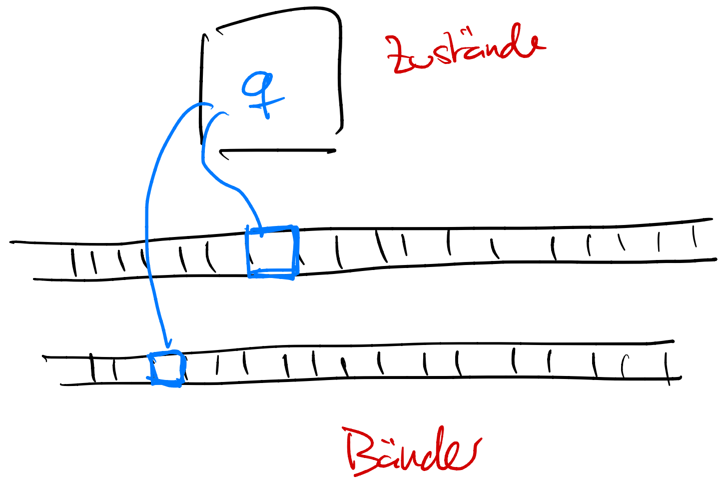
Berechenbarkeit

2 Turingmaschinen

Wir betrachten das folgende, sehr bekannte, Berechnungsmodell.
Anschaulich lässt es sich wie folgt beschreiben.



- Es gibt einen „Speicher"
  ⇝ k unendlich lange Arrays (Bänder)
- Es gibt einen „Arbeitsspeicher"
  ⇝ eine endl. Menge von Zuständen, die die Maschine einnehmen kann
- Für jedes Band gibt es einen Schreib- und Lesekopf
- Jeder Schritt ist wie folgt:
  Abhängig vom Zustand und gelesenen Symbolen, schreiben die Köpfe genau ein Symbol, bewegen sich um max. eine Position und der Zustand der

Maschine wird geändert.

- Stellt die Maschine ihr schrittweises Arbeiten ein, so wird die Ausgabe entweder dem Zustand entnommen oder von einem der Bänder in geeigneter Weise abgelesen.

Definition 2.1 (Turingmaschine, Alan Turing, 1936)

Sei $k \in \mathbb{N}$. Eine k-Band-Turingmaschine, kurz k-TM, ist ein Tupel

$M = (Q, \Sigma, \Gamma, \Delta, s, F)$. Dabei ist

- Q eine endliche Menge, Zustandsmenge,
- Σ das Eingabealphabet, ein Alphabet mit $\square \notin \Sigma$
- Γ das Bandalphabet, ein Alphabet mit $\Sigma \subseteq \Gamma$ und $\square \in \Gamma \setminus \Sigma$
- $\Delta \subseteq Q \times \Gamma^k \times Q \times \Gamma^k \times \{L, S, R\}^k$ die Übergangsrelation
- $s \in Q$ der Startzustand
- $F \subseteq Q$ die Menge der akzeptierenden Zustände.

Das Symbol $\square$ heißt Blank. Die Element von Δ heißen Instruktionen. Für eine Instruktion $(q, a_1, \ldots, a_k, q', a_1', \ldots, a_k', B_1, \ldots, B_k)$ heißt $(q, a_1, \ldots, a_k)$ der Bedingungsteil und $(q', a_1', \ldots, a_k', B_1, \ldots, B_k)$ Anweisungsteil.

Die TM M ist eine deterministische k-Band Turingmaschine, kurz k-DTM, wenn es $\forall b \in Q \times \Gamma^k$ höchstens eine Instruktion $i \in \Delta$ mit Bedingungsteil b.

Definition 2.2 (Konfiguration)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ ein k-TM. Eine Konfiguration von M ist ein Tupel

$$C = (q, w_1, \ldots, w_k, p_1, \ldots, p_k) \in Q \times (\Gamma^*)^k \times \mathbb{N}^k$$

mit $p_i \in [|w_i|]$ $\forall i \in [k]$.

Die Startkonfiguration von M zur Eingabe $(u_1, \ldots, u_n) \in (\Sigma^*)^n$, wobei $n \in \mathbb{N}$, ist die Konfiguration

$$\mathrm{start}_M(u_1, \ldots, u_n) = (s, u_1 \square u_2 \square \ldots \square u_n, \square, \ldots, \square, 1, \ldots, 1).$$

Die Konfiguration C ist eine Stoppkonfiguration von M, wenn es keine Instruktion $i \in \Delta$ mit Bedingungsteil $(q, w_1(p_1), \_\_, w_k(p_k))$ gibt.

