# Theoretische Grundlagen der Informatik

## Vorlesung am 22.11.2022

Torsten Ueckerdt | 22. November 2022

# Letzte Vorlesung

(deterministische) **Turing-Maschine**

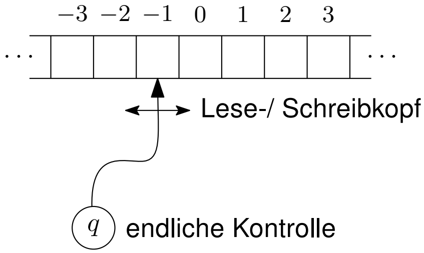
- Bestandteile:
  - Lese-/Schreibkopf ______ liest/schreibt Zeichen auf Band
  - endliche Kontrolle ______ stets in einem Zustand
  - unendliches Rechenband ______ enthält Eingabe
- formal: $\mathcal{M} = (Q, \Sigma, \sqcup, \Gamma \supseteq (\Sigma \cup \{\sqcup\}), s \in Q, \delta, F \subseteq Q)$
- Übergangsfunktion: $\delta(q, a) = (q', a', X \in \{L, N, R\})$

# Letzte Vorlesung

(deterministische) **Turing-Maschine**

- Bestandteile:
  - Lese-/Schreibkopf ______ liest/schreibt Zeichen auf Band
  - endliche Kontrolle ______ stets in einem Zustand
  - unendliches Rechenband ______ enthält Eingabe
- formal: $\mathcal{M} = (Q, \Sigma, \sqcup, \Gamma \supseteq (\Sigma \cup \{\sqcup\}), s \in Q, \delta, F \subseteq Q)$
- Übergangsfunktion: $\delta(q, a) = (q', a', X \in \{L, N, R\})$
- Der Übergang $\delta(q, a) = (p, b, L)$ wird graphisch

  dargestellt als $q \xrightarrow{a|b,\ L} p$

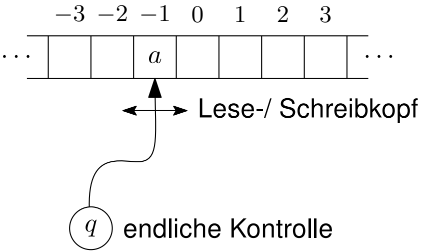


# Letzte Vorlesung

(deterministische) **Turing-Maschine**

- Bestandteile:
  - Lese-/Schreibkopf ______ liest/schreibt Zeichen auf Band
  - endliche Kontrolle ______ stets in einem Zustand
  - unendliches Rechenband ______ enthält Eingabe
- formal: $\mathcal{M} = (Q, \Sigma, \sqcup, \Gamma \supseteq (\Sigma \cup \{\sqcup\}), s \in Q, \delta, F \subseteq Q)$
- Übergangsfunktion: $\delta(q, a) = (q', a', X \in \{L, N, R\})$
- Der Übergang $\delta(q, a) = (p, b, L)$ wird graphisch

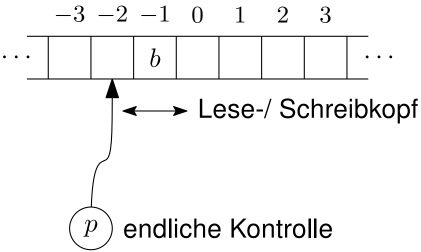
  dargestellt als $q \xrightarrow{a|b,\ L} p$



# Letzte Vorlesung

(deterministische) **Turing-Maschine**

- Bestandteile:
  - Lese-/Schreibkopf ______ liest/schreibt Zeichen auf Band
  - endliche Kontrolle ______ stets in einem Zustand
  - unendliches Rechenband ______ enthält Eingabe
- formal: $\mathcal{M} = (Q, \Sigma, \sqcup, \Gamma \supseteq (\Sigma \cup \{\sqcup\}), s \in Q, \delta, F \subseteq Q)$
- Übergangsfunktion: $\delta(q, a) = (q', a', X \in \{L, N, R\})$
- Der Übergang $\delta(q, a) = (p, b, L)$ wird graphisch dargestellt als $q \xrightarrow{a|b,\ L} p$

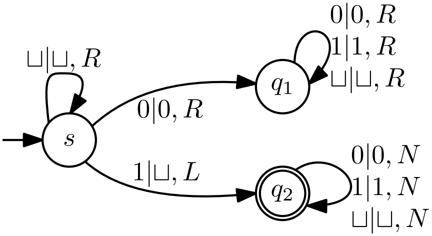


# Konventionen bei der TM

- Die Turing-Maschine startet im Zustand $s$.
- Der Lese-/Schreibkopf startet an der linkesten Stelle des Bandes, in der ein Eingabesymbol steht.
- Die Turing-Maschine stoppt, wenn sie
  - zum ersten Mal in einen Endzustand kommt, oder
  - in einem Zustand $q$ ein Symbol $a$ liest und $\delta(q, a) = (q, a, N)$ ist.
- Das bedeutet, dass Übergänge, die aus Endzuständen herausführen, überflüssig sind.

