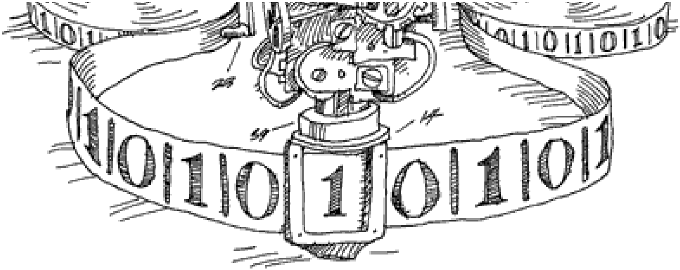
Theoretische Grundlagen der Informatik

Tutorium 13

Institut für Theoretische Informatik



Definitionen zur TM (Vorlesung)

Karlsruher Institut für Technologie

1. Eine TM *akzeptiert* eine Eingabe $w \in \Sigma^*$, wenn sie nach Lesen von w in einem Zustand aus F stoppt.
2. Sie *akzeptiert* eine Sprache $L \subseteq \Sigma^*$, wenn sie genau die Wörter w aus L als Eingabe akzeptiert.

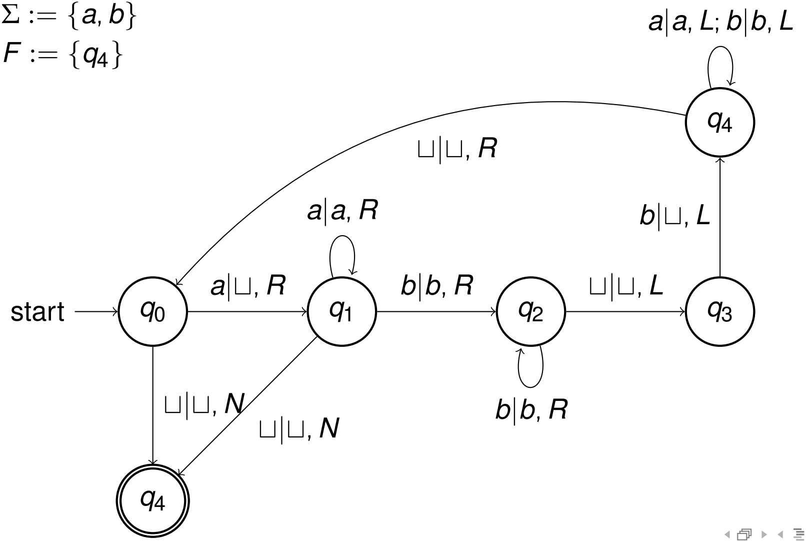
Definitionen zur TM (Vorlesung)

3. Eine Sprache $L \subseteq \Sigma^*$ heißt *rekursiv* oder *entscheidbar*, wenn es eine Turingmaschine gibt, die auf allen Eingaben stoppt und ein Wort $w \in \Sigma^*$ genau dann akzeptiert, wenn $w \in L$ gilt.
4. Eine Sprache $L \subseteq \Sigma^*$ heißt *rekursiv-aufzählbar* oder *semi-entscheidbar*, wenn es eine Turingmaschine gibt, die ein Wort $w \in \Sigma^*$ genau dann akzeptiert, wenn $w \in L$ gilt.
   Das Verhalten der Turingmaschine für Eingaben $w \notin L$ ist damit nicht definiert. Sie stoppt entweder nicht in einem Endzustand oder aber stoppt gar nicht.
5. Eine TM *realisiert* die Funktion $f : \Sigma^* \to \Gamma^*$ mit

$$f(w) = \begin{cases} \text{Ausgabe der TM nach Abarbeitung von } w & \text{wenn die TM hält} \\ \text{undefiniert} & \text{sonst} \end{cases}$$