Definition 2.3 (Nachfolgekonfiguration)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM. Für Konfig.

$$C = (q, w_1, \_\_, w_k, p_1, \_\_, p_k) \quad \text{und} \quad C' = (q', w_1', \_\_, w_k', p_1', \_\_, p_k')$$

von M ist die Konfig. C' Nachfolgekonfiguration von C, wenn es eine Instruktion

$$(q, w_1(p_1), \_\_, w_k(p_k), q', a_1', \_\_, a_k', B_1, \_\_, B_k) \in \Delta$$

gibt, sodass

$$w_i' = \begin{cases} \square a_i' \, w_i(2) \_\_ w_i(|w_i|) & \text{falls } p_i = 1 \text{ und } B_i = L \\ w_i(1) \_\_ w_i(|w_i|-1) \, a_i' \, \square & \text{falls } p_i = |w_i| \text{ und } B_i = R \\ w_i(1) \_\_ w_i(p_i - 1) \, a_i' \, w_i(p_i + 1) \_\_ w_i(|w_i|) & \text{sonst} \end{cases}$$

und

$$p_i' = \begin{cases} 1 & \text{falls } p_i = 1 \text{ und } B_i = L \\ p_i - 1 & \text{falls } p_i \geq 2 \text{ und } B_i = L \\ p_i & \text{falls } B_i = S \\ p_i + 1 & \text{falls } B_i = R \end{cases}$$

für $\forall i \in [k)$ gelten.

Es bezeichne $\rightarrow_M$ die Relation auf der Menge der Konfig von M sodass

$C \rightarrow_M C'$ falls C, C' Konfig von M sind wobei C' eine Nachfolgekonfig. von C ist.

Definition 2.4 (Rechnung)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM. Eine endliche partielle Rechnung von M ist eine endl. Folge $C_1, \dots, C_n$ von Konfig. von M mit $C_i \rightarrow_M C_{i+1}$ $\forall i \in [n-1]$.

Eine unendliche partielle Rechnung von M ist eine unendl. Folge $C_1, C_2, \dots$ von Konfig. von M mit $C_i \rightarrow_M C_{i+1}$ $\forall i \in \mathbb{N}$.

Eine Rechnung von M zur Eingabe $(w_1, \dots, w_k) \in (\Sigma^*)^k$ (mit $k \in \mathbb{N}$) ist eine

eine endliche partielle Rechnung $\text{start}_M(w_1, \ldots, w_n) = C_1, \ldots, C_m$ bei der C_m eine Stoppkonfig. von M ist oder eine unendliche part. Rechnung $\text{start}_M(w_1, \ldots, w_n) = C_1, C_2, \ldots$

Bemerkung 2.5

Ist M eine k-DTM, so gibt es $\forall n \in \mathbb{N}$ und $(w_1, \ldots, w_n) \in (\Sigma^*)^n$ genau eine Rechnung zur Eingabe $(w_1, \ldots, w_n)$.

Definition 2.6 (total)

Eine k-DTM $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ terminiert bei Eingabe $(w_1, \ldots, w_n) \in (\Sigma^*)^n$ wenn die Rechnung von M zur Eingabe $(w_1, \ldots, w_n)$ endlich ist. Eine k-TM $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ ist total, wenn $\forall n \in \mathbb{N}$ und $(w_1, \ldots, w_n) \in (\Sigma^*)^n$ alle Rechnungen von M zur Eingabe $(w_1, \ldots, w_n)$ endlich sind.

Definition 2.7 (akzeptierte Sprache)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM. Eine Stoppkonfig $(q, w_1, \ldots, w_k, p_1, \ldots, p_k)$ von M ist akzeptierend, wenn $q \in F$. Die akzeptierte Sprache $L(M)$ von M

ist die Sprache über dem Alphabet Σ so dass $\forall w \in \Sigma^*$ genau dann $w \in L(M)$ gilt, wenn es eine endl. Rechnung $C_1, \ldots, C_n$ von M zur Eingabe w gibt, bei der C_n eine akzep. Stoppkonfig. von M ist.

Hinweis: Für nicht det. TM heißt das insbesondere, dass es für die Wörter w in der akzept. Sprache nur mind. eine in einer akzep. Stoppkonfig. endende endliche Rechnung zur Eingabe w geben muss.
Für Wörter w, die nicht in $L(M)$ sind, sind alle Rechnungen von M zur Eingabe am Ende nicht in einer akzept. Stoppkonfig oder unendlich.

Definition 2.8 (entscheidbar)
Eine Sprache L ist genau dann entscheidbar, wenn es eine totale k-TM M mit $L(M) = L$ gibt. Wir schreiben REC für die Klasse der entscheidbaren Sprachen.

Der Begriff entscheidbar für Sprachen ergibt sich hier daraus, dass effektiv entschieden werden kann ob eine gegebene Eingabe in der Sprache liegt oder nicht. Insb. setzt dies voraus, dass Eingaben, die nicht in der Sprache liegen effektiv als nicht in der Sprache liegend erkannt werden.

<u>Begriff:</u> effektiv ⇝ eine TM erledigt dies in endl. Zeit

Da sich der durch TM formalisierte Berechenbarkeitsbegriff, also die Formalisierung dessen was effektiv durchführbar ist, auch äquivalent durch rekursive Funktion definieren lässt, werden entscheidbare Sprachen auch als rekursiv bezeichnet.

