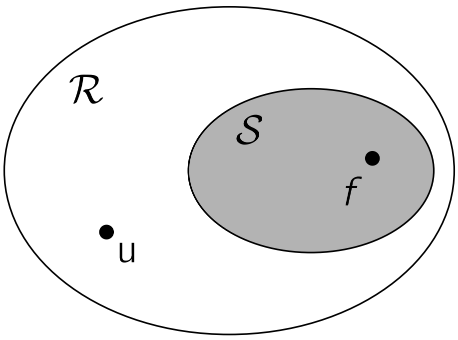
Vorlesung 8
Rekursive Aufzählbarkeit

Wdh.: Der Satz von Rice

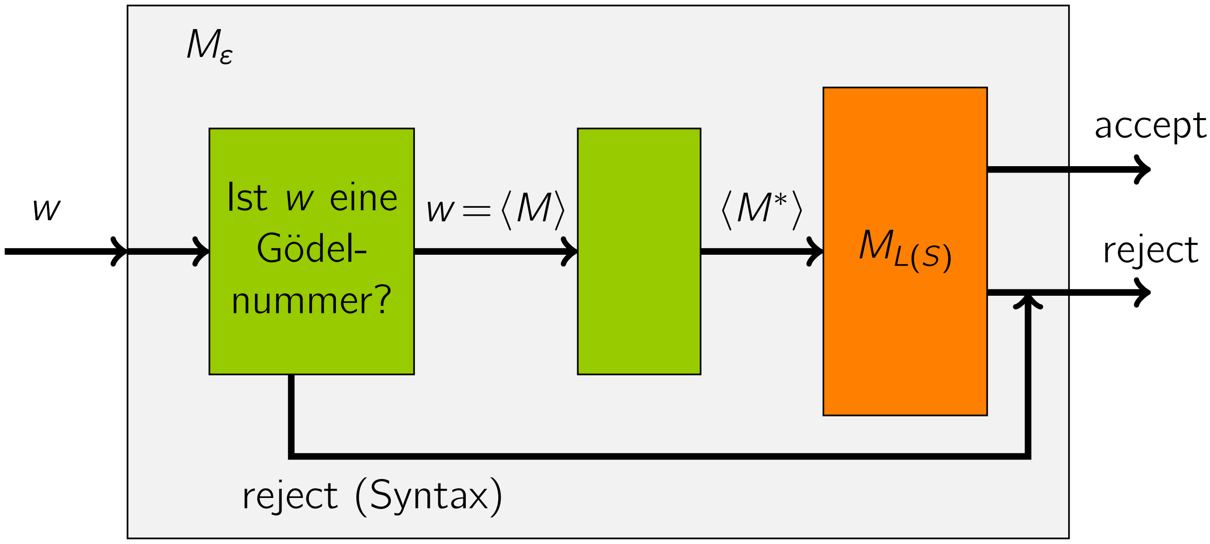
Satz
Sei $\mathcal{R}$ die Menge der von TMen berechenbaren partiellen Funktionen und $\mathcal{S}$ eine Teilmenge von $\mathcal{R}$ mit $\emptyset \subsetneq \mathcal{S} \subsetneq \mathcal{R}$. Dann ist die Sprache

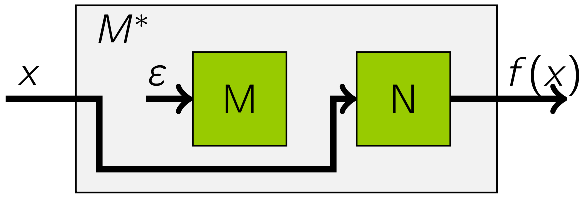
$$L(\mathcal{S}) = \{\langle M \rangle \mid M \text{ berechnet eine Funktion aus } \mathcal{S}\}$$

unentscheidbar.



Wdh.: Satz von Rice – Illustration





Wdh.: Satz von Rice – Anwendungsbeispiele

Beispiel 2

- Sei $\mathcal{S} = \{f_M \mid \forall w \in \{0,1\}^* \colon f_M(w) \neq \bot\}$.
- Dann ist

$$\begin{aligned} L(\mathcal{S}) &= \{\langle M \rangle \mid M \text{ berechnet eine Funktion aus } \mathcal{S}\} \\ &= \{\langle M \rangle \mid M \text{ hält auf jeder Eingabe}\} \end{aligned}$$

- Diese Sprache ist auch als das *allgemeine Halteproblem* H_{all} bekannt.
- Gemäß Satz von Rice ist H_{all} unentscheidbar.

Beispiel 5

- Sei $H_{42} = \{\langle M \rangle \mid$
  Auf jeder Eingabe hält M nach höchstens 42 Schritten$\}$.
- Über diese Sprache sagt der Satz von Rice nichts aus!
- Ist H_{42} entscheidbar?