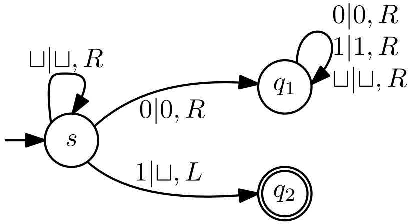
# Beispiel-Turing-Maschine

- Die Turing-Maschine stoppt, wenn sie
  - zum ersten Mal in einen Endzustand kommt, oder
  - in einem Zustand *q* ein Symbol *a* liest und $\delta(q, a) = (q, a, N)$ ist.



# Beispiel-Turing-Maschine

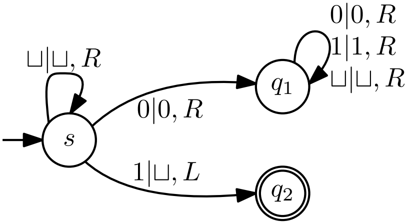
**Frage: Was erkennt / berechnet diese TM ?**



# Beispiel-Turing-Maschine

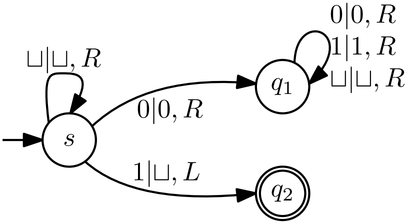
**Frage: Was erkennt / berechnet diese TM ?**

- Die TM erkennt alle Wörter aus $\{0, 1\}^*$, die mit einer Eins beginnen.
- Die TM löscht die führende Eins, falls vorhanden.
- Alles andere auf dem Band bleibt unverändert.
- Der Lese-/Schreibkopf steht nach dem Stop links neben der Stelle an der die führende Eins gelesen wurde.



# Beispiel-Turing-Maschine

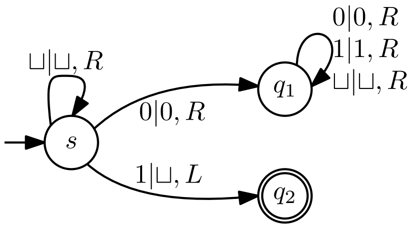
- Es gibt Eingaben, für die eine Turing-Maschine unter Umständen niemals stoppt.
- **Welche Eingaben sind dies in diesem Beispiel?**



# Beispiel-Turing-Maschine

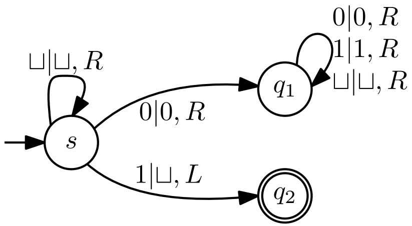
- Es gibt Eingaben, für die eine Turing-Maschine unter Umständen niemals stoppt.
- **Welche Eingaben sind dies in diesem Beispiel?**

- Die TM stoppt nicht, falls die Eingabe nicht mit Eins beginnt.



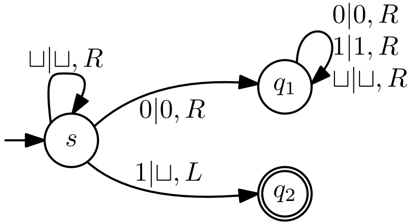
# Beispiel-Turing-Maschine

- Eine Turing-Maschine erkennt nicht nur eine Sprache,
- sondern sie verändert auch die Eingabe, und
- hat insofern auch eine Ausgabe (= Inhalt des Bandes nach der akzeptierender Bearbeitung).
- Die Turing-Maschine realisiert also eine partielle Funktion $f\colon \Sigma^* \to \Gamma^*$.



# Beispiel-Turing-Maschine

- Eine Turing-Maschine erkennt nicht nur eine Sprache,
- sondern sie verändert auch die Eingabe, und
- hat insofern auch eine Ausgabe (= Inhalt des Bandes nach der akzeptierender Bearbeitung).
- Die Turing-Maschine realisiert also eine partielle Funktion $f\colon \Sigma^* \to \Gamma^*$.

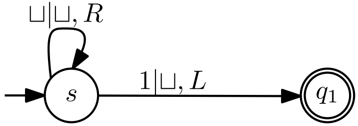


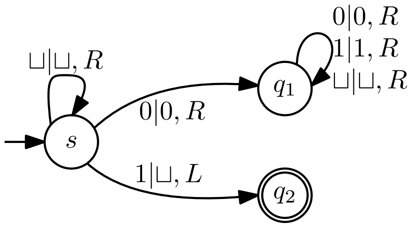

- Im Beispiel ist

$$f(w) = \begin{cases} v & \text{falls } w = 1v \\ \text{undefiniert} & \text{sonst} \end{cases}$$

# Bemerkungen zur TM

- Oft werden wir die Turing-Maschine beziehungsweise deren Übergangsfunktion nur unvollständig beschreiben.
- Beispiel:




- Eine Vervollständigung ist immer möglich.
- Wenn für eine bestimmte Kombination $q, a$ kein Übergang $\delta(q, a)$ definiert ist, dann stoppt die Turing-Maschine im Zustand $q$.

