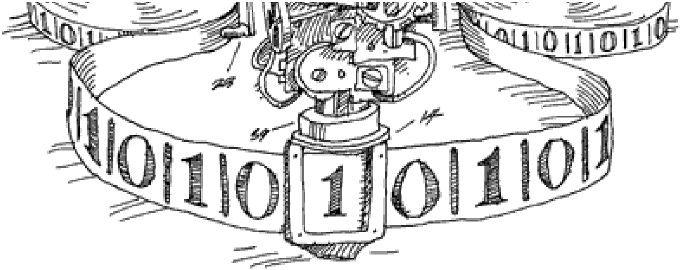
Theoretische Grundlagen der Informatik

Tutorium 4

Institut für Theoretische Informatik



Definitionen zur TM (Vorlesung)

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.

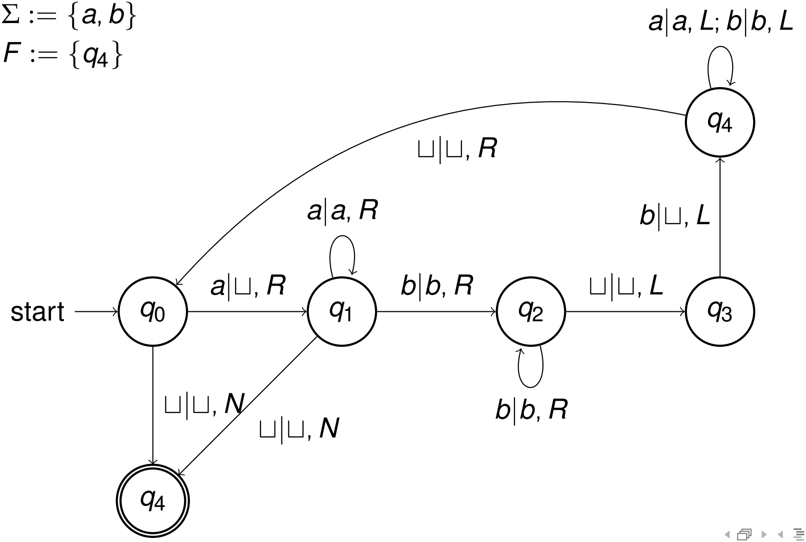
Definitionen zur TM (Vorlesung)

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

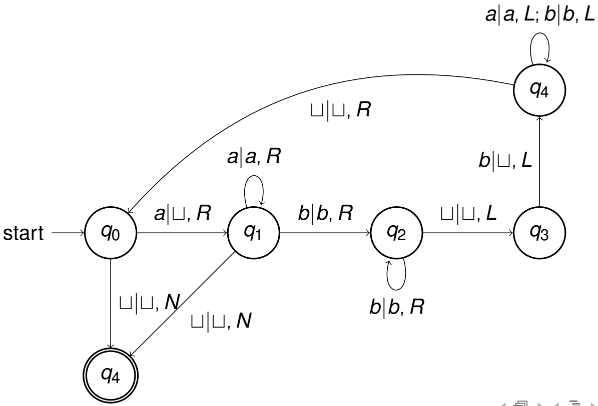
- $\Sigma := \{a, b\}$
- $F := \{q_4\}$



Sprache

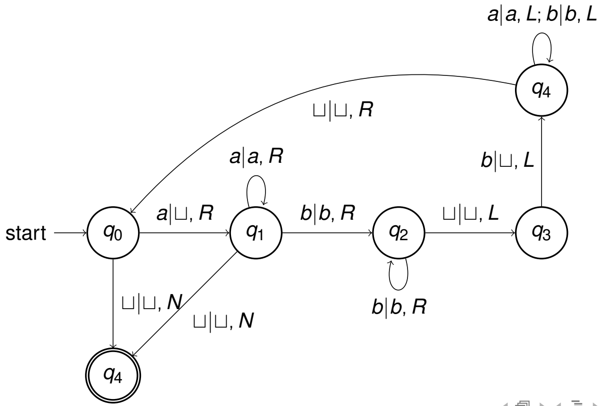
Karlsruher Institut für Technologie

- *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 $L(TM) := \{a^k b^l \mid k \geq l\}$