Erkennen vs Entscheiden

Definition

Sei M eine Turingmaschine. Dann ist $L(M)$ die Menge der von M akzeptieren Wörter, also

$$L(M) = \{w \in \{0,1\}^* \mid M \text{ akzeptiert } w\}.$$

Wir nennen $L(M)$ die von M erkannte Sprache.

Achtung: Bei Eingabe eines Wortes $w \notin L(M)$ hält M entweder nicht an oder hält an und verwirft.

Beobachtung

$$L(M) = \{w \mid \text{das Wort } f_M(w) \text{ beginnt mit einer } 1\}.$$

Beachte

Wenn M auf allen Eingaben hält, dann ist $L(M)$ gerade die von M entschiedene Sprache.

Semi-Entscheidbarkeit

Eine Sprache L wird von einer TM M entschieden, wenn

- M auf jeder Eingabe hält, und
- M genau die Wörter aus L akzeptiert.

Eine Sprache L, für die eine TM existiert, die L entscheidet, wird als entscheidbar bezeichnet.

Eine Sprache L wird von einer TM M erkannt, wenn

- M jedes Wort aus L akzeptiert, und
- M kein Wort akzeptiert, das nicht in L enthalten ist.

Es ist also $L(M)$ gerade die von M erkannte Sprache.

Definition

Eine Sprache L, für die eine TM existiert, die L erkennt, wird als semi-entscheidbar bezeichnet.

Beispiel

Das Halteproblem

$$H = \{\langle M\rangle w \mid M \text{ hält auf } w\}.$$

ist unentscheidbar.

Behauptung

Das Halteproblem ist semi-entscheidbar.

Beweis

Die folgende TM M' erkennt H:
Erhält M' eine Eingabe der Form $\langle M\rangle w$, so

- simuliert M' die TM M mit Eingabe w, und
- akzeptiert, falls M auf w hält.

Syntaktisch inkorrekte Eingaben werden von M' verworfen. $\square$

Aufzähler, rekursive Aufzählbarkeit

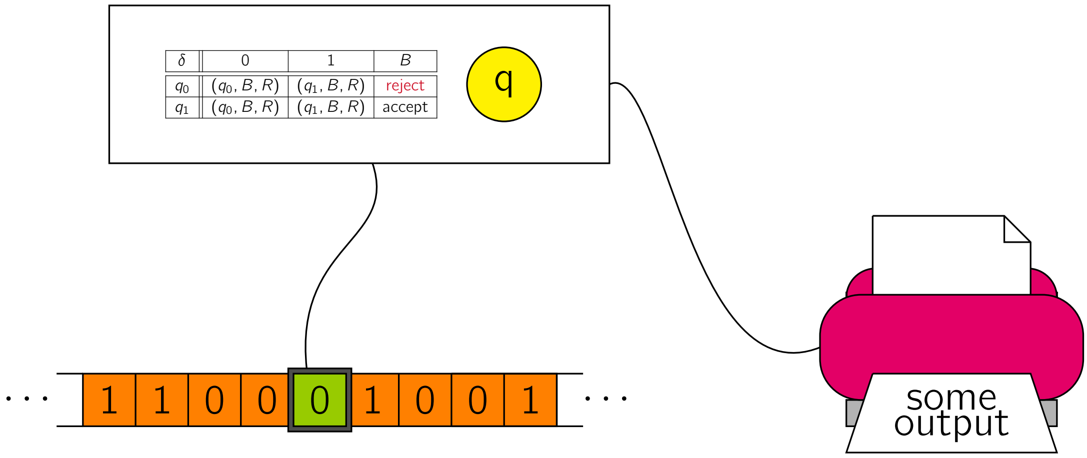
Ein Aufzähler für eine Sprache $L \subseteq \Sigma^*$ ist eine Variante einer TM mit einem angeschlossenen Drucker im Sinne eines zusätzlichen Ausgabebandes, auf dem sich der Kopf nur nach rechts bewegt.

- Gestartet mit leerem Arbeitsband, gibt der Aufzähler alle Wörter aus L (möglicherweise mit Wiederholungen) auf dem Drucker aus.
- Die ausgegebenen Wörter sind dabei durch ein Zeichen getrennt, das nicht in Σ enthalten ist.

