Technische Universität München
Fakultät für Informatik
Prof. Tobias Nipkow, Ph.D.
Sascha Böhme, Lars Noschinski

Sommersemester 2011
Lösungsblatt 9
25. Juli 2011

Einführung in die Theoretische Informatik

Hinweis: *Bitte beachten Sie unbedingt die Hinweise zum Übungsablauf und zu den Aufgabentypen auf der THEO-Website (`http://theo.in.tum.de/`).*

Hausaufgabe 1 (5 Punkte)

Sei $a : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die Ackermann-Funktion.

1. Zeigen Sie, dass $f(m,n) := \mathit{twopow}(a(m,n))$ nicht primitiv rekursiv ist, wobei $\mathit{twopow}(x) = 2^x$ gelten soll.

2. Zeigen Sie, dass $g(m,n) := \mathit{max}(10 \dot{-} a(m,n), 4)$ primitiv rekursiv ist. Geben Sie dazu eine Definition für g an und begründen Sie.

Lösungsvorschlag

1. Angenommen, f sei primitiv rekursiv. Dann ist auch $h(m,n) := \mathit{log2}\,(f(m,n))$ primitiv rekursiv, wobei *log2* der ganzzahlige Logarithmus zur Basis 2 ist, denn *log2* ist ebenfalls primitiv rekursiv:

$$\mathit{log2}\,(x) = \mathit{max}\,\{y \leq x \mid \mathit{twopow}\,(y) \leq x\}$$

   Damit ist auch $h(m,n)$ primitiv rekursiv, und es gilt

$$h(m,n) = \mathit{log2}\,(f(m,n)) = \mathit{log2}\,(\mathit{twopow}(a(m,n))) = a(m,n)$$

   Da die Ackermann-Funktion nicht primitiv rekursiv ist, ist dies ein Widerspruch zur Annahme. Daher kann f nicht primitiv rekursiv sein.

2. Wir beobachten: Falls $a(m,n)$ größer als 6 (also $10 - 4$) ist, gilt $g(m,n) = 4$. Da es nur endliche viele Werte gibt, für die die Ackermann-Funktion einen Wert kleiner oder gleich 6 annimmt, kann $g(m,n)$ durch entsprechende Fallunterscheidungen angegeben werden und ist damit primitiv-rekursiv. Diese Fallunterscheidungen können zum Beispiel auf den Gleichungen auf Seite 279 der Vorlesung basieren:

$$\begin{aligned} g(m,n) = {} & \mathit{ifthen}\,(m = 0,\ \mathit{max}\,(10 \dot{-} (n+1),\ 4), \\ & \mathit{ifthen}\,(m = 1,\ \mathit{max}\,(10 \dot{-} (n+2),\ 4), \\ & \mathit{ifthen}\,(m = 2,\ \mathit{max}\,(10 \dot{-} (2(n+3) \dot{-} 3),\ 4), \\ & \mathit{ifthen}\,(m + n = 3,\ 5,\ 4))))) \end{aligned}$$

   Alle verwendeten Funktionen sind primitiv rekursiv.

Hausaufgabe 2 (5 Punkte)

Zeigen Sie die Unentscheidbarkeit der folgenden Mengen und wenden Sie zum Beweis Techniken der Reduzierbarkeit eines Problems A auf ein Problem B an.

1. $H_{\Sigma^*} = \{w \mid M_w \text{ hält für alle Eingaben}\}$
2. $C = \{w \mid M_w \text{ berechnet die Funktion } g \text{ mit } g(n) = 0 \text{ für alle n}\}$
3. Sei h eine totale, berechenbare Funktion. Dann ist $A = \{w \mid M_w \text{ berechnet } h\}$ unentscheidbar.

Lösungsvorschlag

Wir gehen davon aus (siehe Vorlesung), dass es eine universelle Turingmaschine U gibt, die die Berechnungen jeder Turingmaschine M_w auf deren Eingabe x simulieren kann, und deshalb insbesondere genau dann hält, wenn M_w hält. Außerdem wurde in der Vorlesung bewiesen, dass das Halteproblem H_0 auf leerem Band nicht entscheidbar ist.