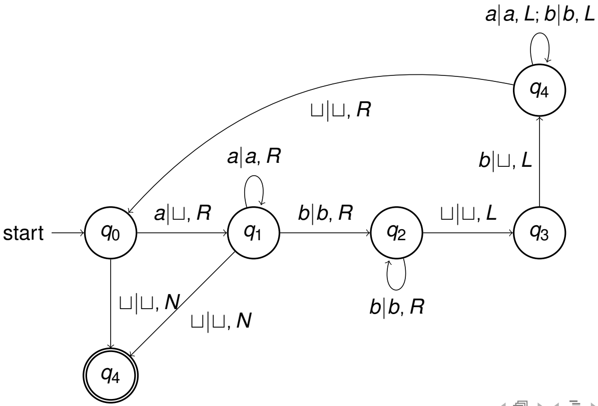
Beispiel zur Akzeptanz

- $\Sigma := \{a, b\}$
- $F := \{q_4\}$



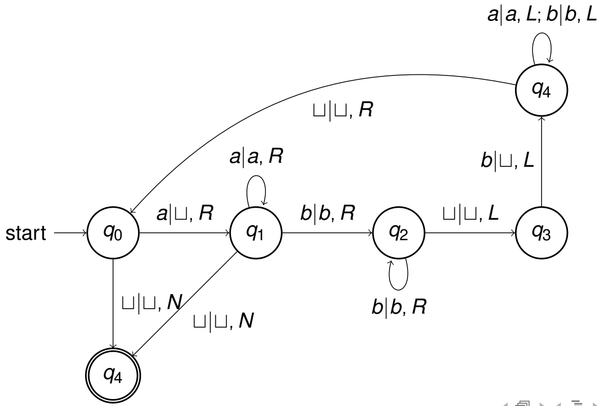
Sprache

- *aab* wird von der TM akzeptiert.
- *abb* nicht.
- Die akzeptierte Sprache ist $L(TM) := \{a^k b^l \mid k \geq l\}$



Sprache

- *aab* wird von der TM akzeptiert.
- *abb* nicht.
- Die akzeptierte Sprache ist $L(TM) := \{a^k b^l \mid k \geq l\}$



Halteproblem

. Das Halteproblem beschreibt die Aufgabe zu entscheiden, ob eine Turingmaschine bei gegeben Eingabewort hält oder nicht. Dieses Problem ist im allgemeinen Fall semi-entscheidbar, aber nicht entscheidbar.

- Formal: $(\langle M \rangle, w) \in HALT \Leftrightarrow M$ hält bei der Eingabe w

Aufgabe zur Entscheidbarkeit

Sei L eine nicht entscheidbare Sprache über dem Alphabet $\Sigma = \{0, 1\}$ und sei $(01)^k \notin L$ für jedes $k \in \mathbb{N}_{>0}$.
Zeige: Die Sprache

$$L' = \{w_1 \# w_2 \mid (w_1 \in L \land w_2 \notin L) \text{ oder } (w_1 \notin L \land w_2 \in L)\}$$

über dem Alphabet $\Sigma' = \{0, 1, \#\}$ ist nicht entscheidbar.

Aufgabe zur Entscheidbarkeit

Sei L eine nicht entscheidbare Sprache über dem Alphabet $\Sigma = \{0, 1\}$ und sei $(01)^k \notin L$ für jedes $k \in \mathbb{N}_{>0}$.
Zeige: Die Sprache

$$L' = \{w_1 \# w_2 \mid (w_1 \in L \wedge w_2 \notin L) \text{ oder } (w_1 \notin L \wedge w_2 \in L)\}$$

über dem Alphabet $\Sigma' = \{0, 1, \#\}$ ist nicht entscheidbar.

Annahme: L' ist entscheidbar. Sei M' dann eine Turingmaschine, die L' entscheidet. M' entscheidet also zunächst, ob der erste Teil des Wortes in L liegt, und dann, ob der zweite Teil des Wortes nicht in L liegt.
Sei M eine Turingmaschine, die die erste Phase von M' simuliert. Dann entscheidet M die Sprache L, was ein Widerspruch zur Annahme ist.

Aufgabe zur Entscheidbarkeit

Sei L eine nicht entscheidbare Sprache über dem Alphabet $\Sigma = \{0, 1\}$ und sei $(01)^k \notin L$ für jedes $k \in \mathbb{N}_{>0}$.
Zeige: Die Sprache

$$L' = \{w_1 \# w_2 \mid (w_1 \in L \land w_2 \notin L) \text{ oder } (w_1 \notin L \land w_2 \in L)\}$$

über dem Alphabet $\Sigma' = \{0, 1, \#\}$ ist nicht entscheidbar.