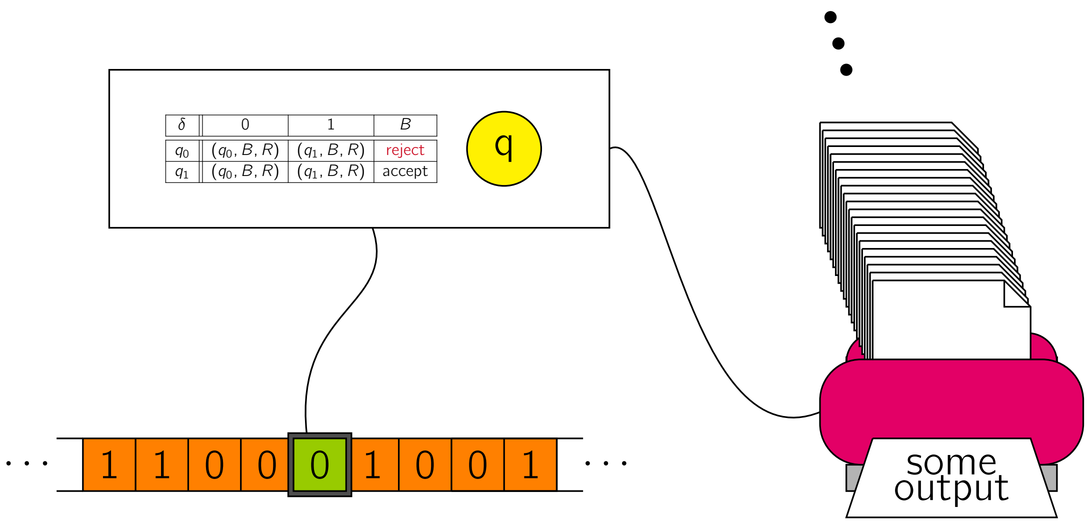
Definition

Eine Sprache, für die es einen Aufzähler gibt, heißt rekursiv aufzählbar.