  (z.B. setze $\delta(q, a) := (q, a, N)$)

# Turing-Maschine Übersicht

(deterministische) **Turing-Maschine**

- Bestandteile:
  - Lese-/Schreibkopf ______ liest/schreibt Zeichen auf Band
  - endliche Kontrolle ______ stets in einem Zustand
  - unendliches Rechenband ______ enthält Eingabe
- formal: $\mathcal{M} = (Q, \Sigma, \sqcup, \Gamma \supseteq (\Sigma \cup \{\sqcup\}), s \in Q, \delta, F \subseteq Q)$
- Übergangsfunktion: $\delta(q, a) = (q', a', X \in \{L, N, R\})$

# Turing-Maschine Übersicht

(deterministische) **Turing-Maschine**

- Bestandteile:
  - Lese-/Schreibkopf ______ liest/schreibt Zeichen auf Band
  - endliche Kontrolle ______ stets in einem Zustand
  - unendliches Rechenband ______ enthält Eingabe
- formal: $\mathcal{M} = (Q, \Sigma, \sqcup, \Gamma \supseteq (\Sigma \cup \{\sqcup\}), s \in Q, \delta, F \subseteq Q)$
- Übergangsfunktion: $\delta(q, a) = (q', a', X \in \{L, N, R\})$

Bei der Bearbeitung einer Eingabe $w$ gibt es drei Möglichkeiten:

| | | |
|---|---|---|
| $\mathcal{M}$ "läuft" in einen Zustand in $F$. <br> $\rightsquigarrow \mathcal{M}$ akzeptiert $w$ | $\mathcal{M}$ "läuft" in einen Übergang $\delta(q, a) = (q, a, N)$. <br> $\rightsquigarrow \mathcal{M}$ lehnt $w$ ab | $\mathcal{M}$ "läuft" unendlich lange. <br> $\rightsquigarrow \mathcal{M}$ stoppt nicht |

# Definitionen zur TM

- Eine Turing-Maschine **akzeptiert** eine Eingabe $w \in \Sigma^*$, wenn sie nach Lesen von $w$ in einem Zustand aus $F$ stoppt.
- Sie **akzeptiert** eine Sprache $L$ genau dann, wenn sie ausschließlich Wörter $w \in L$ als Eingabe akzeptiert.
- Eine Sprache $L \subseteq \Sigma^*$ heißt **rekursiv** oder **entscheidbar**, wenn es eine Turing-Maschine gibt, die auf allen Eingaben stoppt und eine Eingabe $w$ genau dann akzeptiert, wenn $w \in L$ gilt.
- Eine Sprache $L \subseteq \Sigma^*$ heißt **rekursiv-aufzählbar** oder **semi-entscheidbar**, wenn es eine Turing-Maschine gibt, die genau die Eingaben $w$ akzeptiert für die $w \in L$.

  Das Verhalten der Turing-Maschine für Eingaben $w \notin L$ ist damit nicht genau definiert. D.h., die Turing-Maschine stoppt entweder nicht in einem Endzustand oder aber stoppt gar nicht.

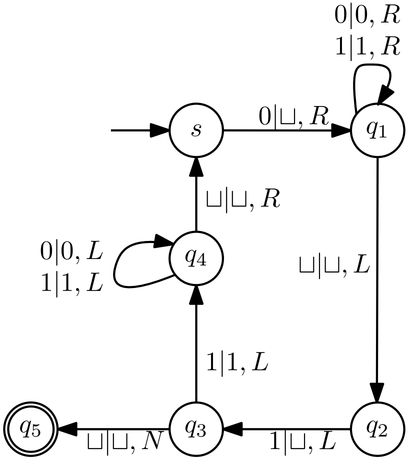
# Notation: Konfiguration

- Situation in der sich eine TM $\mathcal{M} := (Q, \Sigma, \Gamma, \delta, s, F)$ befindet, wird durch die Angabe der **Konfiguration** kodiert.

- Eine Konfiguration hat die Form $w(q)av$, wobei
  - $w, v \in \Gamma^*$
  - $a \in \Gamma$
  - $q \in Q$

- Bedeutung:
  - $\mathcal{M}$ befindet sich gerade im Zustand $q$.
  - Der Lese-/Schreibkopf steht auf dem Zeichen $a$.
  - Links vom Lese-/Schreibkopf steht das Wort $w$ auf dem Rechenband.
  - Rechts vom Lese-/Schreibkopf steht das Wort $v$ auf dem Rechenband.

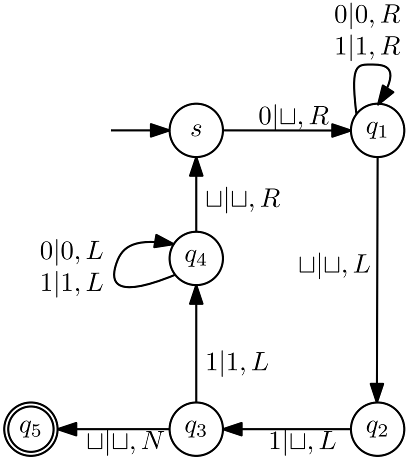
# Beispiel: Konfiguration




TM akzeptiert $\{0^n 1^n : n \geq 1\}$.