1. Wir reduzieren das bekannte Halteproblem H_0 auf das Problem H_{Σ^*} durch Konstruktion einer totalen und berechenbaren Funktion f wie folgt.

   Es sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die bei Eingabe eines Wortes y folgendes ausführt: Zunächst wird die Turingmaschine M_w bei leerer Eingabe simuliert (beispielsweise auf einem zweiten Band). Falls M_w hält, dann hält auch $M_{w'}$.

   Offenbar gilt nun $w \in H_0 \Leftrightarrow f(w) \in H_{\Sigma^*}$, d.h. f reduziert H_0 auf H_{Σ^*}.

2. Wir verfahren analog zur Lösung der vorhergehenden Aufgabe und reduzieren mit Hilfe einer Funktion f das Problem H_0 auf C:

   Es sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die bei Eingabe eines Wortes y folgendes ausführt: Zunächst wird die Turingmaschine M_w bei leerer Eingabe simuliert. Falls M_w hält, dann schreibt $M_{w'}$ eine 0 auf das Band und terminiert.

   Offenbar gilt nun $w \in H_0 \Leftrightarrow f(w) \in C$, d.h. f reduziert H_0 auf C.

3. Sei T_h eine Turingmaschine, die h berechnet. Wir reduzieren H_0 auf A.

   Es sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die bei Eingabe eines Wortes y folgendes ausführt: Zunächst wird die Turingmaschine M_w bei leerer Eingabe simuliert. Falls M_w hält, dann simuliert $M_{w'}$ die Turingmaschine T_h auf der Eingabe y.

   Offenbar gilt nun $w \in H_0 \Leftrightarrow f(w) \in A$, d.h. f reduziert H_0 auf A.

Hausaufgabe 3 (5 Punkte)

Sei L eine unendliche Sprache über einem endlichen Alphabet Σ, die von einer Turingmaschine M entsprechend der Länge der Wörter aufgezählt wird (zuerst alle Wörter kleinerer Länge).
Zeigen Sie, dass L entscheidbar ist.

Lösungsvorschlag

Die Turingmaschine M berechnet eine totale und surjektive Funktion $f : \mathbb{N} \to L$. Wir geben einen Algorithmus an, der die Funktion χ_L berechnet (also ob $w \in L$ gilt):

- Setze $i = 0$
- Wiederhole:
  - Ist $f(i) = w$, gib „wahr“ zurück und brich ab
  - Ist $|f(i)| > |w|$, gib „falsch“ zurück und brich ab
  - Setze $i = i + 1$

Dieser Algorithmus terminiert immer: Da L unendlich und Σ endlich ist, gibt es ein Wort $u \in L$ mit $|w| < |u|$. Da f surjektiv ist, gibt es ein n, so dass $f(n) = u$, d.h. der Algorithmus terminiert spätestens nach n Schritten. Weiterhin werden die Wörter ihrer Länge nach aufgezählt. Wenn also w in L liegt, gibt es einen Index $m < n$ mit $f(m) = w$; der Algorithmus ist also korrekt.
Also ist χ_L berechenbar und damit L entscheidbar.

Hausaufgabe 4 (5 Punkte)

Ist es entscheidbar, ob bei der Ausführung

(i) eines `LOOP`-Programms P (ii) eines `WHILE`-Programms P

auf Eingabe 0 jeder Variable mehr als 1000 Mal ein Wert zugewiesen wurde? Begründen Sie Ihre Behauptung.