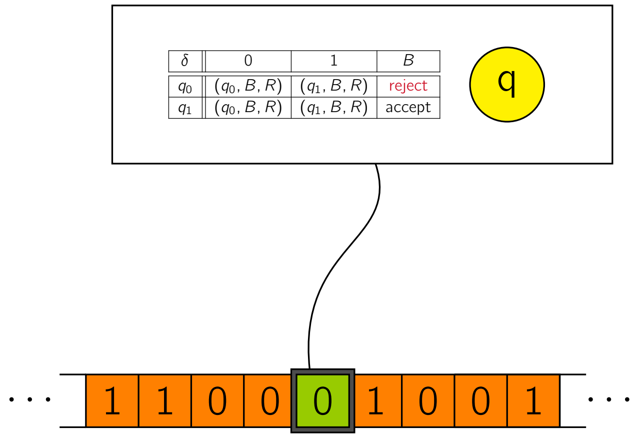
Aufzähler – Illustration



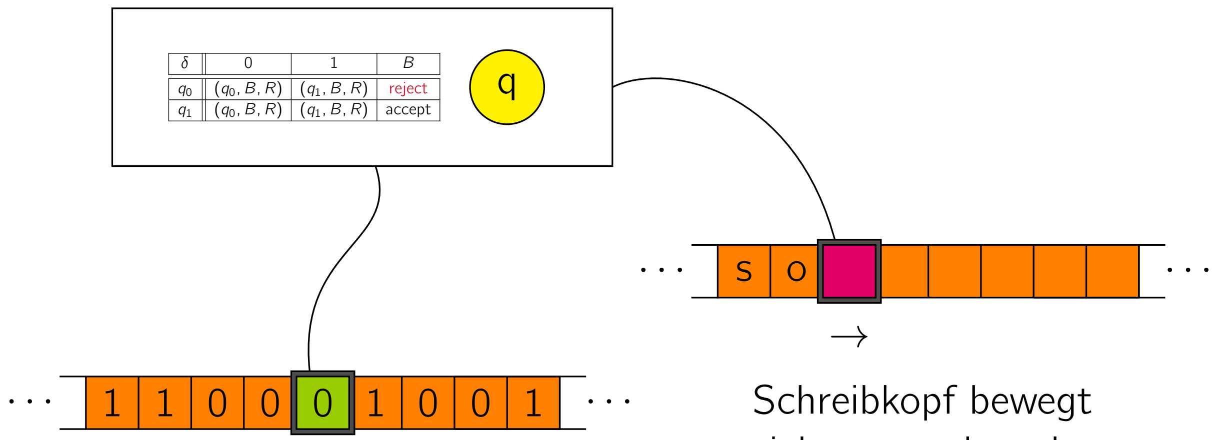
Aufzähler – Illustration



Aufzähler – Illustration

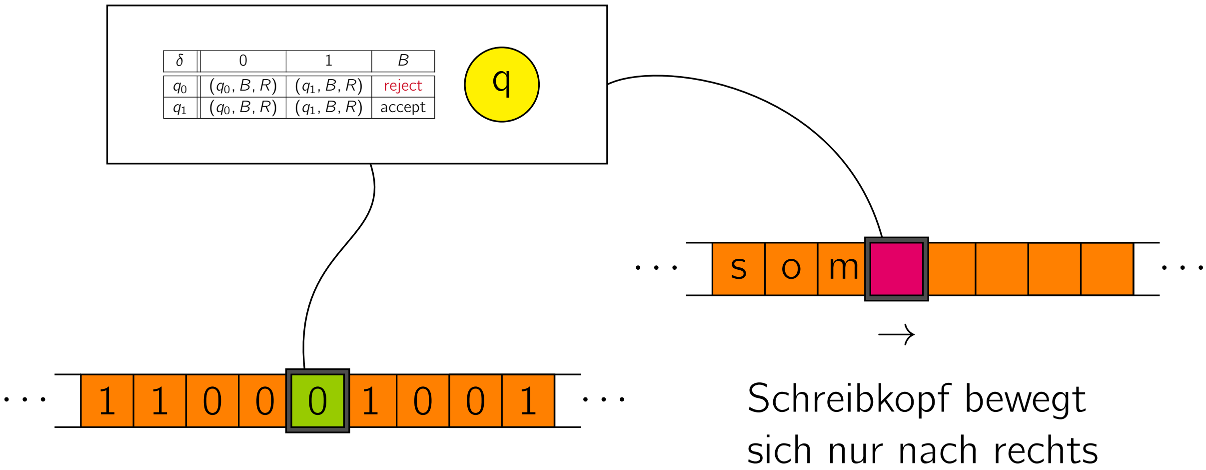


Aufzähler – Illustration

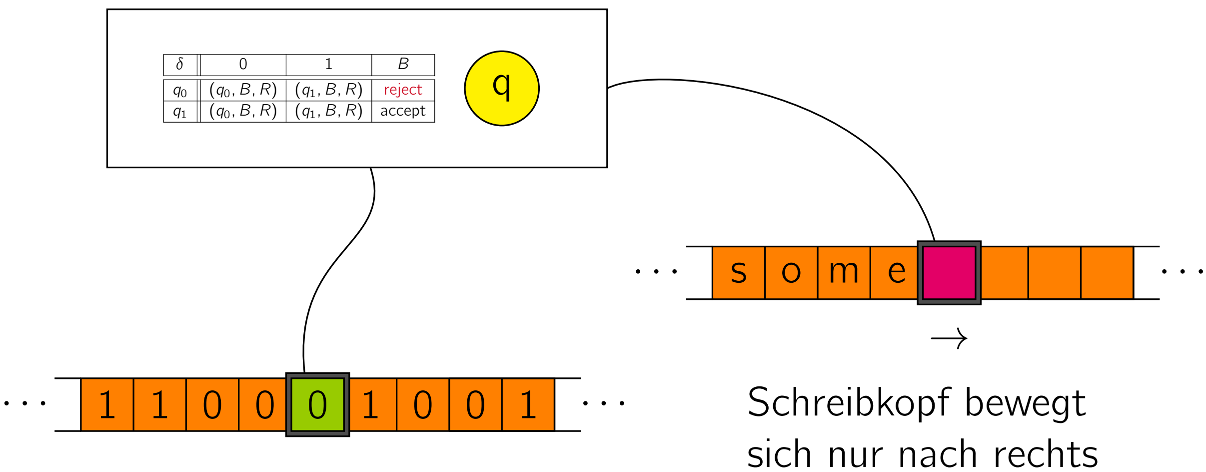


Schreibkopf bewegt
sich nur nach rechts

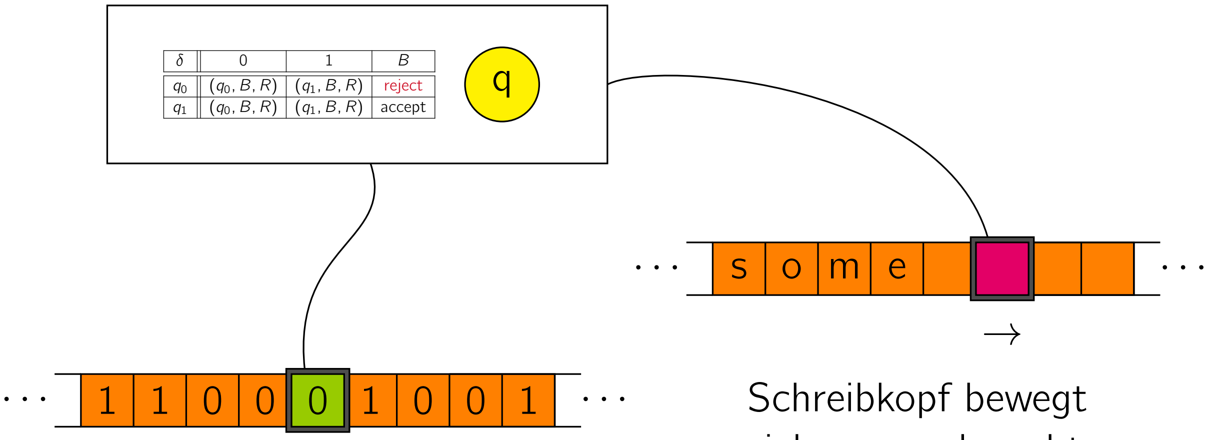
Aufzähler – Illustration



Aufzähler – Illustration



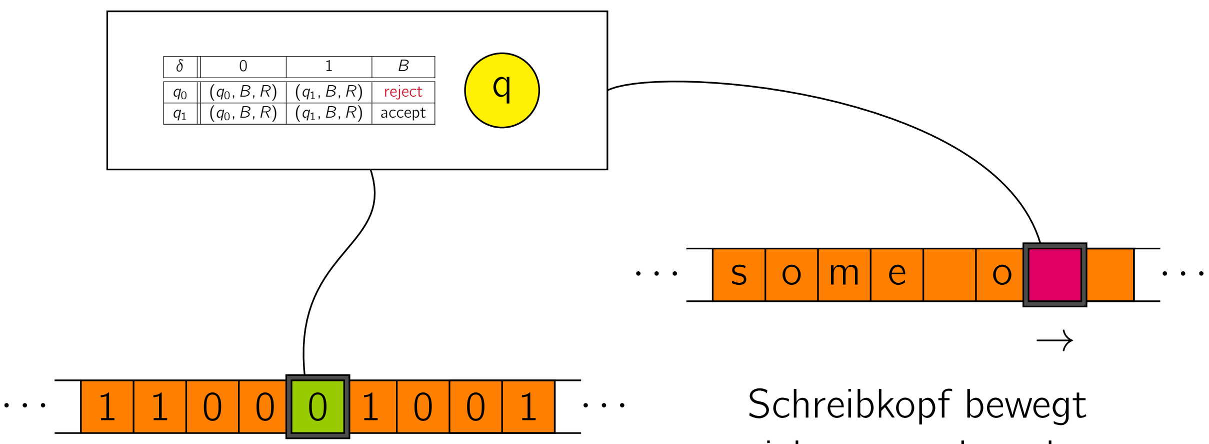
Aufzähler – Illustration



δ	0	1	B
q_0	(q_0, B, R)	(q_1, B, R)	reject
q_1	(q_0, B, R)	(q_1, B, R)	accept

Schreibkopf bewegt
sich nur nach rechts

Aufzähler – Illustration



δ	0	1	B
q_0	(q_0, B, R)	(q_1, B, R)	reject
q_1	(q_0, B, R)	(q_1, B, R)	accept

Schreibkopf bewegt
sich nur nach rechts

rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache L ist genau dann semi-entscheidbar, wenn sie rekursiv aufzählbar ist.

Beweis: L rekursiv aufzählbar $\Rightarrow$ L semi-entscheidbar

Sei A ein Aufzähler für L. Wir konstruieren eine TM M, die L erkennt.

Bei Eingabe w arbeitet M wie folgt:

- M simuliert A mit Hilfe einer Spur, welche die Rolle des Druckers übernimmt.
- Immer wenn ein neues Wort gedruckt worden ist, vergleicht M dieses Wort mit w und akzeptiert bei Übereinstimmung.

Korrektheit:

- Falls $w \in L$, so wird w irgendwann gedruckt und somit von M akzeptiert.
