

Berechenbarkeit

Script, Kapitel 2

- Intuitiver Berechenbarkeitsbegriff
- Turing-Berechenbarkeit
- WHILE-Berechenbarkeit
- Church'sche These
- Entscheidungsprobleme
- Unentscheidbarkeit des Halteproblems für Turingmaschinen

Intuitiver Algorithmusbegriff

Ein Algorithmus

- überführt **Eingabedaten** in **Ausgabedaten** (wobei die Art der Daten vom Problem, das durch den Algorithmus gelöst werden soll, abhängig ist),
- besteht aus einer **endlichen Folge von Anweisungen** mit folgenden Eigenschaften:
 - es gibt eine **eindeutig** festgelegte Anweisung, die als **erste** auszuführen ist,
 - nach Abarbeitung einer Anweisung gibt es eine **eindeutig** festgelegte Anweisung, die als **nächste** abzuarbeiten ist, oder die Abarbeitung des Algorithmus ist beendet und hat eindeutig bestimmte Ausgabedaten geliefert,
 - die Abarbeitung einer Anweisung erfordert **keine Intelligenz** (ist also prinzipiell durch eine Maschine realisierbar).

Beispiele für Algorithmen

- der GAUSSsche¹ Algorithmus zur Lösung von linearen Gleichungssystemen (über den rationalen Zahlen),
- Kochrezepte (mit Zutaten und Kochgeräten als Eingabe und dem fertigen Gericht als Ausgabe),
- Bedienungsanweisungen für Geräte,
- PASCAL-Programme.

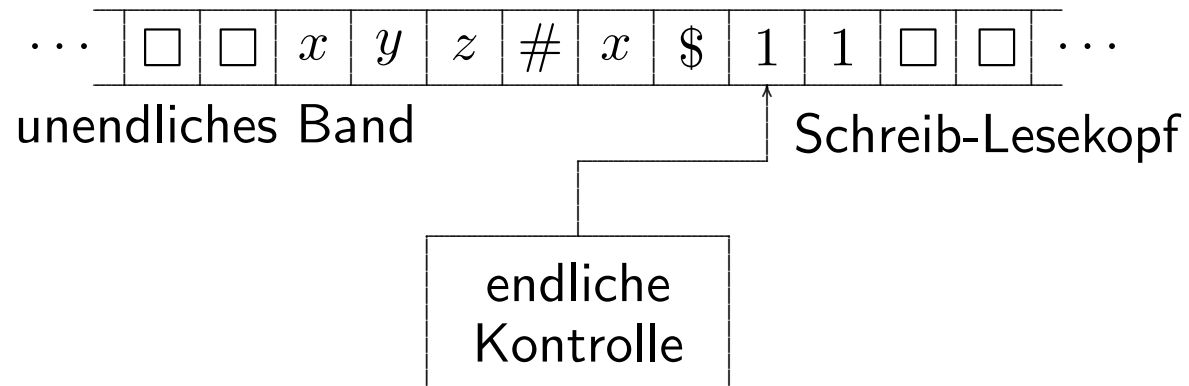
¹CARL FRIEDRICH GAUSS, 1777–1855, deutscher Mathematiker.

Idee der Turingmaschine²

- **Arbeitsband**
 - beidseitig (potenziell) unendliches Band, in Felder eingeteilt;
 - jedes Feld enthält einen Buchstaben aus dem **Bandalphabet**;
 - zum Bandalphabet gehört das **Leerzeichen** oder **Blank-Symbol** □.
- **Schreib-Lesekopf**
 - über dem Band beweglich
 - bearbeitet ein aktuelles Feld
- **endliche Steuerung** oder **Kontrolle**
 - endliche Menge von internen **Zuständen**, die den Programmablauf regeln

²ALAN MATHISON TURING (1912–1954), britischer Mathematiker, Logiker, Kryptoanalytiker und Computerkonstrukteur.

Veranschaulichung einer Turingmaschine



Definition der Turingmaschine

Eine (deterministische) Turingmaschine (kurz: TM) ist gegeben durch ein 7-Tupel $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$. Hierbei sind

- Z eine endliche Menge (Zustandsmenge),
- Σ ein Alphabet (Eingabealphabet),
- Γ ein Alphabet (Bandalphabet) mit $\Sigma \subseteq \Gamma$,
- $\delta: (Z \setminus E) \times \Gamma \rightarrow Z \times \Gamma \times \{L, R, N\}$ eine Funktion (Überföhrungsfunktion),
- $z_0 \in Z$ (Anfangszustand),
- $\square \in \Gamma \setminus \Sigma$ (Leerzeichen, Blank),
- $E \subseteq Z$ (Menge der Endzustände).

