Technische Universität München
Fakultät für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Werner Meixner, Dr. Alexander Krauss

Sommersemester 2010
Lösungsblatt 11
15. Juli 2010

Einführung in die Theoretische Informatik

Hausaufgabe 1 (5 Punkte)

Seien die Funktionen $f_1, f_2 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ für alle $m, n \in \mathbb{N}$ definiert durch

$$f_1(m,n) = n \dot{-} m^2 \quad \text{bzw.} \quad f_2(m,n) = m^2 \dot{-} n\,.$$

Stellen Sie die Funktionen $g_1 = \mu f_1$ und $g_2 = \mu f_2$ durch übliche Operationen über den reellen Zahlen dar und begründen Sie die Gültigkeit Ihrer Darstellungen.

Lösungsvorschlag

- Darstellung von g_1: Es gilt

$$\begin{aligned}(\mu f_1)(n) &= \begin{cases} \min\{m \in \mathbb{N} \mid f_1(m,n) = 0\} & \text{falls ein solches } m \text{ existiert und } \underbrace{f_1(k,n) \neq \bot \text{ für alle } k \leq m \text{ gilt}}_{\text{immer erfüllt, da } f_1 \text{ total ist}} \\ \bot & \text{sonst} \end{cases} \\ &= \min\{m \in \mathbb{N} \mid n \dot{-} m^2 = 0\} \\ &= \min\{m \in \mathbb{N} \mid n \leq m^2\} \\ &= \min\{m \in \mathbb{N} \mid \sqrt{n} \leq m\} \\ &= \lceil \sqrt{n} \rceil\,. \end{aligned}$$

- Darstellung von g_2: Es gilt

$$\begin{aligned}(\mu f_2)(n) &= \min\{m \in \mathbb{N} \mid m^2 \dot{-} n = 0\} \\ &= \min\{m \in \mathbb{N} \mid m^2 \leq n\} \\ &= 0\,. \end{aligned}$$

Hausaufgabe 2 (5 Punkte)

Wir betrachten die lexikographische Ordnung $\prec$ auf Wörtern über Σ. Man beachte, dass für verschiedene Buchstaben $a, b \in \Sigma$ und alle $u, v \in \Sigma^*$ stets $au \prec bv$ folgt, falls $a \prec b$ gilt. Seien $f : \Sigma^* \to \mathbb{N}$ eine totale und berechenbare Funktion und
$L = \{y \in \mathbb{N} \mid \exists u, v \in \Sigma^*.\ u \neq v, f(u) = f(v) \leq y\}$.

Zeigen Sie: Wenn f monoton ist (d.h. $u \prec v \implies f(u) \leq f(v)$), dann ist L entscheidbar.

Ist stets $L = \emptyset$ entscheidbar?

Lösungsvorschlag

Sei a der kleinste Buchstabe im Alphabet Σ. Dann gilt $\epsilon \prec a \prec a^2 \prec a^3 \prec \ldots \prec a^n \prec \ldots$. Falls $u \in \Sigma^*$ einen anderen Buchstaben $b \neq a$ enthält, gilt $a^n \prec b$ für alle $n \in \mathbb{N}$.

Sei nun $y \in \mathbb{N}$.
Falls $f(a^y) < y$, dann gibt es nach dem Schubfachprinzip $k < l \leq y$, so dass $f(a^k) = f(a^l)$. Es folgt $y \in L$.
Falls $y \leq f(a^y)$. Dann wird geprüft, ob $f(a^k) \neq f(a^l)$ für alle $k, l \in [0, y+1]$. Falls dies gilt, folgt $y \notin L$. Anderfalls folgt $y \in L$.

$L = \emptyset$ ist nicht entscheidbar. Ein Beweis wurde nicht verlangt, aber die Aussage kann mit dem Satz von Rice bewiesen werden, wenn man das Problem als Menge von Turingmaschinen umschreibt, die die betreffenden Funktionen darstellen.

Hausaufgabe 3 (5 Punkte)

1. Ist es entscheidbar, ob bei der Ausführung

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

   auf Eingabe 0, ein bestimmter Programmpunkt 50000 Mal erreicht wird? Begründen Sie Ihre Behauptung.

2. Wir betrachten eine modifizierte WHILE-Sprache, in der die Anzahl der Variablen auf 50000 beschränkt ist.

   Ist das Halteproblem für solche WHILE-Programme entscheidbar? Begründung!

Lösungsvorschlag

Programme P sind Programmtext. Die Ausführung von P kann simuliert werden durch ein weiteres Programm, in das zusätzliche Beobachtungsvariablen und -anweisungen $o := o+1$ eingefügt werden können, so dass o in P nicht vorkommt und mit 0 initialisiert wurde. Falls P stoppt oder abgebrochen wird, dann zeigt der Wert von o an, ob P zum Programmpunkt der Anweisung $o := 1$ mindestens 50000 Mal gekommen war.