- Falls $w \notin L$, so wird w nicht gedruckt und somit nicht akzeptiert.

Beweis: L semi-entscheidbar $\Rightarrow$ L rekursiv aufzählbar

Sei M eine TM, die L erkennt. Wir konstruieren einen Aufzähler A für L.

Sei $x_0, x_1, x_2, \ldots$ die Aufzählung von Σ^* in kanonischer Reihenfolge.

Programm von A:

Für $i = 1, 2, 3, \ldots$

- Simuliere je i Schritte von M auf jedem Wort aus $\{x_0, \ldots, x_{i-1}\}$.
- Wann immer dabei eines der Worte akzeptiert wird, so drucke es aus.

Korrektheit:

Aufzähler A druckt offensichtlich nur Wörter aus L aus. Aber druckt er auch alle Wörter aus L aus?

- Sei x_k ein Wort aus L.
- Dann wird x_k von M nach einer endlichen Anzahl von Schritten t_k akzeptiert.
- Das heißt, x_k wird in jeder Iteration i mit $i \geq \max\{k, t_k\}$ von A ausgedruckt.

$\square$

Schnitte von Sprachen

Satz

a) *Wenn die Sprachen L_1 und L_2 entscheidbar sind, so ist auch die Sprache $L_1 \cap L_2$ entscheidbar.*

b) *Wenn die Sprachen L_1 und L_2 semi-entscheidbar sind, so ist auch die Sprache $L_1 \cap L_2$ semi-entscheidbar.*

Schnitte von Sprachen – Beweis a)

Seien M_1 und M_2 zwei TMen, die L_1 bzw. L_2 entscheiden.

TM M, die $L_1 \cap L_2$ entscheidet:

1. Bei Eingabe w simuliert M zunächst das Verhalten von M_1 auf w und dann das Verhalten von M_2 auf w.
2. Falls M_1 und M_2 das Wort w akzeptieren, so akzeptiert auch M; sonst verwirft M.