Erstes Beispiel einer Turingmaschine

Gegeben sei die Turingmaschine

$$M = (\{z_0, z_1, z_2, z_e\}, \{0, 1\}, \{0, 1, \square\}, \delta, z_0, \square, \{z_e\}),$$

wobei δ wie folgt definiert ist. Wir geben dabei δ in einer Tabelle an, wobei im Kreuzungspunkt der Zeile mit der Bezeichnung a und der Spalte mit der Bezeichnung z der Funktionswert $\delta(z, a)$ steht.

δ	z_0	z_1	z_2
\square	(z_1, \square, L)	$(z_e, 1, N)$	(z_e, \square, R)
0	$(z_0, 0, R)$	$(z_2, 1, L)$	$(z_2, 0, L)$
1	$(z_0, 1, R)$	$(z_1, 0, L)$	$(z_2, 1, L)$

Konfiguration einer Turingmaschine

Eine **Konfiguration** k einer Turingmaschine $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ ist ein Wort $k \in \Gamma^* Z \Gamma^*$. Dabei soll $k = \alpha z \beta$ folgendermaßen interpretiert werden:

- $\alpha\beta$ steht auf dem Eingabeband, des weiteren stehen nur noch Blankzeichen \square auf dem Band,
- die Turingmaschine befindet sich im Zustand z und
- der Kopf der Turingmaschine steht über dem ersten Symbol von β .

Eine **Startkonfiguration** ist $k_0 = z_0 w$ mit $w \in \Sigma^*$.

Eine **Endkonfiguration** ist $k_e = \square^m z_e w' \square^n$ mit $z_e \in E$, $w' \in \Sigma^*$, $m, n \in \mathbb{N}$.

Die binäre Relation \vdash in der Menge der Konfigurationen

(Änderung der Konfiguration **in einem Schritt** der TM)

$$a_1 \dots a_m z b_1 \dots b_n$$
$$\vdash \begin{cases} a_1 \dots a_m z' c b_2 \dots b_n & \text{falls } \delta(z, b_1) = (z', c, N), m \geq 0, n \geq 1, \\ a_1 \dots a_m c z' b_2 \dots b_n & \text{falls } \delta(z, b_1) = (z', c, R), m \geq 0, n \geq 2, \\ a_1 \dots a_{m-1} z' a_m c b_2 \dots b_n & \text{falls } \delta(z, b_1) = (z', c, L), m \geq 1, n \geq 1, \end{cases}$$

$$a_1 \dots a_m z b_1 \vdash a_1 \dots a_m c z' \square \text{ falls } \delta(z, b_1) = (z', c, R), m \geq 0,$$

$$z b_1 \dots b_n \vdash z' \square c b_2 \dots b_n \text{ falls } \delta(z, b_1) = (z', c, L), n \geq 1.$$

Die binäre Relation \vdash^*

(Änderung der Konfiguration **in endlich vielen Schritten** der TM)

Mit \vdash^* bezeichnen wir den reflexiven und transitiven Abschluss der binären Relation \vdash , also es gilt $k_0 \vdash^* k_e$ genau dann, wenn

- (i) $k_0 = k_e$ ist, oder
- (ii) eine Zahl $n \geq 0$ und Konfigurationen k_1, k_2, \dots, k_n existieren, so dass

$$k_0 \vdash k_1 \vdash k_2 \vdash \dots \vdash k_n \vdash k_e.$$

Turingberechenbarkeit

Eine Funktion $f : \Sigma^* \rightarrow \Sigma^*$ heißt **Turingberechenbar**, falls es eine Turingmaschine $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ gibt, so dass für alle $x, y \in \Sigma^*$ gilt:

$f(x) = y$ genau dann, wenn $z_0x \vdash^* \square \dots \square z_e y \square \dots \square$ für ein $z_e \in E$.

Eine Funktion $f : \mathbb{N}^k \rightarrow \mathbb{N}$ für ein $k \in \mathbb{N}$ heißt **Turingberechenbar**, falls es eine Turingmaschine $M = (Z, \{0, 1, \#\}, \Gamma, \delta, z_0, \square, E)$ gibt, so dass für alle $n_1, n_2, \dots, n_k, m \in \mathbb{N}$ gilt:

$f(n_1, n_2, \dots, n_k) = m$ genau dann, wenn für ein $z_e \in E$:
 $z_0 \text{bin}(n_1) \# \text{bin}(n_2) \# \dots \# \text{bin}(n_k) \vdash^* \square \dots \square z_e \text{bin}(m) \square \dots \square$

Beispiele für Turingberechenbare Funktionen

Die **Nachfolgerfunktion**

$$f: \mathbb{N} \rightarrow \mathbb{N} \text{ vermöge } n \mapsto f(n) = n + 1$$

ist Turingberechenbar, da die Turingmaschine aus unserem ersten Beispiel die Eingabe $\text{bin}(n)$ in die Ausgabe $\text{bin}(n + 1)$ transformiert.