Beispiel: Eingabe $w = 0011$

# Beispiel: Konfiguration




$(s)0011$

$\sqcup(q_1)011$

$0(q_1)11$

$01(q_1)1$

$011(q_1)\sqcup$

$01(q_2)1$

$0(q_3)1\sqcup$

$(q_4)01$

$(q_4)\sqcup 01$

$\sqcup(s)01$

$\sqcup(q_1)1$

$1(q_1)\sqcup$

$(q_2)1$

$(q_3)\sqcup$

$(q_5)\sqcup$

TM akzeptiert $\{0^n1^n : n \geq 1\}$.

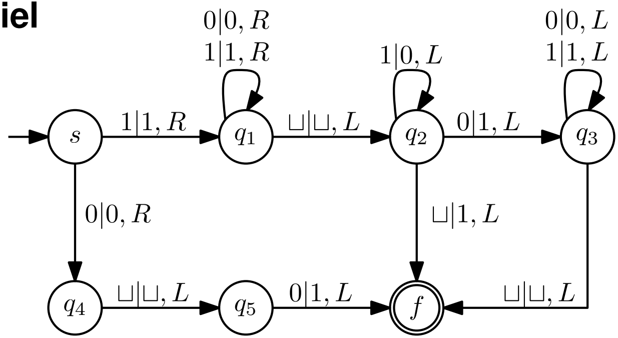
Beispiel: Eingabe $w = 0011$

# Definition: berechenbar / totalrekursiv

- Eine Funktion $f\colon \Sigma^* \longrightarrow \Gamma^*$ heißt **(Turing-)berechenbar** oder **totalrekursiv**, wenn es eine Turing-Maschine gibt, die bei Eingabe von $w \in \Sigma^*$ genau dann akzeptiert wenn $f(w)$ nicht undefiniert ist und in diesem Fall den Funktionswert $f(w) \in \Gamma^*$ ausgibt.
- Eine Turing-Maschine **realisiert** eine Funktion $f\colon \Sigma^* \longrightarrow \Gamma^*$, falls gilt:

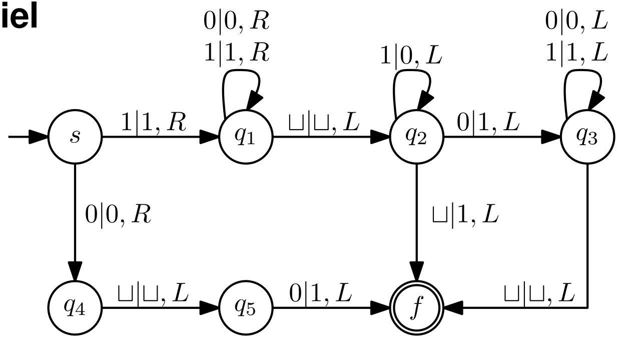
$$f(w) = \begin{cases} \text{Ausgabe der Turing-Maschine,} & \\ & \text{wenn sie bei Eingabe } w \text{ akzeptiert.} \\ \text{undefiniert,} & \\ & \text{sonst.} \end{cases}$$

## Beispiel



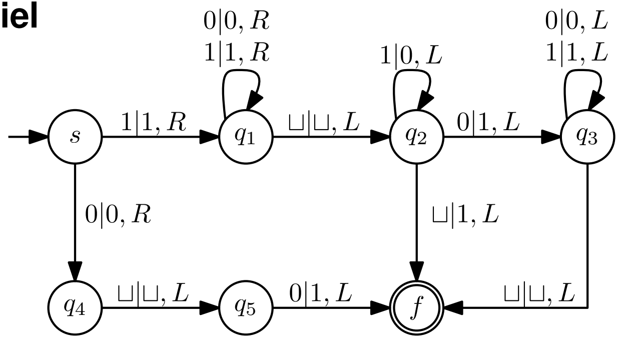

- Fasse die Eingabe $w$ als binäre Zahl auf.
- Es sollen nur Eingaben ohne führende Nullen und die Null selbst akzeptiert werden.
- Addiere zur Eingabe $w \in (0 \cup 1)^*$ eine Eins.