Lösungsvorschlag

1. Programme P sind Programmtext mit endlich vielen Variablen x_0 bis x_n. Die Ausführung von P kann simuliert werden durch ein weiteres Programm, in das hinter jeder Zuweisung zu Variable x_i folgende Anweisungen eingefügt werden:

   $$x'_i := x'_i + 1; \quad \texttt{IF } x'_i = 1001 \texttt{ THEN } y := y + 2^i$$

   wobei die x'_i (für alle $0 \leq i \leq n$) und y Variablen sind, die nicht in P vorkommen und mit 0 initialisiert wurden. Falls P stoppt, dann zeigt der Wert von y an, ob allen Variablen von P mindestens 1000 mal ein Wert zugewiesen wurde.

   Jedes `LOOP`-Programm terminiert. Deshalb kann in jedem Fall die verlangte Bedingung geprüft werden.

2. Dieselbe Eigenschaft ist für `WHILE`-Programme unentscheidbar. Wäre sie entscheidbar, so wäre es auch das Halteproblem für `WHILE`-Programme auf Eingabe 0: Man fügt am Ende des Programmes die folgenden Anweisungen mit `LOOP`-Schleife an: Für jede Variable X, die im Programm vorkommt die Anweisungen

   $$X := 1000; \ LOOP\ X\ DO\ X := X + 1\ END$$

   und für eine Variable Y, die im ursprünglichen Programm nicht verwendet wurde:

   $$Y := 1000; \ LOOP\ Y\ DO\ Y := Y + 1\ END$$

Das neue Programm erfüllt die Eigenschaft genau dann, wenn das ursprüngliche Programm terminiert: Terminiert das ursprüngliche Programm nicht, so wird zumindest Y im neuen Programm nicht mehr als 1000 beschrieben. Terminiert das ursprüngliche Problem jedoch, so stellen die zusätzlichen Anweisungen sicher, dass jede Variable mehr als 1000 mal beschrieben wurde.

Die Unentscheidbarkeit des Halteproblems für `WHILE`-Programm kann man analog zum Halteproblem auf leerem Band bei Turingmaschinen zeigen.

Quiz 1

Beantworten Sie kurz die folgenden Fragen:

1. Jede unentscheidbare Sprache enthält eine entscheidbare Teilmenge.
2. Jede Teilmenge einer entscheidbaren Sprache ist entscheidbar.
3. Für jede unentscheidbare Sprache A gibt es eine echte Obermenge, die ebenfalls unentscheidbar ist.
4. Aus „A entscheidbar“ und „$A \cap B$ entscheidbar“ folgt „B entscheidbar“.

Lösungsvorschlag

Eine Menge $A \subseteq \Sigma^*$ heißt entscheidbar, falls die charakteristische Funktion χ_A total auf Σ^* und berechenbar ist.
Dann ergeben sich die folgenden Antworten und Begründungen.

1. Ja, denn jede Sprache $L \subseteq \Sigma^*$ enthält eine endliche Teilmenge, und jede endliche Teilmenge von Σ^* ist entscheidbar.

   Bemerkung: Eine z. B. einelementige Menge $\{a\}$ ist zwar entscheidbar. Dies muss aber nicht bedeuten, dass man dieses eine Element $a \in \Sigma^*$ kennt bzw. zu konstruieren in der Lage ist. Wir wissen lediglich die abstrakte Existenz dieses a bzw. die Existenz eines Algorithmus, der für vorgelegtes w den Wert der charakteristischen Funktion berechnet, d. h. prüft, ob $w = a$ gilt, wobei der Vergleich $w = a$ für Wörter natürlich mit abbrechendem Algorithmus berechenbar ist.
2. Nein, denn $\{0, 1\}^*$ ist entscheidbar, aber das allgemeine Halteproblem $H \subseteq \{0, 1\}^*$ ist nicht entscheidbar.
3. Ja, denn für ein $w \notin A$ (und das existiert immer, denn Σ^* ist entscheidbar) ist $A \cup \{w\}$ ebenfalls unentscheidbar, falls A unentscheidbar ist.
4. Nein. Gegenbeispiel: $\varnothing$ und $\varnothing \cap H$ sind entscheidbar, weil beide Mengen endlich sind. H (das allgemeine Halteproblem) ist aber nicht entscheidbar.