Universelle Turingmaschine

Karlsruher Institut für Technologie

Gödelnummer

- Jede TM lässt sich eindeutig als Zahl, ihre *Gödelnummer*, kodieren.
- Ungültige Gödelnummer $\triangleq$ TM, die alle Eingaben ablehnt.

Universelle Turingmaschine

- Eingabe:
  1. Kodierung einer TM
  2. Eingabe für die zu simulierende TM
- Simulation der übergebenen TM
- Ausgabe: Ausgabe der simulierten TM.

Aufgaben zur Entscheidbarkeit

Karlsruher Institut für Technologie

- Seien L_1 und L_2 zwei Sprachen über einem Alphabet Σ. Beweise:
  1. Ist L_1 entscheidbar, so ist auch L_1^c entscheidbar.
  2. Sind L_1 und L_2 entscheidbar, so ist auch $L_1 \cup L_2$ entscheidbar.
  3. Sind L_1 und L_2 entscheidbar, so ist auch $L_1 \setminus L_2$ entscheidbar.
  4. Ist L_1 entscheidbar, so ist auch die Sprache

     $$\min(L_1) := \{x \in L_1 \,|\, \text{kein echtes Präfix von } x \text{ ist in } L_1\}$$

     entscheidbar.
- Ist die Sprache der Busy-Beaver-Turingmaschinen mit maximal 42 Zuständen entscheidbar?

Aufgabe zur Semi-Entscheidbarkeit

Sei L eine semi-entscheidbare Sprache, die nicht entscheidbar ist.
Ist die Sprache

$$L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$$

entscheidbar, semi-entscheidbar oder keins von beiden?

Hinweis: Nimm an, L' sei semi-entscheidbar, und folgere daraus, dass dann L entscheidbar ist.

Aufgabe: Halteproblem

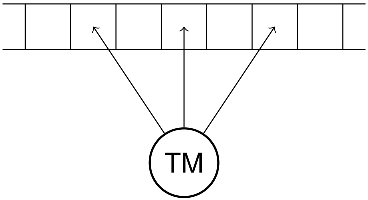
Zeige, dass das Halteproblem semi-entscheidbar ist.

Erweiterungen von Turingmaschinen

Erweiterungen

Es gibt mehrere zu Turingmaschinen (bezüglich Berechenbarkeit) äquivalente Berechnungsmodelle, die der Turingmaschine sehr ähnlich sind. Man spricht hier von Erweiterungen von Turingmaschinen. Diese verwendet man gerne in Beweisen, da sie oft übersichtlicher sind.

Mehrkopf-Turingmaschinen

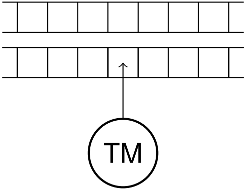


Eine Mehrkopf-Turingmaschine hat mehrere Lese-/Schreibeköpfe. Die Zustandsänderung hängt nun von dem gelesenen Zeichen aller Köpfe ab und kann auch alle Köpfe verschieben bzw. mit allen gleichzeitig schreiben.

Änderungen in der Definition

$$\delta : Q \times \Gamma^n \to Q \times \Gamma^n \times \{L, N, R\}^n$$

Mehrband-Turingmaschinen

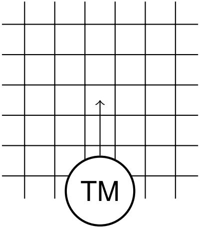


Eine Mehrband-Turingmaschine hat mehrere Bänder. Die Zustandsänderung kann nun auch auf ein anderes Band wechseln. Dabei wird immer auf die zuletzt eingenommene Position auf dem anderen Band gewechselt

Änderungen in der Definition

$$\delta : Q \times \Gamma \times \{1, \dots, n\} \to Q \times \Gamma \times \{L, N, R\} \times \{1, \dots, n\}$$

Mehrdimensionale Turingmaschinen



Eine mehrdimensionale Turingmaschine hat ein mehrdimensionales Band, der Kopf kann sich dann in allen verfügbaren Dimensionen bewegen.