1. Jedes LOOP-Programm terminiert. Deshalb kann in jedem Fall die verlangte Bedinung 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 (dessen Unentscheidbarkeit kann man analog zum Halteproblem auf leerem Band bei Turingmaschinen zeigen). Man fügt am Ende nach entsprechenden Umbenennungen die folgenden Anweisungen mit LOOP-Schleife an:
   $x_0 := 50000;\ LOOP\ x_0\ DO\ o := o + 1\ END$. Das neue Programm erfüllt die Eigenschaft genau dann, wenn das ursprüngliche Programm terminiert.

Hausaufgabe 4 (5 Punkte)

Sei $\Sigma = \{0, 1\}$.

1. Seien $A, B \subseteq \Sigma^*$ und A sei reduzierbar auf B, i. Z. $A \leq B$. Beweisen oder widerlegen Sie: Wenn B regulär ist, dann ist auch A regulär.

2. Sei M_{w_0} eine Turingmaschine mit $L(M_{w_0}) \neq \emptyset$. Verwenden Sie Reduktion, um zu zeigen, dass die folgenden Mengen unentscheidbar sind:

   - $L_1 = \{w \in \Sigma^* \mid M_w \text{ hält auf allen Eingaben aus } L(M_{w_0})\}$
   - $L_2 = \{(v, w) \in \Sigma^* \times \Sigma^* \mid L(M_v) = L(M_w)\}$

Lösungsvorschlag

1. Wir widerlegen die Aussage.

   Sei A eine kontextfreie Sprache, die nicht regulär ist. Dann ist A entscheidbar und es gibt eine totale berechenbare charakteristische Funktion $\chi : \Sigma^* \to \mathbb{N}$.

   Wir kodieren 1 und 0 aus $\mathbb{N}$ entsprechend als Wort aus Σ^* und bezeichnen die χ entsprechende Abbildung als $f : \Sigma^* \to \Sigma^*$. Wir setzen $B = \{1\} \subseteq \Sigma^*$. Dann ist f offenbar eine Reduktion von A auf B und B ist regulär. Widerspruch!

2. - Wir reduzieren H_0 auf L_1: Für einen gegebenen Code einer Turingmaschine w sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die die Ausführung von M_w auf dem leeren Band simuliert. Falls M_w hält, wird M_{w_0} ausgeführt.

     Die Abbildung f, die diese Konstruktion beschreibt, ist offensichtlich total und berechenbar und es gilt

     $$w \in H_0 \iff M_w[0] \downarrow \iff (\forall v \in L(M_{w_0}).\, M_{w'}[v] \downarrow) .$$

     Damit ist $w \in H_0$ genau dann wenn $w' \in L_1$. Es folgt $H_0 \leq L_1$ und damit ist L_1 unentscheidbar.

   - Sei v_0 die Kodierung einer Turingmaschine, für die $L(M_{v_0}) = \Sigma^*$ gilt. Wir zeigen die Unentscheidbarkeit der Menge $L_2' = \{w \mid L(M_{v_0}) = L(M_w)\}$, woraus sofort die Unentscheidbarkeit der Menge L_2 folgt, denn wir können mit $f'(w) = (v_0, w)$ die Menge L_2' auf L_2 reduzieren.

     Nun reduzieren wir H_0 auf L_2': Für einen gegebenen Code einer Turingmaschine w sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die bei Eingabe y zunächst die Ausführung von M_w auf dem leeren Band simuliert. Falls M_w hält, wird die Eingabe y akzeptiert.

     Die Abbildung f, die diese Konstruktion beschreibt, ist offensichtlich total und berechenbar und es gilt

     $$w \in H_0 \iff M_w[0] \downarrow \iff (\forall v \in \Sigma^*.\, M_{w'}[v] \downarrow) .$$

     Damit gilt $w \in H_0$ genau dann, wenn $w' \in L_2'$. Es folgt $H_0 \leq L_2'$ und damit ist auch L_2 unentscheidbar.

Vorbereitung 1

Wir betrachten wieder die in der Vorlesung beschriebene Kodierung von Turingmaschinen durch Wörter über $\Sigma^* = \{0,1\}^*$. Für ein $w \in \Sigma^*$ beschreibt $\varphi_w : \Sigma^* \to \Sigma^*$ dann die Funktion, die durch die Turingmaschine M_w berechnet wird. Finden Sie informelle Beschreibungen für die folgenden Mengen:

1. $A = \{w \in \Sigma^* \mid \varphi_w = \Omega\}$.
2. $B = \{w \in \Sigma^* \mid \varphi_w(101) \neq \bot\}$.
3. $C = \{w \in \Sigma^* \mid \varphi_w(\epsilon) = w\}$.
4. $D = \{(u,v,w) \in \Sigma^* \times \Sigma^* \times \Sigma^* \mid \varphi_u(w) = \varphi_v(w)\}$.

