# Turingmachine

*A Turing machine is like a wise old person, sitting at an endless table, playing a complex game. They have a magical pen that reads and writes on the game board. They follow strict rules, do not move from their spot, but the table mysteriously moves back and forth. Their concentration is deep and calm as they perform a complex ballet of reading, writing, and state-changing.*
*- ChatGPT*

# Berechenbarkeit

## 1 Turingmachine

Wir Betrachte das folgende, sehr bekannt, berechnunsmodell. Anschaulich lässt es sich wie folht beschreiben.

- Es gibt einen "Speicher" $\rightsquigarrow$ k unendlich lange Arrays(**Bänder**)
- Es gibt einen "Arbeitsspeicher" $\rightsquigarrow$ eine endliche Menge von Zusänden, die die Machine einnehmen kann
- Für jedes Band gibt es einen Schreib- und Lesekopf
- Jeder Schritt ist wie folgt:
  Abhängig von Zustand und gelesenene Symbol, Schreiben die Küpfe genau ein Symbol, bewegen sich nun maximal eine Position und der Zustand der Machine wird geändert.
- Stellt die Machine ihhr schrittweises Arbeiten ein, so wird die Ausgabe entweder den Zustand entnommen oder von einem der Bänder in geeigneter Weise abgelesen.

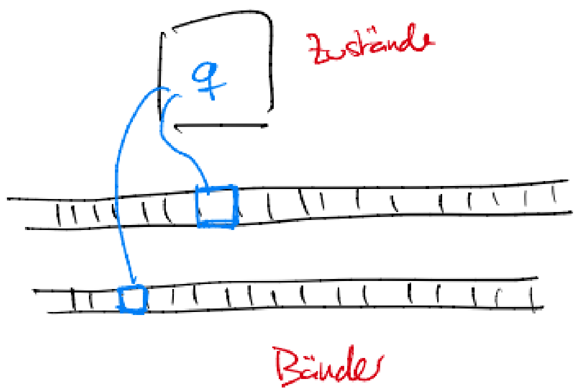


Figure 1: Turingmachine.

### 1.1 Definition (Turingmachine, Alan Tuing, 1936)

Sei $k \in \mathbb{N}$ eine **k-Band-Turingmachinem** kurz k-TM, ist ein Tupe $M = (Q, \Sigma, \Gamma, \Delta, s, F)$. Dabei ist:

- Q eine endliche Menge, **Zustandmenge**
- $\Sigma$ das **Eingabealphabet**, ein Alphabet $\square \notin \Sigma$
- $\Gamma$ das **Bandaphabet**, ein Alphabet mit $\Sigma \subseteq \Gamma$ und $\square \in \Gamma / \Sigma$
- $\Delta \subseteq Q \times \Gamma^k \rightarrow \subseteq Q \times \Gamma^k \times L, S, R^k$ die **Übergangsrelation**
- $s \in Q$ der **Startzustand**
- $F \subseteq Q$ die Menge der **akzeptierenden Zustände**

Das Symbol $\square$ heißt **Blank**. Die Elemente von $\Delta$ heißen **instruktionen**. Für eine Instruktion $(q_1, a_1, \cdots, a_k, q', a'_1, \cdots, a'_k, B_1, \cdots, B_k)$ **Anweisungteil**. Die TM M ist eine **deterministische k-Band Turingmachine**, kurz k-DTM, wenn es $\forall b \in Q \times \Gamma^k$ höchstens eine Instruktion $i \in \Delta$ mit Bedingungsteil b.