## Beispiel




Es gilt: $f(w) = \begin{cases} w + 1 & \text{falls } w \in 0 \cup 1(0 \cup 1)^*, \\ & w \text{ interpretiert als Binärzahl} \\ \text{undefiniert} & \text{sonst} \end{cases}$

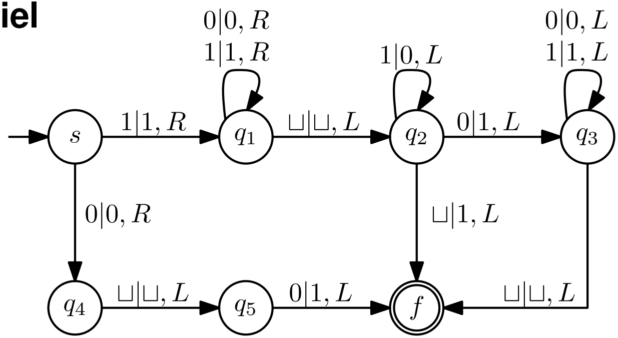
## Beispiel




Dabei sind die Zustände jeweils für die folgenden Aufgaben verantwortlich:

- $q_1$ Bewegung des Lese-/Schreibkopfes nach rechts bis zum Eingabeende,
- $q_2$ Bildung des Übertrages, der durch die Addition von Eins zu einer bereits vorhandenen Eins entsteht,

# Beispiel




Dabei sind die Zustände jeweils für die folgenden Aufgaben verantwortlich:

- $q_3$ Bewegung des Lese-/Schreibkopfes nach links, nachdem die Aufsummierung abgeschlossen ist (kein Übertrag mehr),
- $q_4, q_5$ Sonderbehandlung für den Fall der Eingabe 0, und
- $f$ Endzustand.

# Entscheidbarkeit und Berechenbarkeit

Entscheidbarkeit von Sprachen und Berechenbarkeit von Funktionen sind verwandt:

- Eine Turing-Maschine akzeptiert eine Sprache $L$, wenn sie genau auf den Eingaben $w \in L$ in einem ausgezeichneten Endzustand stoppt.
- $L$ ist entscheidbar, wenn es eine Turing-Maschine gibt, die auf allen Eingaben stoppt und $L$ akzeptiert.
- Die Funktion $f$ heißt berechenbar, wenn eine Turing-Maschine existiert, die $f$ realisiert.

# Entscheidbarkeit und Berechenbarkeit

Entscheidbarkeit von Sprachen und Berechenbarkeit von Funktionen sind verwandt:

- Man kann eine Turing-Maschine $\mathcal{M}$, die auf allen Eingaben stoppt, so modifizieren, dass es zwei ausgezeichnete Zustände $q_J$ und $q_N$ gibt und dass die modifizierte Turing-Maschine $\tilde{\mathcal{M}}$ stets in einem der Zustände $q_J$ oder $q_N$ hält und akzeptiert.
- Dabei stoppt $\tilde{\mathcal{M}}$ bei der Eingabe $w$ genau dann in $q_J$, wenn $\mathcal{M}$ das Wort $w$ akzeptiert, ansonsten stoppt $\tilde{\mathcal{M}}$ in $q_N$.
- Damit ist die Sprache $L$ genau dann entscheidbar, wenn es eine Turing-Maschine gibt, die immer in einem der Zustände $\{q_J, q_N\}$ stoppt, wobei sie gerade für $w \in L$ in $q_J$ hält.

# Entscheidbarkeit und Berechenbarkeit

Entscheidbarkeit von Sprachen und Berechenbarkeit von Funktionen sind verwandt:

- Man kann eine Turing-Maschine $\mathcal{M}$, die auf allen Eingaben stoppt, so modifizieren, dass es zwei ausgezeichnete Zustände $q_J$ und $q_N$ gibt und dass die modifizierte Turing-Maschine $\tilde{\mathcal{M}}$ stets in einem der Zustände $q_J$ oder $q_N$ hält und akzeptiert.
- Dabei stoppt $\tilde{\mathcal{M}}$ bei der Eingabe $w$ genau dann in $q_J$, wenn $\mathcal{M}$ das Wort $w$ akzeptiert, ansonsten stoppt $\tilde{\mathcal{M}}$ in $q_N$.

  **Testen Sie sich:** Finden Sie diese Modifikation $\tilde{\mathcal{M}}$ von $\mathcal{M}$?

- Damit ist die Sprache $L$ genau dann entscheidbar, wenn es eine Turing-Maschine gibt, die immer in einem der Zustände $\{q_J, q_N\}$ stoppt, wobei sie gerade für $w \in L$ in $q_J$ hält.

# Korollar

- Eine Sprache $L \subseteq \Sigma^*$ ist **entscheidbar** genau dann, wenn ihre **charakteristische Funktion** $\chi_L$ berechenbar ist, wobei gilt:

$$\chi_L : \Sigma^* \to \{0, 1\} \quad \text{mit} \qquad \chi_L(w) = \begin{cases} 1 & \text{falls } w \in L \\ 0 & \text{sonst} \end{cases}$$

- Eine Sprache $L$ ist **semi-entscheidbar** genau dann, wenn die partielle Funktion $\chi_L^*$ berechenbar ist, wobei gilt:

$$\chi_L^*(w) = \begin{cases} 1 & \text{falls } w \in L \\ \text{undefiniert} & \text{sonst} \end{cases}$$

# Die Church'sche These

**Church'sche These.**

Die Menge der (Turing-)berechenbaren Funktionen ist genau die Menge der im intuitiven Sinne überhaupt berechenbaren Funktionen.