Lösungsvorschlag

1. Die Menge aller (Codes der) Turingmaschinen, die die überall undefinierte Funktion Ω berechnen, die also auf keiner Eingabe halten.
2. Die Menge aller (Codes der) Turingmaschinen, die auf der Eingabe 101 anhalten.
3. Die Menge aller (Codes der) Turingmaschinen, die ihren eigenen Code auf das leere Band schreiben. Das entspricht den sogenannten Quines (nach dem Logiker und Philosophen Willard V. O. Quine), den Programmen, die bei leerer Eingabe ihren eigenen Quellcode ausgeben.
4. Ein Tripel (u,v,w) liegt genau dann in D, wenn die Turingmaschinen M_u und M_v bei der Eingabe w die gleiche Ausgabe liefern. Insbesondere muss auch $M_u[w] \downarrow \iff M_v[w] \downarrow$ gelten.

Vorbereitung 2

Sei Σ ein Alphabet. Ein Paar (l,r) mit $l,r \in \Sigma^*$ nennen wir *konsistent*, falls l Präfix von r oder r Präfix von l ist. Sei P eine endliche Menge von Tupeln (x,y) mit $x,y \in \Sigma^+$. Dann nennen wir eine Folge von konsistenten Paaren $(l_0,r_0),\ldots,(l_k,r_k)$ eine *Berechnung in P*, falls es für alle i mit $0 \leq i < k$ ein Paar $(x,y) \in P$ gibt mit $l_i x = l_{i+1}$ und $r_i y = r_{i+1}$, wobei auch $(l_0,r_0) \in P$ gelten soll. Die Berechnung terminiert, wenn $l_k = r_k$ gilt.

1. Machen Sie sich klar, dass es genau dann eine terminierende Berechnung in P gibt, wenn das Post'sche Korrespondenzproblem P lösbar ist.
2. Sei $P = \{(1,111),(10111,10),(100,00)\}$. Geben Sie eine terminierende Berechnung an, die mit $(10111,10)$ beginnt. Welche Berechnungsschritte sind deterministisch?
3. Warum kann es nur triviale Berechnungen in P geben, wenn für ein bestimmtes n und für alle $(x,y) \in P$ die Längenbedingung $|x| = |y| = n$ gilt?

Lösungsvorschlag

1. Ein geordnetes n-Tupel $P = ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ nennen wir ein Post'sches Korrespondenzproblem. Eine Menge von Indizes $i_1, i_2, \ldots, i_l$ nennen wir eine Lösung von P, falls die Wörter $x = x_{i_1} x_{i_2} \ldots x_{i_l}$ und $y = y_{i_1} y_{i_2} \ldots y_{i_l}$ gleich sind, d. h. $x = y$.

   Bei gegebener Lösung setzen wir $l_0 = x_{i_1}$ und $r_0 = y_{i_1}$. Aus $x = y$ und $|x_{i_1}| \leq |y_{i_1}|$ folgt x_{i_1} Präfix von y_{i_1}. Aus $x = y$ und $|x_{i_1}| \geq |y_{i_1}|$ folgt y_{i_1} Präfix von x_{i_1}.

   Im Allgemeinen setzen wir $l_j = l_{j-1} x_{i_{j+1}}$ und $r_j = r_{j-1} y_{i_{j+1}}$ und zeigen die Konsistenz per Induktion.

   Umgekehrt kann man aus gegebener terminierender Berechnung $(l_0, r_0), \ldots, (l_k, r_k)$ leicht eine Lösung rekonstruieren.

2. Wir bringen die Tupel (x, y) wie folgt zur Anwendung
   $(x_0, y_0) = (10111, 10)$, $(x_1, y_1) = (1, 111)$, $(x_2, y_2) = (1, 111)$, $(x_3, y_3) = (100, 00)$.

   Die Berechnungen sind

l_0	$=$	10111
r_0	$=$	10
l_1	$=$	101111
r_1	$=$	10111
l_2	$=$	1011111
r_2	$=$	10111111
l_3	$=$	1011111100
r_3	$=$	1011111100

   Nur (x_1, y_1) wurde deterministisch ausgewählt.

3. Für das erste Tupel (x, y) folgt aus der Präfixbedingung der Konsistenz bereits $x = y$. Dies folgt auch für jedes weitere Tupel, das konsistent zur Anwendung kommt. (x, x) ist stets eine triviale Lösung des Post'chen Problems.