### 1.2 Definition (Konfiguration)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM. EIne **Konfigration** von M ist ein Tupel

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

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

$$Start_M(u_1, \cdots, u_n) = (s, u_1 \Box u_2 \Box \cdots \Box u_n, \Box, \cdots, 1, \cdots, 1)$$

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

### 1.3 Definition (Nachfolgekonfiguration)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-DTM. Für Konfiguration $C = q_1, w_1, \cdots, w_k, p_1, \cdots, p_k$ und $C' = q'_1, w'_1, \cdots, w'_k, p'_1, \cdots, p'_k$ von M ist die Konfigration C' Nachfolgekonfiguration von C, wenn es eine Instruktion

$$(q, w_1(p_1), \cdots, w_k(p_k), a'_1, a'_k, B_1, \cdots, B_k) \in \Delta$$

gibt, sodass

$$w'_i = \begin{cases} \Box a'_i w_i(2) \cdots w_i(|w_i|), & \text{falls } p_i = 1 \text{und} B_i = L \\ w_i \cdots w_i(|w_i| - 1) a'_i \Box, & \text{falls } pi = |w_i| \text{und} B_i = R \\ w_i \cdots w_i(p_i - 1) a'_i w_i(p_i + 1) \cdots w_i(|w_i|), & \text{sonst} \end{cases}$$

und

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

$\forall i \in [k]$ gelten.
Es bezeichnen $\rightarrow M$ die Relation auf der Menge der Konfiguration von M, sodass $C \rightarrow_M C'$ falls C, C' Konfig von M sind wobei C' eine Nachfolgekonfiguration von C ist.

### 1.4 Definition (Rechnung)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-DTM. Eine **endliche partielle Rechnung** von M ist eine endliche Folge $C_1, \cdots, C_n$ 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 unendliche Folge $C_1, C_2, \cdots$ von Konfigration von M mit $C_1 \rightarrow_M C_{1+1} \forall i \in \mathbb{N}$. Eine **Rechnung von M zur Eingabe** $(w_1, \cdots, w_n) \in (\Sigma^*)^n$ (mit $n \in \mathbb{N}$) ist eine endliche partielle Rechnung $start_M = C_1, \cdots, C_m$ bei der $C_m$ eine Stoppkonfiguration von M oder eine unendliche partielle rechnung $start_M(w_1, \cdots, w_n) = C_1, C_2, \cdots$

### 1.5 Bemerkung

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

### 1.6 Definition (total)

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

## 1.7 Definition (akzeptierte Sprache)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM. Eine Stoppkonfiguration $(q, w_1, \cdots, w_k, p_1, \cdots, p_k)$ von M ist **akzeptierend**, wenn $q \in F$. Die **akzeptierte Sprache L(M)** von M ist die Sprache über dem Alphabet $\Sigma$ so dass $w \in L(M)$ gilt, wenn es eien endliche Rechnung $C_1, \cdots, C_n$ von M zur Eingabe w gibt, bei der $C_n$ eine akzeptierende Stoppkonfigration von M ist.
**Hinweis:** Für nicht deterministische TM heißt das insbesondere, dass es für die Wörter w in der akzeptierten Sprache nur mindestend **eine** im einer akzeptierten Stoppkonfigration 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 akzeptierten Stoppkonfigration oder unendlich.

## 1.8 Definition(entscheidbar)

Eine Sprache L ist genau dann **entscheidbar**, wenn es eien 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. Insbesondere steht? dies voraus, dass Eingabe, die nicht in der Sprache liegen effektiv als nicht in der Sprache liegend erkannt werden.
**Begriff:** effektiv $\rightsquigarrow$ eine TM erlefigt dies in endicher Zeit. Da sich der durch TM formatierte Berechenbarkeitsbegriff, also die Formalisierung dessen was effektiv durchführbar ist, auch äquivalent durch rekursive Funktion definieren lässt, weden entscheidbare Sprachen auch als rekuriv bezeichnet.

## 1.9 Definition(rekursiv aufzählbar)

Eine Sprache L ist genau dann **rekursiv aufzähbar**, wenn es eine k-TM mit akzeptierten Sprache L gibt. Wir schreiben **RE** für die Klasse der rekursiv aufzählbaren Sprachen. Die Aufzählbarkeit leitet sih daraus ab, dass es für eine rekuriv aufzählbare Sprache L über einem Alühabet $\Sigma$ möglich ist effektive Verfahren anzugeben ,die die Wörter von L aufzählen, also dass eine endlich oder unendliche Aufzählung von $A = w_1, w_2, \cdots$ mit $w_1, w_2, \cdots = L$ existiert.

## 1.10 Bemerkung

Jede entscheidbare Sprache ist rekursiv aufzähbar.