Definition 2.9 (rekursiv aufzählbar)

Eine Sprache L ist genau dann <u>rekursiv aufzählbar</u>, wenn es eine k-TM mit akzeptierter Sprache L gibt. Wir schreiben <u>RE</u> für die Klasse der rekursiv aufzählbaren Sprachen.

Die Aufzählbarkeit leitet sich hier daraus ab, dass es für eine rek. aufz. Sprache L über einem Alphabet Σ möglich ist effektive Verfahren anzugeben, die die Wörter von L aufzählen, also dass eine endlich oder unendliche Aufzählung von $A = w_1, w_2, \ldots$ mit $\{w_1, w_2, \ldots\} = L$ existiert.

Bemerkung 2.10

Jede entscheidbare Sprache ist rek. aufzählb.

Bemerkung 2.11

Alle endlichen Sprachen sind entscheidbar.

Bemerkung 2.12

Eine Sprache L über einem Alphabet Σ ist genau dann entscheidbar, wenn L und $L^c := (\Sigma^*) \setminus L$ rekursiv aufzählb. sind.

Definition 2.13 (Ausgabe)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM und $C = (q, w_1, \ldots, w_k, p_1, \ldots, p_k)$ eine Konfig. von M. Die Ausgabe $out_M(C)$ von M bei Konfig. C ist das Präfix

$w \sqsubseteq w_1(p_1) \ldots w_1(|w_1|)$ max. Länge mit $w \in (\Gamma \setminus \{\square\})^*$.

Definition 2.14 (berechnete Funktion)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-DTM und $n \in \mathbb{N}$. Die von M berechnete <u>n-äre partielle Funktion</u> φ_M ist die part. Fkt $\varphi_M : (\Sigma^*)^n \to (\Gamma \setminus \{\square\})^*$, so dass $\forall (w_1, \ldots, w_n) \in (\Sigma^*)^n$ folgendes gilt:

- Ist die Rechnung von M zur Eingabe $(w_1, \ldots, w_n)$ die endliche Rechnung $C_1, \ldots, C_m$, so gilt $\varphi_M(w_1, \ldots, w_n) = out_M(C_m)$.

- Ist die Rechnung von M zur Eingabe $(w_1, \ldots, w_n)$ unendlich, so gilt $\varphi_M(w_1, \ldots, w_n)\uparrow$

Für $w_1, \dots, w_n \in \Sigma^*$ schreiben wir statt $\varphi_M(w_1, \dots, w_n)$ auch $M(w_1, \dots, w_n)$.

Definition 2.15 (partiell berechenbar)

Für Alphabete Σ, Γ und eine part. Fkt $\varphi: \Sigma^* \rightsquigarrow \Gamma^*$ ist φ partiell berechenbar, wenn es ein $k \in \mathbb{N}$ gibt und eine k-DTM M mit $\varphi_M = \varphi$ gibt.
Ist φ total und partiell berechenbar, so ist φ berechenbar.
Wir schreiben RF für die Klasse der part. berechenbar part. Fkt.

Mittels der Indentifikation von $\mathbb{N}_0$ und $\{0,1\}^*$ können so auch part. Fkt., die von oder nach $\mathbb{N}_0$ abbilden als (part) berechenbare Fkt. bezeichnet werden.
Beispielsweise ist eine part. Fkt $\varphi: \mathbb{N}_0 \rightsquigarrow \mathbb{N}_0$ demnach genau dann part. berechenbar, wenn die part. Fkt $\mathrm{bin} \circ \varphi \circ \mathrm{bin}^{-1}$ part. berechenbar ist.

Gewissermaßen verfügen die hier def. TM über zwei Ausgabemechanismen.
Die Ausgabe im engeren Sinne in Def 2.13 und das Ablesen von Akzeptanz anhand des schließlich erreichten Zustands in Def 2.7. Im Sinne der folg. Bemerkung wäre der zweite Fall nicht notwendig, allerdings ist dies ein wichtiger Spezialfall.

Definition 2.16 (charakteristische Funktion, part. charakteristische Fkt)

Sei L eine Sprache über dem Alphabet Σ.

(i) Die charakteristische Funktion von L als Sprache über Σ ist die Fkt
$\mathbb{1}_L : \Sigma^* \to \{0,1\}$ mit $\mathbb{1}_L(w) = 1 \ \forall w \in L$ und $\mathbb{1}_L(w) = 0 \ \forall w \in \Sigma^* \setminus L$.

(ii) Die partiell charakteristische Funktion von L als Sprache über Σ ist die part. Fkt $\chi_L : \Sigma^* \to \{1\}$ mit $\chi_L(w) = 1 \ \forall w \in L$ und $\chi_L(w)\uparrow \ \forall w \in \Sigma^* \setminus L$.

Bemerkung 2.17