Korrektheit:

- Falls $w \in L_1 \cap L_2$, so wird w akzeptiert.
- Sonst wird w verworfen.

$\square$

Schnitte von Sprachen – Beweis b)

Seien nun M_1 und M_2 zwei TMen, die L_1 bzw. L_2 erkennen.

TM M, die $L_1 \cap L_2$ erkennt:

1. Bei Eingabe w simuliert M zunächst das Verhalten von M_1 auf w und dann das Verhalten von M_2 auf w.
2. Falls M_1 und M_2 akzeptieren, so akzeptiert auch M.

Wir verwenden die gleiche Konstruktion für M wie in a).

Korrektheit:

- Falls $w \in L_1 \cap L_2$, so wird w von M akzeptiert.
- Sonst wird w nicht akzeptiert.

$\square$

Vereinigungen von Sprachen

Satz

a) *Wenn die Sprachen L_1 und L_2 entscheidbar sind, so ist auch die Sprache $L_1 \cup L_2$ entscheidbar.*

b) *Wenn die Sprachen L_1 und L_2 semi-entscheidbar sind, so ist auch die Sprache $L_1 \cup L_2$ semi-entscheidbar.*

Vereinigungen von Sprachen – Beweis a)

Seien M_1 und M_2 zwei TMen, die L_1 bzw. L_2 entscheiden.

TM M, die $L_1 \cup L_2$ entscheidet

- Bei Eingabe w simuliert M zunächst das Verhalten von M_1 auf w und dann das Verhalten von M_2 auf w.
- Falls M_1 oder M_2 akzeptiert, so akzeptiert auch M. Sonst verwirft M.

Korrektheit:

- Falls $w \in L_1 \cup L_2$, so wird w von M_1 oder von M_2 und somit auch von M akzeptiert.
- Sonst verwerfen M_1 und M_2, so dass auch M verwirft.

$\square$

Vereinigungen von Sprachen – Beweis b)

Seien nun M_1 und M_2 zwei TMen, die L_1 bzw. L_2 erkennen.

Welches Problem tritt auf, wenn wir die Simulation aus a) einfach übernehmen?

Idee: Simuliere M_1 und M_2 parallel statt sequentiell.

Vereinigungen von Sprachen – Beweis b)

TM M, die $L_1 \cup L_2$ erkennt

- Wir nehmen o.B.d.A. an, dass M über zwei Bänder verfügt.
- Auf Band 1 wird M_1 auf w simuliert.
- Auf Band 2 wird M_2 auf w simuliert.
- Sobald ein Schritt erreicht wird, in dem M_1 oder M_2 akzeptiert, akzeptiert auch M.

Korrektheit:

- Falls $w \in L_1 \cup L_2$, so wird w von M_1 oder von M_2 und somit auch von M akzeptiert.
- Sonst wird w nicht akzeptiert.

$\square$

„2×semi-entscheidbar ⇔ entscheidbar“

Lemma ∗

Seien $L \subseteq \Sigma^*$ *und* $\overline{L} := \Sigma^* \setminus L$ *semi-entscheidbar. Dann ist* L *entscheidbar.*

Beweis

Seien M und $\overline{M}$ Maschinen, die L bzw. $\overline{L}$ erkennen.

Die TM M' entscheidet L durch eine parallele Simulation von M und $\overline{M}$ auf der Eingabe w:

- M' akzeptiert w, sobald M akzeptiert.
- M' verwirft w, sobald $\overline{M}$ akzeptiert.

Da entweder $w \in L$ oder $w \notin L$, tritt eines der obigen Ereignisse nach endlicher Zeit ein, so dass die Terminierung von M' sichergestellt ist. □

Komplemente von Sprachen

Beobachtung 1

Wenn die Sprache L entscheidbar ist, so ist auch $\overline{L}$ *entscheidbar, da wir das Akzeptanzverhalten einer TM M, die M entscheidet, invertieren können.*

Beobachtung 2

Die Menge der semi-entscheidbaren Sprachen ist hingegen nicht unter Komplementbildung abgeschlossen.

Beispiel

- H ist semi-entscheidbar.
- Wäre $\overline{H}$ ebenfalls semi-entscheidbar, so wäre H nach Lemma ∗ entscheidbar.
- Also ist $\overline{H}$ nicht semi-entscheidbar.