Annahme: L' ist entscheidbar. Sei M' dann eine Turingmaschine, die L' entscheidet. M' entscheidet also zunächst, ob der erste Teil des Wortes in L liegt, und dann, ob der zweite Teil des Wortes nicht in L liegt.
Sei M eine Turingmaschine, die die erste Phase von M' simuliert. Dann entscheidet M die Sprache L, was ein Widerspruch zur Annahme ist.

NOPE

Aufgabe zur Entscheidbarkeit

Sei L eine nicht entscheidbare Sprache über dem Alphabet $\Sigma = \{0, 1\}$ und sei $(01)^k \notin L$ für jedes $k \in \mathbb{N}_{>0}$.
Zeige: Die Sprache

$$L' = \{w_1 \# w_2 \mid (w_1 \in L \land w_2 \notin L) \text{ oder } (w_1 \notin L \land w_2 \in L)\}$$

über dem Alphabet $\Sigma' = \{0, 1, \#\}$ ist nicht entscheidbar.

Annahme: L' ist entscheidbar. Sei M' eine TM, die L' entscheidet. Für ein Wort $w \in \Sigma^*$ konstruiere das Wort $w' = w\#01$. Dann gilt:
$w' \in L' \Leftrightarrow w \in L$.
Damit kann man L wie folgt entscheiden, im Widerspruch zur Nichtentscheidbarkeit von L:
Simuliere die TM M' auf Eingabe w' und akzeptiere $w \in L$ gdw. M' $w' \in L'$ akzeptiert.

Beispielaufgabe Entscheidbarkeit (B5 A4)

Zeigen Sie, dass die Sprache
$\mathcal{L} = \{\langle \mathcal{M} \rangle \mid \text{Turingmaschine } \mathcal{M} \text{ akzeptiert jede Eingabe}\}$
nicht entscheidbar ist!

Aufgabe B6 A1

Zeigen Sie, dass die Sprache
$\mathcal{L} =$
$\{\langle \mathcal{M} \rangle \mid \text{Turingmaschine } \mathcal{M} \text{ hat mindestens einen unerreichbaren Zustand}\}$
nicht entscheidbar ist!

Aufgabe B6 A3 rekursiv aufzählbare Mengen

Karlsruher Institut für Technologie

Welche der folgenden Mengen sind rekursiv aufzählbar?
Beweisen Sie Ihre Aussage!

1. $M_1 := \{q \in \mathbb{Q} \mid 0 < q < 1\}$
2. $M_2 := \{r \in \mathbb{R} \mid 0 < r < 1\}$

Satz von Rice

Der Satz von Rice sagt aus, dass es unmöglich ist, eine nichttriviale Eigenschaft von Turingmaschinen algorithmisch zu entscheiden.

Formale Version

Es sei R die Klasse aller Turing-berechenbaren Funktionen und S eine beliebige nichttriviale (das bedeutet $S \neq \emptyset$ und $S \neq R$) Teilmenge davon. Dann ist die Sprache

$$C_S = \{M \mid M \text{ realisiert eine Funktion aus } S\}$$

unentscheidbar.

Beispiel

Die Klasse aller Programme, die etwas auf einem Rechner tun, das der Benutzer nicht möchte (das ist eine Teilmenge der turing-berechenbaren Funktionen) ist unentscheidbar. Daraus folgt, dass es keinen perfekten Virenscanner geben kann!

Aufgabe: Entscheidbarkeit

1. Zeige, dass die Sprache

$$L_\emptyset := \{\langle \mathcal{M} \rangle \mid \mathcal{M} \text{ Turingmaschine}, L(\mathcal{M}) = \emptyset\}$$

   nicht entscheidbar ist.

Aufgabe

Karlsruher Institut für Technologie

Kann man die Entscheidbarkeit der folgenden Mengen mithilfe des Satzes von Rice bestimmen?

- Alle Turingmaschinen, die bei Eingabe des leeren Wortes nur 0 aufs Band schreiben.
- Alle Turingmaschinen, die im ersten Schritt genau eine 0 aufs Band schreiben und im zweiten Schritt anhalten.
- Alle Turingmaschinen, die bei Eingabe des leeren Wortes das Band irgendwann einmal verändern.

Klausur 0405 Aufgabe 3 b

(Siehe Klausur.)

Klausur 0304 Aufgabe 3 b

(Siehe Klausur.)

Beliebige Klausuraufgaben

(Siehe Klausur.)

Bis zum nächsten Mal!

Lizenzen