Tutoraufgabe 1

1. Falls A auf B mit Funktion f reduzierbar ist, dann gilt $f^{-1}(B) = A$, aber nicht notwendigerweise $f(A) = B$. Beweis!
2. Falls A reduzierbar auf B und B semi-entscheidbar ist, dann ist auch A semi-entscheidbar. Beweis!
3. Sei $B \subseteq \Sigma^*$ mit $B \neq \Sigma^*$ und $B \neq \emptyset$ entscheidbar.
   Zeigen Sie: B ist reduzierbar auf $\Sigma^* \setminus B$.

Lösungsvorschlag

1. Nach Definition gilt $x \in A \Leftrightarrow f(x) \in B$, und wir erinnern an die Definition $f^{-1}(B) = \{x \mid f(x) \in B\}$.

   Nun ist $x \in A \Rightarrow f(x) \in B$ gleichbedeutend mit $f(A) \subseteq B$, woraus insbesondere auch $A \subseteq f^{-1}(B)$ folgt. Aus $x \in A \Leftarrow f(x) \in B$ folgt $A \supseteq f^{-1}(B)$. Damit hat man $A = f^{-1}(B)$ bewiesen.

   Durch Angabe eines Beispiels zeigen wir nun, dass i. A. $f(A) \neq B$ gilt.

   Sei $\Sigma^* = \{0,1\}^*$, $B = \{0\}^* \subseteq \Sigma^*$ und $A = \{0x \mid x \in \Sigma^*\}$.

   Die Abbildung $f : \Sigma^* \to \Sigma^*$ mit

$$f(w) = \begin{cases} 0 & \text{falls } w \in A \\ 1 & \text{falls } w \notin A \end{cases}$$

   ist offenbar total, berechenbar und reduziert A auf B. Es gilt $00 \in B$, aber $00 \notin f(A)$.

   Man bemerkt, dass B beliebig vergrößert werden kann, solange $B \cap f(\overline{A}) = \emptyset$ erfüllt bleibt. Im Beispiel gilt $f(\overline{A}) = \{1\}$.

2. Sei f eine totale, berechenbare Funktion, die A auf B reduziert. Dann gilt für die charakteristische Funktion χ'_B von B

$$w \in A \Rightarrow f(w) \in B \Rightarrow \chi'_B(f(w)) = 1$$

   und

$$w \notin A \Rightarrow f(w) \notin B \Rightarrow \chi'_B(f(w)) = \perp$$

   Daraus folgt $\chi'_A(w) = \chi'_B(f(w))$.

   Offenbar ist χ_A berechenbar, mithin ist A semi-entscheidbar.

3. Seien $a \in \Sigma^* \setminus B$ und $b \in B$. Dann definieren wir die Abbildung $f : \Sigma^* \to \Sigma^*$ mit

$$f(w) = \begin{cases} a & \text{für } \chi_B(w) = 1 \\ b & \text{für } \chi_B(w) = 0 \end{cases}$$

   f ist offenbar total, berechenbar und reduziert B auf $\Sigma^* \setminus B$.

Tutoraufgabe 2

1. Seien L_1 und L_2 rekursiv aufzählbare Mengen. Sind die folgenden Mengen L_a und L_b rekursiv aufzählbar? Beweisen Sie Ihre Antwort!

   (i) $L_a = L_1 \cup L_2$ (i) $L_b = \{x \mid x \in L_1 \Leftrightarrow x \in L_2\}$

2. Sei $L_n \subseteq A$ für alle $n \in N$ rekursiv aufzählbar. Zeigen Sie, dass dann auch

$$L = \bigcup_{i \in \mathbb{N}} L_i$$

   rekursiv aufzählbar ist.

3. Sei $R \subseteq M \times M$ eine rekursiv aufzählbare Relation über einer Grundmenge M.

   Zeigen Sie, dass die transitive Hülle R^+ von R rekursiv aufzählbar ist.