Schlussfolgerung

Korollar

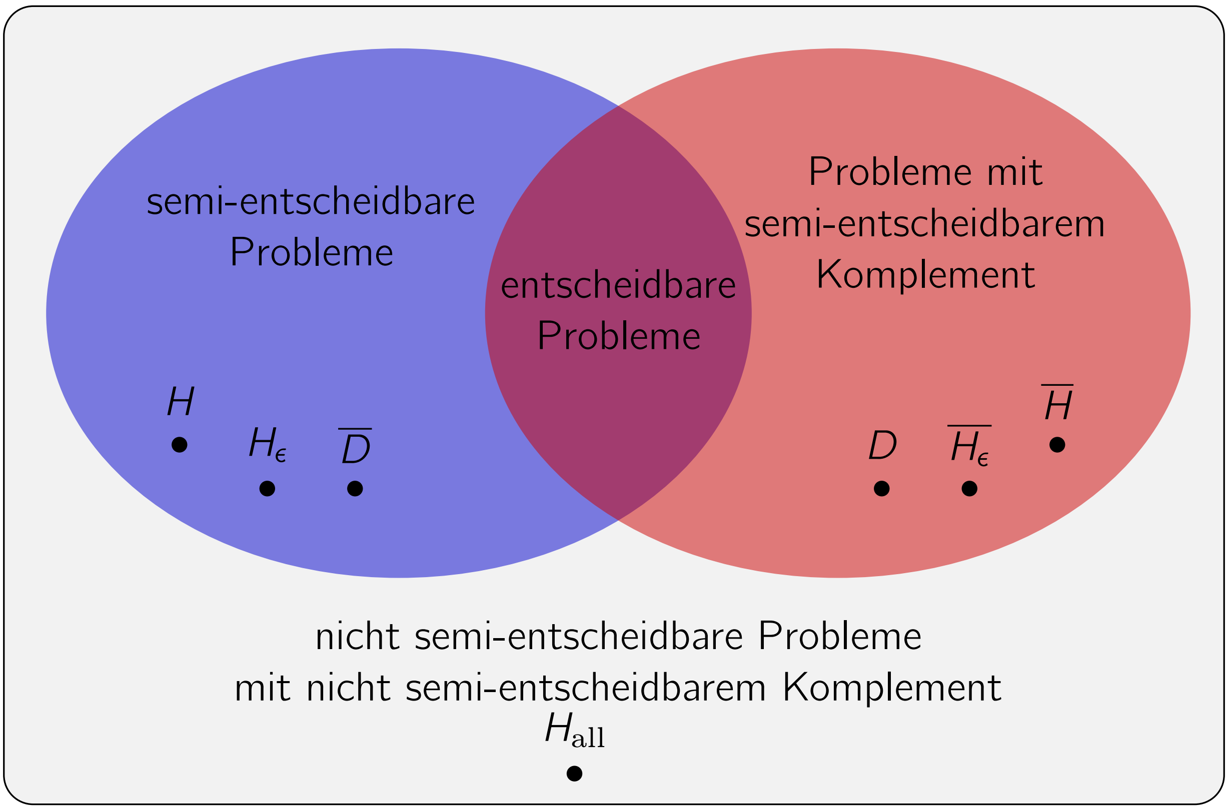
Für jede Sprache L gilt genau eine der vier folgenden Eigenschaften.

1. *L ist entscheidbar und sowohl L als auch $\overline{L}$ sind semi-entscheidbar;*
2. *L ist semi-entscheidbar, aber $\overline{L}$ ist nicht semi-entscheidbar;*
3. *$\overline{L}$ ist semi-entscheidbar, aber L ist nicht semi-entscheidbar;*
4. *sowohl L als auch $\overline{L}$ sind nicht semi-entscheidbar*

Beispiele

- Kategorie 1: Graphzusammenhang, Hamiltonkreis
- Kategorie 2: H, H_ϵ, $\overline{D}$
- Kategorie 3: $\overline{H}$, $\overline{H_\epsilon}$, D,
- Kategorie 4: $H_{\text{all}} = \{\langle M \rangle \mid M \text{ hält auf jeder Eingabe}\}$

Berechenbarkeitslandschaft



Allgemeines Halteproblem

Das *allgemeine Halteproblem* ist definiert als

$$H_{\text{all}} \;=\; \{\langle M\rangle \mid M \text{ hält auf jeder Eingabe}\}$$

Wie kann man nachweisen, dass sowohl H_{all} als auch $\overline{H}_{\text{all}}$ nicht semi-entscheidbar sind?