# Die Church'sche These

**Church'sche These.**

Die Menge der (Turing-)berechenbaren Funktionen ist genau die Menge der im intuitiven Sinne überhaupt berechenbaren Funktionen.

## Interpretation

- Turing-Maschinen sind formale Modelle für Algorithmen.
- Kein Berechnungsverfahren kann algorithmisch genannt werden, wenn es nicht von einer Turing-Maschine ausführbar ist.

## Bemerkung

- Die Church'sche These ist ohne eine präzise Definition von intuitiv berechenbar nicht beweisbar.
- Sie ist aber in der Informatik allgemein akzeptiert.

# Die Church’sche These

**Church’sche These.**

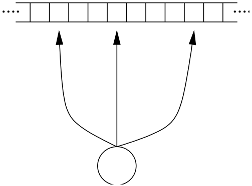
Die Menge der (Turing-)berechenbaren Funktionen ist genau die Menge der im intuitiven Sinne überhaupt berechenbaren Funktionen.

**Begründung**

- Es existieren keine Beispiele von Funktionen, die als intuitiv berechenbar angesehen werden, aber nicht Turing-berechenbar sind.
- Alle Versuche, realistische Modelle aufzustellen, die mächtiger sind als Turing-Maschinen, schlugen fehl.
- Eine Reihe von völlig andersartigen Ansätzen, den Begriff der Berechenbarkeit formal zu fassen, wie zum Beispiel die Registermaschine, haben sich als äquivalent erwiesen.

# Erweiterungen der Turing-Maschine

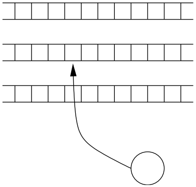
**Mehrere Lese-/Schreibköpfe**



- Mehrere Lese-/Schreibköpfe ($n \in \mathbb{N}$) greifen auf das eine Eingabeband zu und werden von der endlichen Kontrolle gesteuert.
- Die Übergangsfunktion ist dann vom Typ $\delta\colon\ Q \times \Gamma^n \to Q \times \Gamma^n \times \{L, N, R\}^n$.
- Die Zustände $q \in Q$ kann man als $n$-Tupel auffassen.
- Es ist nötig eine Prioritätenregel für die einzelnen Köpfe anzugeben, falls mehrere auf einem Feld des Eingabebandes stehen.

# Erweiterungen der Turing-Maschine

**Mehrere Bänder**



- Ein Lese-/Schreibkopf kann auf mehrere Eingabebänder ($n \in \mathbb{N}$) zugreifen.
- Die Übergangsfunktion ist dann vom Typ

$$\delta\colon Q \times \Gamma \times \{1, \ldots, n\} \to Q \times \Gamma \times \{L, N, R\} \times \{1, \ldots, n\}.$$

# Erweiterungen der Turing-Maschine

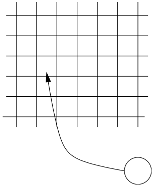
**Mehrere Lese-/Schreibköpfe für mehrere Bänder**

- Wir haben jetzt $m$ Bänder und $n$ Lese-/Schreibköpfe.
- Die Übergangsfunktion ist dann vom Typ

$$\delta\colon Q \times \Gamma^n \times \{1, \ldots, m\}^n \to Q \times \Gamma^n \times \{L, N, R\}^n \times \{1, \ldots, m\}^n.$$

# Erweiterungen der Turing-Maschine

**Mehrdimensionale Bänder**



- Das Eingabeband ist nun mehrdimensional und hat hier im Beispiel die Dimension zwei.
- Wir sprechen dann von einem Arbeitsfeld.
- Dabei ist

$$\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L(eft), U(p), R(ight), D(own), N(othing)\}$$

# Erweiterungen der Turing-Maschine

## Bemerkungen

- Fragestellungen der Art:
  - Wann stoppt eine Mehrkopf-Maschine?
  - Welcher Kopf ‚gewinnt', wenn mehrere Köpfe (verschiedene) Symbole an dieselbe Stelle schreiben wollen?

  müssen bei solchen Modifikationen noch geklärt werden.

- Es hat sich allerdings gezeigt, dass keine dieser Erweiterungen mehr leistet, als eine normale Turing-Maschine.

- **Alle angegebenen Modifikationen können durch eine normale $1$–Band Turing-Maschine simuliert werden.**

# Die universelle Turing-Maschine

**Ziel**

- Bisher: Nur Turing-Maschinen, die eine bestimmte Aufgabe erfüllen.
- Jetzt: Konstruktion einer Turing-Maschine, die als Eingabe
  - ein Programm und
  - eine spezielle Eingabe

  erhält.
- Die Aufgabe besteht darin, das gegebene Programm auf der gegebenen speziellen Eingabe auszuführen.

# Die universelle Turing-Maschine

Wir betrachten dazu eine normierte Turing-Maschine, d.h.