Änderungen in der Definition (für Dimension 2)

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, N, R, U, D\}$$

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

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.

Postsches Korrespondenzproblem

Gegeben sei eine Folge P von Paaren $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ von nichtleeren Worten über einem endlichen Alphabet. Dies nennt man eine **Instanz** des PKP. Eine nichtleere Folge $I = i_1, i_2, \ldots, i_m$ von Indizes aus $\{1, \ldots, n\}$ heißt Lösung zu P, wenn $x_{i_1} x_{i_2} \ldots x_{i_m} = y_{i_1} y_{i_2} \ldots y_{i_m}$.

Beispiel

$$\left(\begin{pmatrix} a \\ aba \end{pmatrix}, \begin{pmatrix} ab \\ bb \end{pmatrix}, \begin{pmatrix} baa \\ aa \end{pmatrix}\right)$$

Lösung: $1, 3, 2, 3$

$$\begin{pmatrix} a \\ aba \end{pmatrix}, \begin{pmatrix} baa \\ aa \end{pmatrix}, \begin{pmatrix} ab \\ bb \end{pmatrix}, \begin{pmatrix} baa \\ aa \end{pmatrix}$$

Aufgabe

1. Finde eine Lösung für folgende Instanz des Post'schen Korrespondenzproblems:

$$((aa, a), (b, aa), (a, aab))$$

2. Zeige, dass folgende Instanz des Post'schen Korrespondenzproblems keine Lösung hat:

$$((ab, b), (bba, abb), (ab, ba), (a, abb))$$

3. Zeige, dass das Post'sche Korrespondenzproblem für Wörter über einem Alphabet mit nur einem Symbol entscheidbar ist.

Aufgabe

Finde eine Lösung für folgende Instanz des PKP:

$$((001, 0), (01, 011), (01, 101), (10, 001))$$

Aufgabe

Finde eine Lösung für folgende Instanz des PKP:

$$((001, 0), (01, 011), (01, 101), (10, 001))$$

Eine kürzeste Lösung hat mindestens die Länge 66, z.B:

$$\begin{aligned} I_1 = (&2, 4, 3, 4, 4, 2, 1, 2, 4, 3, 4, 3, 4, 4, 3, 4, 4, 2, 1, 4, \\ &4, 2, 1, 3, 4, 1, 1, 3, 4, 4, 4, 2, 1, 2, 1, 1, 1, 3, 4, 3, 4, 1, 2, \\ &1, 4, 4, 2, 1, 4, 1, 1, 3, 4, 1, 1, 3, 1, 1, 3, 1, 2, 1, 4, 1, 1, 3) \end{aligned}$$

Aufgabe

Finde eine Lösung für folgende Instanz des PKP:

$$((001, 0), (01, 011), (01, 101), (10, 001))$$

Eine kürzeste Lösung hat mindestens die Länge 66, z.B:

$$I_1 = (2, 4, 3, 4, 4, 2, 1, 2, 4, 3, 4, 3, 4, 4, 3, 4, 4, 2, 1, 4, \\ 4, 2, 1, 3, 4, 1, 1, 3, 4, 4, 4, 2, 1, 2, 1, 1, 1, 3, 4, 3, 4, 1, 2, \\ 1, 4, 4, 2, 1, 4, 1, 1, 3, 4, 1, 1, 3, 1, 1, 3, 1, 2, 1, 4, 1, 1, 3)$$

Bis zum nächsten Mal!

Lizenzen