Theoretische Informatik II

Dr. Eva Richter / Holger Arnold
Universität Potsdam, Theoretische Informatik, Sommersemester 2008

Übungsblatt 2 (Version 2) — Abgabetermin: 5.5.2008, 12.00 Uhr

---

Turingmaschinen und berechenbare Funktionen

In der Vorlesung wurden berechenbare Funktionen als Abbildungen auf Wörtern über einem Alphabet vorgestellt. Die folgende Definition präzisiert diesen Begriff noch einmal:

Definition 1. Eine Funktion $f : \Sigma^* \to \Sigma^*$ auf Wörtern über dem Alphabet Σ heißt *partiell berechenbar*, wenn es eine Turingmaschine M_f gibt, die auf jeder Eingabe w aus dem Definitionsbereich von f mit dem Funktionswert $f(w)$ auf dem Band anhält und auf jeder anderen Eingabe nicht anhält. Die Funktion f heißt *total berechenbar*, wenn sie partiell berechenbar und auf jedem Wort aus Σ^* definiert ist.

Als *Funktionswert* versteht man dabei den Teil des Bandes von der Kopfposition bis zum letzten Zeichen vor dem ersten Zeichen, das nicht in Σ liegt. Wenn also (x, q, yaz) mit $y \in \Sigma^*$, $x, z \in \Gamma^*$ und $a \in \Gamma - \Sigma$ die Endkonfiguration der Turingmaschine auf der Eingabe w ist, dann ist y der berechnete Funktionswert an der Stelle w.

Beachten Sie, dass die Notation $f : A \to B$ im Allgemeinen nicht bedeutet, dass der Definitionsbereich $\operatorname{dom} f$ *der Funktion f gleich A ist. Die Funktion f muss also nicht für alle Elemente von A definiert sein.*

Berechenbare Funktionen auf natürlichen Zahlen

Häufig ist es nützlich, Abbildungen auf abstrakten Objekten wie z.B. Zahlen zu betrachten, die dafür als Wörter codiert werden müssen. Für natürliche Zahlen wird in der Berechenbarkeitstheorie üblicherweise eine unäre Codierung über dem Alphabet $\{0, 1\}$ verwendet. Dabei ist 0^i die Codierung der Zahl i und $0^{i_1}10^{i_2}1 \ldots 10^{i_k}$ ist die Codierung des k-Tupels $(i_1, i_2, \ldots, i_k)$. Wie in der Vorlesung bezeichnen wir die Codierung von i mit $\langle i \rangle$ und die Codierung von $(i_1, \ldots, i_k)$ mit $\langle i_1, \ldots, i_k \rangle$. Die berechenbaren Funktionen auf natürlichen Zahlen sind dann wie folgt definiert:

Definition 2. Eine k-stellige Funktion $f : \mathbb{N}^k \to \mathbb{N}$ auf natürlichen Zahlen heißt *partiell berechenbar*, wenn es eine partiell berechenbare Funktion $\hat{f} : \{0, 1\}^* \to \{0, 1\}^*$ gibt, so dass für alle $i_1, \ldots, i_k \in \mathbb{N}$ gilt: $\hat{f}(\langle i_1, \ldots, i_k \rangle) = \langle m \rangle$ für ein $m \in \mathbb{N}$ genau dann, wenn $f(i_1, \ldots, i_k) = m$ ist. Eine die Funktion $\hat{f}$ berechnende Turingmaschine hält bei der Eingabe $\langle i_1, \ldots, i_k \rangle$ also genau dann mit dem Bandinhalt $\langle m \rangle$ an, wenn die Funktion f an der Stelle $(i_1, \ldots, i_k)$ definiert ist und den Wert m besitzt. Die Funktion f heißt *total berechenbar*, wenn sie partiell berechenbar und für alle $(i_1, \ldots, i_k) \in \mathbb{N}^k$ definiert ist.

Aufgabe 2.1

Sei $f : \mathbb{N} \to \mathbb{N}$ eine Funktion auf natürlichen Zahlen und sei $\operatorname{graph} f = \{\langle x, f(x)\rangle \mid x \in \operatorname{dom} f\}$ die Sprache, die den Graphen von f beschreibt. Beweisen Sie die folgenden Aussagen:

1. f ist genau dann partiell berechenbar, wenn $\operatorname{graph} f$ eine Turing-akzeptierbare Sprache ist.
2. f ist genau dann total berechenbar, wenn f auf ganz $\mathbb{N}$ definiert ist und $\operatorname{graph} f$ eine entscheidbare Sprache ist.

Aufgabe 2.2

Sei $A \subseteq \mathbb{N}$ und sei $\hat{A} = \{\langle x\rangle \mid x \in A\} \subseteq \{0,1\}^*$. Beweisen Sie die Äquivalenz der folgenden Aussagen:

1. Die Sprache $\hat{A}$ ist Turing-akzeptierbar.
2. Wenn A nichtleer ist, dann ist A der Bildbereich einer total berechenbaren Funktion.
3. A ist der Bildbereich einer partiell berechenbaren Funktion.
4. A ist der Definitionsbereich einer partiell berechenbaren Funktion.