Tutoraufgabe 1

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 Paarngsfunktion 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.

Tutoraufgabe 2

Zeigen Sie mit Hilfe des Satzes von Rice:

1. $L_1 = \{w \in \Sigma^* \mid L(M_w) \text{ ist kontextfrei}\}$ ist unentscheidbar.
2. $L_2 = \{w \in \Sigma^* \mid \forall n \in \mathbb{N}.\, \varphi_w(n) = 3n + 5\}$ ist unentscheidbar.

Warum kann man den Satz von Rice auf folgende Menge nicht anwenden?

3. $L_3 = \{w \in \Sigma^* \mid \forall n \in \mathbb{N}.\, \varphi_w(n) = \perp \text{ und } w \text{ ist ein Palindrom}\}$

Lösungsvorschlag

Der Satz von Rice vereinfacht die Reduktionsbeweise. Wir müssen lediglich prüfen, ob die Eigenschaften L_1 und L_2 nicht triviale Eigenschaften der berechneten Funktionen sind.

1. Es gibt mindestens eine Sprache $L \subseteq \Sigma^*$, die nicht kontextfrei ist, denn kontextfreie Sprachen sind entscheidbar, also kann das Halteproblem nicht kontextfrei sein.

   Daraus folgt $L_1 \neq \Sigma^*$.

   Andererseits gibt es mindestens eine kontextfreie Sprache über Σ, d. h. $L_1 \neq \emptyset$.

2. $L_2 \neq \emptyset$, weil es für die Berechnung von $3n + 5$ eine Turingmaschine gibt.

   Andererseits ist klar, dass nicht jede Turingmaschine die Funktion $3n+5$ berechnet, d. h. $L_2 \neq \Sigma^*$.

3. Hier ist der Satz von Rice nicht anwendbar, denn dafür müsste es eine Funktionenmenge F geben, so dass $L_3 = \{w \mid \varphi_w \in F\}$. Es ist aber nicht ausgeschlossen, dass es Wörter v und w gibt, so dass $\varphi_v = \varphi_w$ und v ein Palindrom ist, aber w keines. Ob es solche Wörter tatsächlich gibt hängt von der konkreten Codierung von Turingmaschinen in Wörter ab.

Tutoraufgabe 3

1. Zeigen Sie, dass die folgende Instanz P des Post'schen Korrespondenzproblems keine Lösung hat:
   $$P = \{(10, 101), (011, 11), (101, 011)\}$$
2. Zeigen Sie, dass das Post'sche Korrespondenzproblem über einem Alphabet mit nur einem Symbol entscheidbar ist.

Lösungsvorschlag

1. Nur $(l_0, r_0) = (10, 101)$ kann als erstes Tupel zur Anwendung kommen, weil die anderen Tupel nicht konsistent sind.

   Wir reduzieren die Berechnung äquivalent und beginnen bei $(u_1, v_1) = (\epsilon, 1)$.

   $(10, 101)$ kann nicht zur Anwendung kommen, und auch $(011, 11)$ nicht. Anwendung von $(101, 011)$ führt aber sofort in einem Berechnungszyklus zu (u_1, v_1) zurück.

   $$\begin{array}{lcl} u_1 & = & \epsilon \\ v_1 & = & 1 \\ \hline u_2 & = & 101 \\ v_2 & = & 1011 \\ \hline u_1 & = & \epsilon \\ v_1 & = & 1 \end{array}$$

2. Alle Tupel $(x, y) \in P$ haben die Form (a^m, a^n) mit $m, n > 0$.

   Falls $m = n$ gilt für ein $(x, y) = (a^m, a^n) \in P$, dann ist das Problem trivial lösbar durch den Index von (x, y).

   Falls $m > n$ gilt für alle $(x, y) = (a^m, a^n) \in P$, dann ist das Problem nicht lösbar.

   Falls $m < n$ gilt für alle $(x, y) = (a^m, a^n) \in P$, dann ist das Problem ebenfalls nicht lösbar.

   Falls es $(x_1, y_1) = (a^m, a^n), (x_2, y_2) = (a^k, a^l) \in P$ gibt mit $m > n$ und $k < l$, dann konstruiert man eine Lösung wie folgt:

   Man nimmt $(l - k)$ Mal das Paar (x_1, y_1) und $(m - n)$ Mal das Paar (x_2, y_2). Dann gilt für $(x, y) = (x_1, y_1)^{l-k}(x_2, y_2)^{m-n}$

   $$\begin{aligned} x &= (a^m)^{l-k}(a^k)^{m-n} = a^{ml-kn}, \\ y &= (a^n)^{l-k}(a^l)^{m-n} = a^{-nk+lm}, \end{aligned}$$

   also $x = y$.