Die **für alle Wörter** aus $\{a, b\}^*$ **nicht-definierte Funktion**

$$\Omega: \{a, b\}^* \rightarrow \{a, b\}^* \text{ vermöge } w \mapsto \Omega(w) = \textit{nicht definiert}$$

ist Turing-berechenbar, da sie von der Turingmaschine

$$M = (\{z_0, z_e\}, \{a, b\}, \{a, b, \square\}, \delta, z_0, \square, \{z_e\})$$

mit $\delta(z_0, x) = (z_0, x, N)$ für alle $x \in \{a, b, \square\}$ berechnet wird.

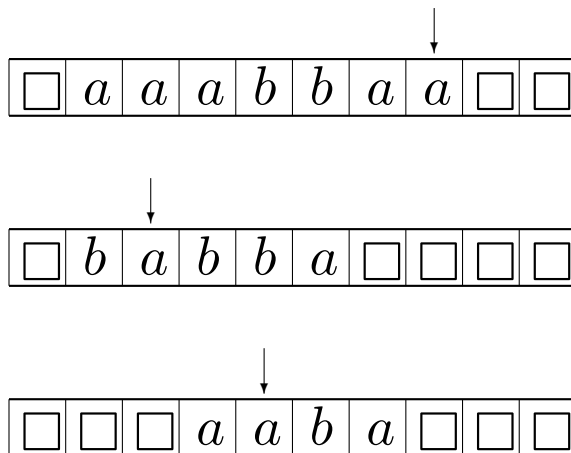
Mehrband-Turingmaschinen

- $k \geq 1$ Bänder (Eingabe/Ausgabe auf Band 1)
- Jedes Band hat einen eigenen Schreib-Lesekopf, der separat bewegt wird.
- Formal $\delta: Z \times \Gamma^k \rightarrow Z \times \Gamma^k \times \{L, R, N\}^k$

Äquivalenz von Mehrband- und (Einband)-TM

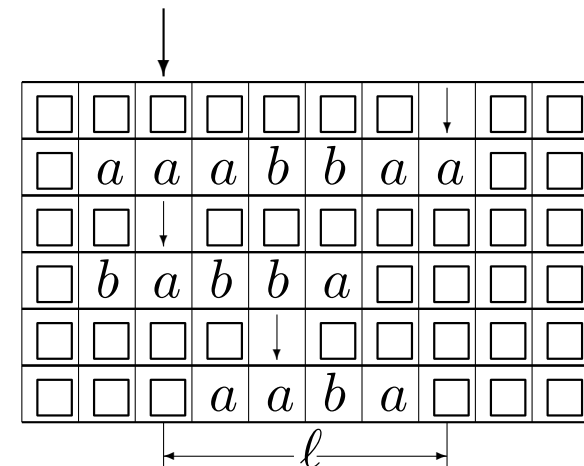
Zu jeder Mehrband-TM M gibt es eine (Einband-)TM M' , die dieselbe Funktion berechnet wie M .

Mehrband-TM



- k Bänder
- 1 Schritt

Einband-TM



- 1 Band mit $2k$ Spuren
- $\approx 2(\ell + k)$ Schritte

Notationen

“ $\dot{-}$ ”: modifizierte Subtraktion

$$\dot{-} : \mathbb{N}^2 \rightarrow \mathbb{N} \text{ verm\"oge } (n_1, n_2) \mapsto n_1 \dot{-} n_2 = \begin{cases} n_1 - n_2 & \text{falls } n_1 \geq n_2, \\ 0 & \text{sonst.} \end{cases}$$

“ $\text{Band} := \text{Band} + 1$ ”: TM, die zu einer Zahl 1 dazuaddiert

“ $\text{Band}(i) := \text{Band}(i) + 1$ ”: k -Band-TM, die auf i -tem Band 1 addiert und alle anderen Bänder unverändert läßt

“ $\text{Band}(i) := \text{Band}(i) \dot{-} 1$ ”: k -Band-TM, die auf i -tem Band 1 modifiziert subtrahiert und alle anderen Bänder unverändert läßt

“ $\text{Band}(i) := \text{Band}(j)$ ”: k -Band-TM, die Inhalt des j -ten Bandes auf i -tes Band kopiert

Nacheinanderausführung von Turingmaschinen

Seien M_1 und M_2 zwei Turingmaschinen, so wollen wir durch

$$\text{start} \longrightarrow M_1 \longrightarrow M_2 \longrightarrow \text{stop}$$