Hinweis: Verwenden Sie die Äquivalenz von Turing-Akzeptierbarkeit und Aufzählbarkeit. Beweisen Sie die Äquivalenz der Aussagen, indem Sie folgende Implikationen zeigen: 1. $\Longrightarrow$ *2.* $\Longrightarrow$ *3.* $\Longrightarrow$ *4.* $\Longrightarrow$ *1.*

Aufgabe 2.3

Sei $\prec \subseteq \Sigma^* \times \Sigma^*$ eine entscheidbare totale Ordnung (die Sprache $\{\langle x, y\rangle \mid x, y \in \Sigma^* \text{ und } x \prec y\}$ ist also entscheidbar), für die sich alle Wörter von Σ^* aufsteigend nach $\prec$ geordnet durch eine Turingmaschine aufzählen lassen. Zeigen Sie, dass eine Sprache $M \subseteq \Sigma^*$ genau dann entscheidbar ist, wenn sich die Wörter von M aufsteigend nach $\prec$ geordnet durch eine Turingmaschine aufzählen lassen.

Ein Beispiel für eine geeignete totale Ordnung auf Wörtern über Σ *ist die sogenannte kanonisch-lexikographische Ordnung* $\prec_{\text{klex}}$*. Sei* $\Sigma = \{a_0, \ldots, a_{k-1}\}$*. Dann ist* $x \prec_{\text{klex}} y$ *genau dann, wenn* $|x| < |y|$ *oder* $|x| = |y|$ *und* $\operatorname{num}(x) < \operatorname{num}(y)$ *gilt, wobei* $\operatorname{num}(x)$ *der numerische Wert von* x *ist, wenn* x *als Darstellung einer Zahl zur Basis* k *interpretiert wird. Für* $\Sigma = \{0,1\}$ *sind die bezüglich* $\prec_{\text{klex}}$ *geordneten Wörter von* Σ^* *also* ϵ*,* 0*,* 1*,* 00*,* 01*,* 10*,* 11*,* 000*,* 001*, ...*

Hinweis: Unterscheiden Sie in Ihrem Beweis zwischen endlichen und unendlichen Sprachen.

Aufgabe 2.4

Sei G die Sprache, die nur das Wort s enthält, wobei $s = 0$, falls Gott nicht existiert, und $s = 1$, falls Gott existiert. Ist G entscheidbar? Warum oder warum nicht? (beachten Sie, dass die Antwort nicht von Ihrer religiösen Überzeugung abhängt)

Hausaufgabe 2.5

Zeigen Sie, dass die Turing-akzeptierbaren Sprachen unter Komplementbildung nicht abgeschlossen sind, falls es Turing-akzeptierbare Sprachen gibt, die nicht entscheidbar sind (was tatsächlich der Fall ist).

Hausaufgabe 2.6

Konstruieren Sie eine Turingmaschine, welche die Funktion $f : \Sigma^* \to \Sigma^*$ für $\Sigma = \{0, 1, \#\}$ berechnet, die wie folgt definiert ist:

$$f(w) = \begin{cases} v^{(2^n)} & \text{falls } w = 0^n \# v \text{ für ein } v \in \{0,1\}^* \text{ und ein } n \in \mathbb{N}, \\ \# & \text{sonst.} \end{cases}$$

Beschreiben Sie informal die Arbeitsweise der Turingmaschine. Verwenden Sie ein Unterprogramm zum Verdoppeln des Wortes rechts des Symbols #. Geben Sie die genauen Vor- und Nachbedingungen und ein Übergangsdiagramm Ihres Unterprogramms an.

Hausaufgabe 2.7

Sei $f : \mathbb{N} \to \mathbb{N}$ eine streng monotone total berechenbare Funktion. Beweisen Sie, dass die Sprache $R_f = \{\langle y \rangle \mid y = f(x) \text{ für ein } x \in \mathbb{N}\}$ entscheidbar ist.

Hausaufgaben Für jede Hausaufgabe können Sie maximal 3 Punkte bekommen. Die Punkte werden nach folgenden Regeln vergeben:

3 Punkte	= die Aufgabe wurde im Wesentlichen korrekt gelöst
2 Punkte	= die Aufgabe wurde nur teilweise gelöst
1 Punkt	= die Lösung der Aufgabe enthielt größere Fehler oder Lücken
0 Punkte	= die Aufgabe wurde nicht gelöst oder enthielt sehr viele Fehler oder Lücken

Sprechzeiten Haben Sie Fragen, Anregungen oder Probleme? Lassen Sie es uns wissen!

- Sprechen Sie in den Übungen Ihre Tutorin bzw. Ihren Tutor an.
- Holger Arnold, Raum 1.21, holger@cs.uni-potsdam.de
  Sprechzeiten: mittwochs 14.00–15.00 Uhr und nach Vereinbarung
- Dr. Eva Richter, Raum 1.25, erichter@cs.uni-potsdam.de
  Sprechzeiten: dienstags 13.30–15.00 Uhr und nach Vereinbarung