 \xrightarrow{\neq a} \tau_2$ bedeutet, dass zuerst τ_1 arbeitet. Hält τ_1 auf einem Feld mit dem Symbol $\neq a$ an, wird τ_2 gestartet.

Aus gegebenen TM können Flussdiagramme aufgebaut werden. Die Knoten dieser Flussdiagramme sind mit den Namen der TM bezeichnet. Die Kanten werden durch Pfeile der Form $\xrightarrow{a}$, $\longrightarrow$ oder $\xrightarrow{\neq a}$ bezeichnet. Schleifen sind erlaubt. Eine TM τ im Flussdiagramm ist durch einen Pfeil $\longrightarrow \tau$ als Start-TM gekennzeichnet.

Veranschaulichung:



Ein Flussdiagramm beschreibt eine „große“ TM, deren Turingtafel man wie folgt erhält:

Schritt 1: Für jedes Vorkommen einer TM τ_i im Flussdiagramm die zugehörige Turingtafel aufstellen.

Schritt 2: Zustände in verschiedenen Tafeln disjunkt machen.

Schritt 3: Gesamttafel erzeugen, indem alle Einzeltafeln (in irgendeiner Reihenfolge) untereinander geschrieben werden.

Schritt 4: *Kopplung*: für jeden Pfeil $\tau_1 \xrightarrow{a} \tau_2$ im Flussdiagramm füge der Gesamttafel die Zeile

$$q_{e\tau_1} a q_{0\tau_2} a S$$

hinzu. Dabei sei $q_{e\tau_1}$ der (gemäß Schritt 2 umbenannte) Endzustand von τ_1 und $q_{0\tau_2}$ der (gemäß Schritt 2 umbenannte) Anfangszustand von τ_2. Analog für $\tau_1 \to \tau_2$ und $\tau_1 \overset{\neq a}{\to} \tau_2$.

Anwendung: Man gebe eine TM zur Berechnung der „minus“-Funktion an:

$$f : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$

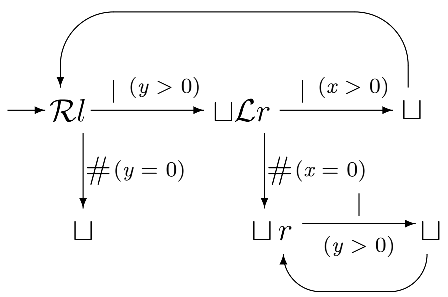
mit

$$f(x,y) = x \dot{-} y = \begin{cases} x - y & \text{für } x \geq y \\ 0 & \text{sonst} \end{cases}$$

Die Anfangskonfiguration der TM ist:

$$\sqcup q_0 \underbrace{|\ldots|}_{x \text{ Mal}} \# \underbrace{|\ldots|}_{y \text{ Mal}} \sqcup$$

Idee: x-Striche so oft löschen, wie y-Striche da sind. Dann restliche y-Striche und # löschen.
Konstruktion der TM mit Hilfe eines Flussdiagramms:



1.3 Nichtterminierende Berechnungen

Bereits bekannt: Halteproblem; Situation bei anderen Automatenmodellen?

- DEAs: $|w|$ Schritte
- NEAs: $|w|$ Schritte bzw. $\leq |Q|^{|w|}$ Schritte für explizite Suche
- ε-NEAs: sättige mit ε-Transitionen: $\leq 2|w|+1$ Schritte bzw. $\leq |Q|^{2|w|+1}$ Schritte für explizite Suche (immer abwechselnd ein Symbol lesen und eine ε-Transition nehmen), weil ε-Schleifen den Gesamtzustand des Automaten nicht ändern (ob wir sie einmal oder zweimal nehmen, spielt keine Rolle)
- PDAs: wandle in Normalform um, z.B. PDA $\rightarrow$ CFG $\rightarrow$ CNF $\rightarrow$ PDA; betrachte dann nur Konfigurationen mit Stacks der Tiefe $\leq |w|$
- LBA (kontextsensitive Sprachen) = TM mit Bandlänge $|w|$: Terminierung durch Duplikatdetektion (nur endlich viele Konfigurationen)