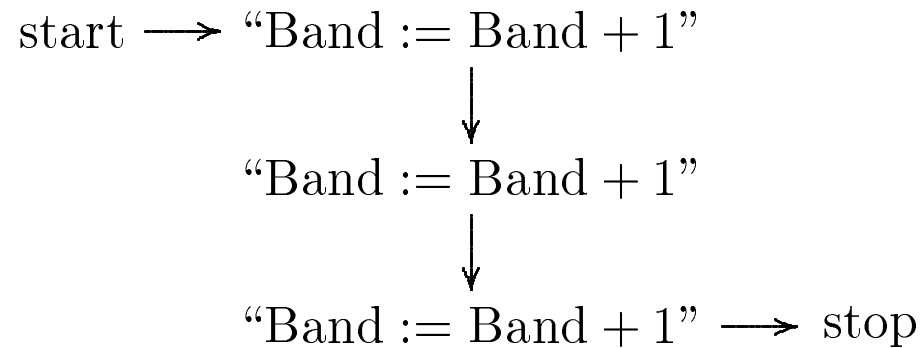
oder auch durch

$$M_1; M_2$$

diejenige Turingmaschine verstehen, die zuerst wie die Turingmaschine M_1 arbeitet und wenn M_1 einen Stopzustand erreichen würde in den Anfangszustand von M_2 übergeht und jetzt wie die Turingmaschine M_2 arbeitet. Sie stoppt dann, wenn M_2 einen Stopzustand erreichen würde.

Beispiel für Nacheinanderausführung von TM

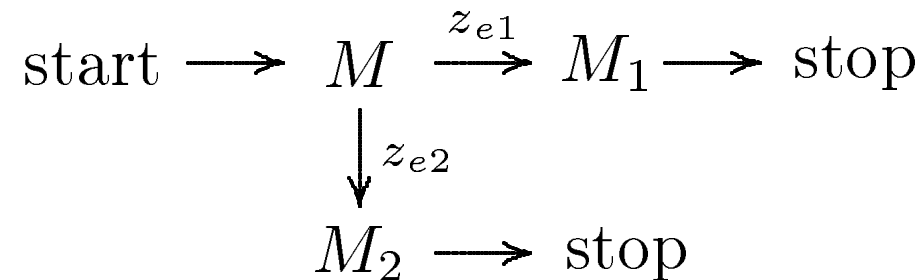
Betrachten wir das Diagramm in folgender Abbildung.



So erkennen wir, dass dort das schematische Flussbild einer Turingmaschine steht, welche dreimal nacheinander zur Zahl auf dem Band 1 addiert, also es sich um die Turingmaschine "Band := Band + 3" handelt.

Beispiel für Verzweigung von TM

Folgende Abbildung stellt eine sich verzweigende TM dar.



Sie soll nach Simulation der Turingmaschine M die Turingmaschine M_1 simulieren, falls sie bei der Simulation von M im Zustand z_{e1} stoppt. Analog soll sie die Turingmaschine M_2 abarbeiten, falls sie bei der Simulation von M im Zustand z_{e2} stoppt.

Beispiel für “Band (i) = 0 ?”

Es sei $M = (\{z_0, z_1, ja, nein\}, \Sigma, \Gamma, \delta, z_0, \square, \{ja, nein\})$ mit $0 \in \Sigma$ sowie mit der Überföhrungsfunktion δ , gegeben durch

$$\delta(z_0, a) = (nein, a, N) \quad \text{für } a \neq 0,$$

$$\delta(z_0, 0) = (z_1, 0, R),$$

$$\delta(z_1, a) = (nein, a, L) \quad \text{für } a \neq \square,$$

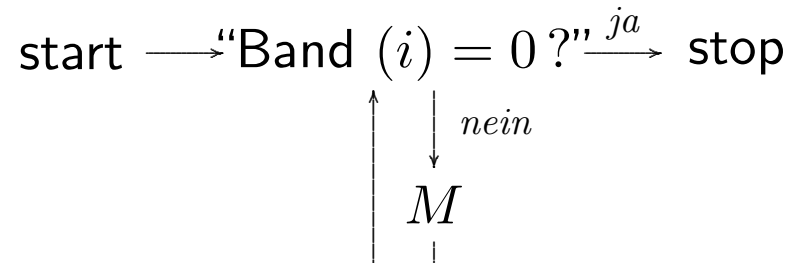
$$\delta(z_1, \square) = (ja, \square, L).$$

Diese Turingmaschine testet, ob die Eingabe genau das Wort 0 ist. Falls ja, stoppt sie im Zustand *ja*, falls nein, stoppt sie im Zustand *nein*. Wir wollen diese Turingmaschine mit “Band = 0 ?” bezeichnen.

Die *k*-Band-TM, die das *i*-te Band auf 0 testet, nennen wir “Band(*i*) = 0 ?”

Beispiel für WHILE-Schleife

Sei M eine beliebige Turingmaschine.



Dann bezeichnen wir die Turingmaschine, die durch das Diagramm gegeben ist, mit **“WHILE Band(i) \neq 0 DO M ”**. Die Arbeitsweise ist einfach zu erkennen.