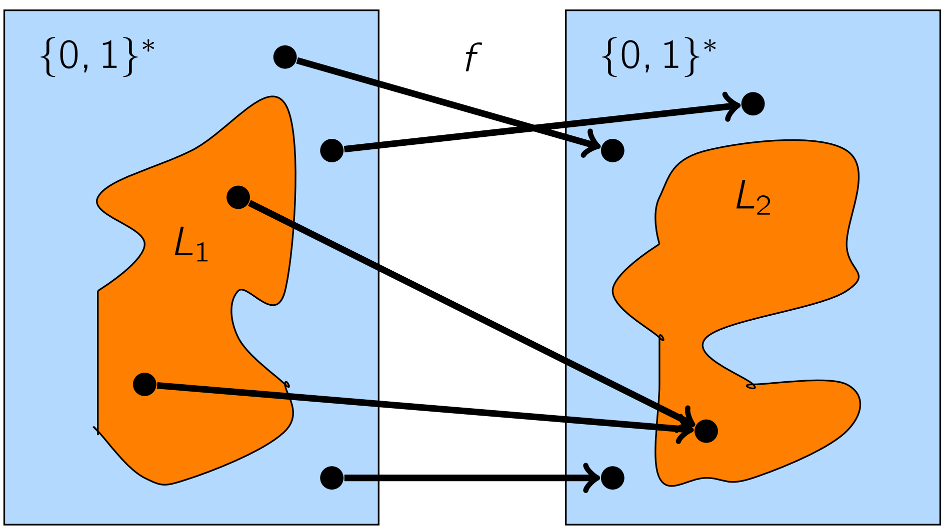
Wir verwenden eine spezielle Variante der Unterprogrammtechnik, die sogenannte Reduktion.

Reduktionen

Definition

Es seien L_1 und L_2 Sprachen über einem Alphabet Σ. Dann heißt L_1 auf L_2 reduzierbar, Notation $L_1 \leq L_2$, wenn es eine berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ gibt, so dass für alle $x \in \Sigma^*$ gilt

$$x \in L_1 \;\Leftrightarrow\; f(x) \in L_2 .$$



Reduktionen

Lemma

Falls $L_1 \leq L_2$ und L_2 semi-entscheidbar ist, so ist L_1 semi-entscheidbar.

Beweis

Wir konstruieren eine TM M_1, die L_1 erkennt, indem als Unterprogramm eine TM M_2 verwenden, die L_2 erkennt:

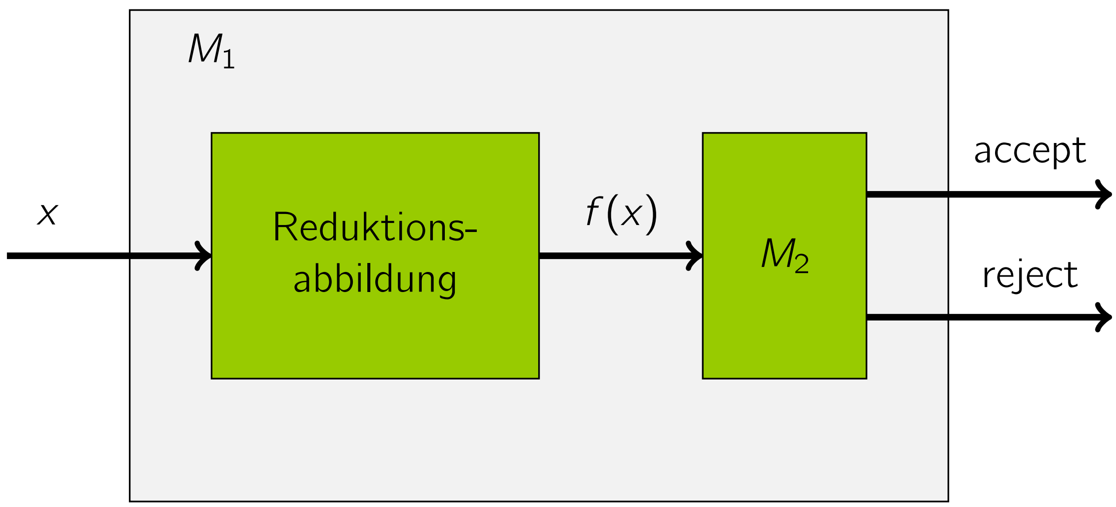
- Die TM M_1 berechnet $f(x)$ auf ihrer Eingabe x. (Dies ist möglich da f berechenbar ist.)
- Dann simuliert M_1 die TM M_2 mit der Eingabe $f(x)$ und übernimmt das Akzeptanzverhalten.

Korrektheit:

$$M_1 \text{ akzeptiert } x \iff M_2 \text{ akzeptiert } f(x) \iff f(x) \in L_2 \iff x \in L_1.$$

□

Reduktionen – Illustration



Verwendung von Reduktionen in umgekehrter Richtung

Im Umkehrschluss gilt:

Korollar

Falls $L_1 \leq L_2$ und L_1 nicht semi-entscheidbar ist, so ist L_2 nicht semi-entscheidbar.

Anwendung

Vorbeobachtung: H_ϵ ist unentscheidbar, aber semi-entscheidbar. Folglich ist $\overline{H_\epsilon}$ nicht semi-entscheidbar.

Wir werden zeigen

Behauptung A

$$\overline{H_\epsilon} \leq \overline{H_{\text{all}}}$$

Behauptung B

$$\overline{H_\epsilon} \leq H_{\text{all}}$$

Aus diesen Reduktionen folgt:

Satz

Weder $\overline{H_{all}}$ noch H_{all} ist semi-entscheidbar.