## 1.11 Bemerkung

Alle endlichen Sprachen sind entscheidbar.

## 1.12 Bemerkung

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

## 1.13 Definition (Ausgabe)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-TM und $C = (q, w_1, \cdots, w_k, p_1, \cdots, p_k)$ eine Konfigration von M. Die Ausgabe $out_M(C)$ von M bei Konfiguration C ist das Präfix $w \sqsubseteq w_1(p_1), \cdots, w_1(|w_1|)$ maximale Länge mit $w \in (\Gamma/\Box)^*$.

## 1.14 Definition (berechnete Funktion)

Sei $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ eine k-DTM und $n \in \mathbb{N}$. Die von M berechnete **n-äre partielle Funktion $\Phi_M$** ist die partielle Funktion $\Phi_M : (\Sigma^*)^n \rightsquigarrow (\Gamma/\Box)^*$, so dass $\forall (w_1, \cdots, w_n) \in (\Sigma^*)^n$ folgendes gilt:

1. Ist die rechnung von M zur Eingabe $(w_1, \cdots, w_n)$ die endliche Rechnung $C_1, \cdots, C_M$, so gilt $\Phi_M(w_1, \cdots, w_n) = out_M(C_M)$.
2. Ist die Rechnung von M zur Eingabe $(w_1, \cdots, w_n)$ unendlich, so gilt $\Phi_M(w_1, \cdots, w_n) \uparrow$

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

### 1.15 Definition (partiell berechenbar)

Für Alphabet $\Sigma, \Gamma$ und eine partielle Funktion $\Phi : \Sigma^* \rightsquigarrow \Gamma^*$ ist $\Phi$ **partiell berechenbar**, wenn es eine $k \in \mathbb{N}$ gibt und eine k-DTM M mit $\Phi_M = \Phi$ gibt. Ist $\Phi$total und partiell berechenbar, so ist $\Phi$ berechenbar. Wir schreiben **RF** für die Klasse der partiellen Funktionen.////Mittels der Induktivität von $\mathbb{N}_0$ und $0, 1^*$ können

### 1.16 Bemerkung (normiert)

Eine 1-DTM $M = (Q, \Sigma, \Gamma, \Delta, s, F)$ heißt **normiert**, wenn $Q = 0, \cdots, n$ für eine $n \in \mathbb{N}_0$, $\Sigma = 0, 1$, $\Gamma = \square, 0, 1$, $s = 0$, $F = s$. Alle TMs mit Eingabealphabet $0, 1$ lassen sich mit folgenden Schritten in eine normierte TM mit gleicher erkannter Sprache und gleicher berechneter Funktion umwandeln.

- Von Nichtdeterminismus zu Determinismus: Eine DTM kann die Rechnungen einer nichtdeterministischen TM parallel im Sinne von abwechend schrittweise durchführen um schließlich das Verhalten der simulierten TM zu ??. Dies entspricht einer **Breitensuche im Rechnungsbaum**. ADD IMAGE HERE
- Von mehreren Bändern zu einem Band: Intuitiv können k Bänder auf ein Band simuliert werden, indem die Felder des einen Bandes in k-teilfelder unterteilt werden, die jeweils die gleiche Bandalphabetbuchstaben wie zufor als Beschreibung zulassen und es zudem erlaubt zu markieren, dass der simulierte Kopf des simulierten Bandes dort steht. Eine dieser Idee folgende Konstruktion wird als **Spurentechnik** bezeichnet. Formal: Übergang vom Bandalphabet $\Gamma$ zu

$$((\Gamma \cup \underline{a} : a \in \Gamma)^k / \square^k) \cup \square$$

wobei $\underline{a} \notin \Gamma$ für $a \in \Gamma$. Hierbei bedeutet $\underline{\text{a}}$, dass das simulierte Feld mit a beschriftet ist und dass dort der simulierte Kopf steht. Weiter spielt $\square$ die Rolle des k-Tupels $(\square, \cdots, \square)$ um der Tatsache gerecht zu werden, dan alle Felderzu Begin mit $\square$ beschriftet sind.