Lösungsvorschlag

1. Seien $f_1 : \mathbb{N} \to L_1$ und $f_2 : \mathbb{N} \to L_2$ totale, berechenbare Funktionen, die beide surjektiv sind.

   (i) Wir zeigen, dass L_a rekursiv aufzählbar ist. Wir können ohne Einschränkung annehmen, dass L_1 und L_2 nicht leer sind, denn ansonsten ist L_a gleich einer der beiden Mengen, und somit rekursiv aufzählbar. Es existieren also Aufzählungsfunktionen f_1 und f_2. Wir definieren

   $$f_a(n) = \begin{cases} f_1(m) & \text{falls } n = 2m \\ f_2(m) & \text{falls } n = 2m+1 \end{cases}$$

   f_a ist surjektiv auf $\mathbb{N} \to L_1 \cup L_2$, total und berechenbar.

   (ii) Seien $L_1, L_2 \subseteq \Sigma^*$. Dann gilt $L_b = (L_1 \cap L_2) \cup (\overline{L_1} \cap \overline{L_2})$. Wir zeigen, dass L_b im Allgemeinen nicht rekursiv aufzählbar ist. Wir benützen dabei die Äquivalenz, dass eine Menge genau dann rekursiv aufzählbar ist, wenn sie semi-entscheidbar ist.

   Es genügt $L_1 = \emptyset$ zu setzen und L_2 so zu wählen, dass L_2 semi-entscheidbar und das Komplement $\overline{L_2}$ nicht semi-entscheidbar ist.

   Sei $L_2 = K = \{w \mid M_w[w]\downarrow\}$. Dann ist nach Korollar 4.75 der Vorlesung $\overline{K}$ nicht semi-entscheidbar und mithin nicht rekursiv aufzählbar. Die leere Menge ist nach Definition rekursiv aufzählbar.

   Wegen $L_b = \overline{L_2}$ ist damit L_b nicht rekursiv aufzählbar.

2. Wieder einmal ist die Cantorsche Paarungsfunktion äußerst hilfreich:

   Wir nehmen ohne Einschränkung an, dass L nicht leer ist, denn ansonsten ist sie trivialerweise rekursiv aufzählbar. Sei d ein beliebiges Element aus L. Für jedes L_i können wir eine Aufzählungsfunktion $f_i : \mathbb{N} \to A$ annehmen, so dass $f_i(\mathbb{N}) = L_i$. Falls L_i leer ist, dann setzen wir $f_i(n) = d$. Nun definieren wir $f : \mathbb{N} \to A$ durch

   $$f(n) = f_{p_1(n)}(p_2(n)) \,.$$

   Offensichtlich ist f total und berechenbar, da f_i, p_1 und p_2 total und berechenbar sind. Man sieht leicht, dass $f(\mathbb{N}) = L$. Damit ist L rekursiv aufzählbar.

3. Wir zeigen zunächst, dass wenn zwei Relationen $R, S \in M \times M$ rekursiv aufzählbar sind, dann auch ihre Komposition $R \circ S = \{(x, z) \mid \exists y.(x, y) \in R \wedge (y, z) \in S\}$: Wir wählen wieder o.E. ein $d \in R \circ S$ und nehmen Aufzählungsfunktionen f_R und f_S an. Dann definieren wir

   $$f_{R \circ S}(n) = \begin{cases} (x, z) & \text{falls } y = y', \text{ wobei } (x, y) := f_R(p_1(n)) \text{ und } (y', z) := f_S(p_2(n)) \\ d & \text{sonst} \end{cases}$$

   Damit ist $R \circ S$ rekursiv aufzählbar.

   Mit einer einfachen Induktion sieht man, dass R^n für alle $n \geq 1$ ebenfalls rekursiv aufzählbar ist, denn $R^{n+1} = R \circ R^n$.

   Mit Aufgabenteil 2 folgt, dass $R^+ = \bigcup_{i \in \mathbb{N}} R^i$ rekursiv aufzählbar ist.