- $Q := \{q_1, \ldots, q_n\}$
- $\Sigma := \{a_1, \ldots, a_k\}$
- $\Gamma := \{\sqcup, a_1, \ldots, a_k, a_{k+1}, \ldots, a_l\}$
- $s := q_1$
- $F := \{q_2\}$

- Dies bedeutet keine Einschränkung in der Mächtigkeit der Turing-Maschinen:
  - Jede beliebige Turing-Maschine kann durch eine derart normierte Turing-Maschine der obigen Form simuliert werden.
  - Jede normierte Turing-Maschine $\mathcal{M}$ lässt sich eindeutig als Wort aus $(0 \cup 1)^*$ kodieren.

# Die Gödelnummer

Sei $\mathcal{M} := (Q, \Sigma, \sqcup, \Gamma, \delta, s, F)$ eine Turing-Maschine.

Die **Gödelnummer** von $\mathcal{M}$, bezeichnet als $\langle \mathcal{M} \rangle$, ist definiert durch folgende Kodierungsvorschrift:

- Kodiere Übergang

  $$\delta(q_i, a_j) = (q_r, a_s, d_t) \text{ durch } 0^i\ 1\ 0^j\ 1\ 0^r\ 1\ 0^s\ 1\ 0^t,$$

  wobei $d_t \in \{d_1, d_2, d_3\}$ und
  - $d_1$ für $L$,
  - $d_2$ für $R$ und
  - $d_3$ für $N$ steht.
- Die Turing-Maschine wird dann kodiert durch:

  $$111\text{code}_1 11\text{code}_2 11 \ldots 11\text{code}_z 111,$$

  wobei $\text{code}_i$ für $i = 1, \ldots, z$ alle Funktionswerte von $\delta$ in beliebiger Reihenfolge beschreibt.

# Die Gödelnummer - Bemerkungen

- Die eigentlichen Werte der Turing-Maschine werden also (unär) durch Nullen beschrieben und die Einsen dienen als Begrenzung der Eingabewerte.
- Jede Turing-Maschine kann also durch ein Wort aus $(0 \cup 1)^*$ kodiert werden.
- Umgekehrt beschreibt jedes Wort aus $(0 \cup 1)^*$ (höchstens) eine Turing-Maschine.
- Wir vereinbaren, dass ein Wort, das keine Turing-Maschine in diesem Sinne beschreibt, (zum Beispiel $\varepsilon$, 0, 000) eine Turing-Maschine kodiert, die $\emptyset$ akzeptiert.
- Eine universelle Turing-Maschine erhält als Eingabe ein Paar $(\langle \mathcal{M} \rangle, w)$, wobei $w \in \{0, 1\}^*$ ist, und sie simuliert $\mathcal{M}$ auf $w$.
- Wir verwenden ein Trennzeichen $\# \notin \{0, 1\}$. Die Eingabe für die universelle Turing-Maschine hat die Form $\langle \mathcal{M} \rangle \# w$.

# Die Gödelnummer - Beispiel

Sei $\mathcal{M} = (Q = \{q_1, q_2, q_3\}, \Sigma = \{0, 1\}, \sqcup, \Gamma = \{0, 1, \sqcup\}, \delta, q_1, \{q_2\})$, mit

$$
\begin{aligned}
\delta(q_1, 1) &= (q_3, 0, R)\\
\delta(q_3, 0) &= (q_1, 1, R)\\
\delta(q_3, 1) &= (q_2, 0, R)\\
\delta(q_3, \sqcup) &= (q_3, 1, L)
\end{aligned}
$$

$\mathcal{M}$ zusammen mit der Eingabe 1011 ist dann:

11101001000101001100010101001001100010010010100110001000100010010111#1011

# Die Gödelnummer - Beispiel

Sei $\mathcal{M} = (Q = \{q_1, q_2, q_3\}, \Sigma = \{0, 1\}, \sqcup, \Gamma = \{0, 1, \sqcup\}, \delta, q_1, \{q_2\})$, mit

$$\delta(q_1, 1) = (q_3, 0, R)$$
$$\delta(q_3, 0) = (q_1, 1, R)$$
$$\delta(q_3, 1) = (q_2, 0, R)$$
$$\delta(q_3, \sqcup) = (q_3, 1, L)$$

$\mathcal{M}$ zusammen mit der Eingabe 1011 ist dann:

111010010001010011000101010010011000100100101001100010001000100101111#1011

# Definition

Zu $w \in \{0,1\}^*$ sei $T_w$

- die Turing-Maschine mit der Gödelnummer $w$, bzw.
- die Turing-Maschine, die $\emptyset$ akzeptiert.

Es sei $L(T_w)$ die Sprache, die von $T_w$ akzeptiert wird.

# Die Diagonalsprache

Wir konstruieren die sogenannte **Diagonalsprache** $L_d$ , wie folgt:

- Betrachte die Wörter aus $\{0, 1\}^*$ in kanonischer Reihenfolge, d.h. $w_i$ steht vor $w_j$ $(i < j)$, falls
  - $|w_i| < |w_j|$, oder
  - $|w_i| = |w_j|$ und $w_i$ lexikographisch vor $w_j$ steht.
- $\mathcal{M}_j$ sei die TM, die durch die Gödelnummer $w_j$ kodiert ist.
- Wir konstruieren eine unendliche Tabelle,
  - an deren Position $(i, j)$ für $1 \leq i, j < \infty$ eine Null oder eine Eins steht, und
  - welche beinhaltet, ob $w_i$ in $L(\mathcal{M}_j)$ ist.
- Damit gilt für die Einträge

$$(i, j) = \begin{cases} 1 & \text{falls } \mathcal{M}_j \ w_i \text{ akzeptiert} \\ 0 & \text{sonst} \end{cases}$$

# Die Diagonalsprache

- Damit gilt für die Einträge

$$(i,j) = \begin{cases} 1 & \text{falls } \mathcal{M}_j \ w_i \text{ akzeptiert} \\ 0 & \text{sonst} \end{cases}$$

- Definiere dazu

$$L_d := \{w_i : \mathcal{M}_i \text{ akzeptiert } w_i \text{ nicht}\}.$$

- $L_d$ enthält also alle $w_i$, für die auf der Diagonalen an der Stelle $(i,i)$ eine Null steht.
- Dies führt später zu einem Diagonalbeweis (Cantor).

# Die Diagonalsprache - Veranschaulichung

- $(i, j) = 1$ falls $\mathcal{M}_j$ $w_i$ akzeptiert, 0 sonst
- $L_d := \{w_i : \mathcal{M}_i \text{ akzeptiert } w_i \text{ nicht}\}$
- $L_d$ enthält also alle $w_i$, für die auf der Diagonalen an der Stelle $(i, i)$ eine Null steht.

| $w \in \{0,1\}^*$ | Gödelnummer | $w_{123}$ | $w_{124}$ | $w_{125}$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\vdots$ | $\vdots$ | | | | | | | | |
| $w_{123}$ | 1 | **0** | 1 | 0 | 1 | 0 | 0 | $w_{123} \in L_d$ | $w_{123} = 1111011 \in L_d$ |
| $w_{124}$ | 0 | 0 | **1** | 0 | 0 | 1 | 1 | $w_{124} \notin L_d$ | $w_{124} = 1111100 \notin L_d$ |
| $w_{125}$ | 1 | 0 | 0 | **1** | 1 | 0 | 1 | $w_{125} \notin L_d$ | $w_{125} = 1111101 \notin L_d$ |
| $\vdots$ | $\vdots$ | | | | | | | | |

# Unentscheidbarkeit der Diagonalsprache

**Satz.**

Die Sprache $L_d$ ist nicht entscheidbar.

# Unentscheidbarkeit der Diagonalsprache

Karlsruher Institut für Technologie

**Satz.**

Die Sprache $L_d$ ist nicht entscheidbar.

**Beweis:**

Wäre $L_d$ entscheidbar, so existierte eine Turing-Maschine $\mathcal{M}$, die

- (1) bei jeder Eingabe hält,
- (2) genau die $w \in L_d$ akzeptiert.

Dann ist $\mathcal{M} = \mathcal{M}_i$ für (mindestens) ein $i$.

Wende nun $\mathcal{M}_i$ auf $w_i$ an:

- Falls $w_i \in L_d$, dann akzeptiert $\mathcal{M}$ das Wort $w_i$ wegen (2).
- Falls $w_i \in L_d$, dann akzeptiert $\mathcal{M}_i$ das Wort $w_i$ nicht, (laut der Definition von $L_d$)

Zusammen ist dies ein Widerspruch zu $\mathcal{M} = \mathcal{M}_i$.

# Unentscheidbarkeit der Diagonalsprache

Karlsruher Institut für Technologie

**Satz.**

Die Sprache $L_d$ ist nicht entscheidbar.

**Beweis:**

Wäre $L_d$ entscheidbar, so existierte eine Turing-Maschine $\mathcal{M}$, die

- (1) bei jeder Eingabe hält,
- (2) genau die $w \in L_d$ akzeptiert.

Dann ist $\mathcal{M} = \mathcal{M}_i$ für (mindestens) ein $i$.

Wende nun $\mathcal{M}_i$ auf $w_i$ an:

- Falls $w_i \notin L_d$, dann akzeptiert $\mathcal{M}$ das Wort $w_i$ nicht wegen (2).
- Falls $w_i \notin L_d$, dann akzeptiert $\mathcal{M}_i$ das Wort $w_i$, (laut der Definition von $L_d$)

Zusammen ist dies ein Widerspruch zu $\mathcal{M} = \mathcal{M}_i$.

# Korollar

**Korollar**

Die Sprache $L_d^c := \{0,1\}^* \backslash L_d$ ist nicht entscheidbar.

# Korollar

**Korollar**

Die Sprache $L_d^c := \{0,1\}^* \setminus L_d$ ist nicht entscheidbar.

**Beweis:**

- Wäre $L_d^c$ entscheidbar, so existierte eine Turing-Maschine, die $L_d^c$ entscheidet.
- Diese könnte aber leicht zu einer Turing-Maschine modifiziert werden, die $L_d$ entscheidet.
- Dies ist ein Widerspruch zur Unentscheidbarkeit der Diagonalsprache.