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

Sommersemester 2011
Lösungsblatt 10
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/`).*

Wenn die Zusatzaufgaben abgegeben und Punkte dadurch erworben werden, dann werden diese Punkte der zweiten Semesterhälfte gutgeschrieben. Die 40% Regelung bezieht sich dabei stets auf die normalen Hausaufgaben. Die Zusatzaufgaben können also dazu benutzt werden, fehlende Punkte auszugleichen.

Hausaufgabe 1 (6 Punkte)

Sei $\Sigma = \{0, 1\}$. Geben sie für jede der folgenden Mengen an, ob sie entscheidbar ist oder nicht. Beweisen Sie Ihre Behauptungen.

1. $L_1 = \{w \in \Sigma^* \mid \varphi_w(001) = 110\}$
2. $L_2 = \{w \in \Sigma^* \mid \varphi_{110}(001) = w\}$
3. $L_3 = \{w \in \Sigma^* \mid \forall x \in \Sigma^*.\ \varphi_w(xx) = \varphi_w(x)\, \varphi_w(x)\}$

Lösungsvorschlag

1. L_1 ist unentscheidbar. Für den Beweis setzen wir

   $$F = \{\varphi \mid \varphi(001) = 110 \text{ und } \varphi \text{ ist berechenbar}\}$$

   Dann besteht F offensichtlich nur aus berechenbaren Funktionen. Außerdem ist F nicht trivial, denn sicherlich ist F nicht leer und es gibt berechenbare Funktionen $\varphi \notin F$. Also ist nach dem Satz von Rice die Sprache $\{w \in \Sigma^* \mid \varphi_w \in F\}$ unentscheidbar. Da diese Sprache genau L_1 ist, ist also L_1 unentscheidbar.

2. L_2 ist entscheidbar, denn $|L_2| \leq 1$, d.h., L_2 ist endlich.

3. L_3 ist unentscheidbar. Wir setzen

   $$F = \{\varphi \mid \varphi(xx) = \varphi(x)\varphi(x) \text{ für alle } x \in \Sigma^* \text{ und } \varphi \text{ ist berechenbar}\}$$

   Wiederum ist F eine nicht-triviale Menge berechenbarer Funktionen. Also folgt aus dem Satz von Rice, dass L_3 unentscheidbar ist.

Hausaufgabe 2 (6 Punkte)

Sei $\Sigma = \{a, b\}$.

1. Bestimmen Sie alle Lösungen des Postschen Korrespondenzproblems
$$P_1 = \{(a, aaa), (abaaa, ab), (ab, b)\}$$

2. Zeigen Sie, dass die folgende Instanz des Postschen Korrespondenzproblems keine Lösung hat:
$$P_2 = \{(ab, aba), (baa, aa), (aba, baa)\}$$

3. Besitzt die folgende Instanz des Postschen Korrespondenzproblems eine Lösung oder nicht? Geben Sie dazu entweder alle Lösungen mit Begründung an oder zeigen Sie, dass es keine Lösung gibt.
$$P_3 = \{(bba, b), (ba, baa), (ab, aab), (aaa, a)\}$$

Lösungsvorschlag

1. Die Menge der Lösungen ist $\{(i_1 i_2 i_3 i_4)^n \mid n \in \mathbb{N}, i_1 = 2, i_2 = i_3 = 1, i_4 = 3\}$.

2. Zu Beginn kann nur das Tupel (ab, aba) zur Anwendung kommen, da alle anderen Tupel nicht konsistent sind.

   Nun können wir die Berechnung reduzieren, so dass wir mit (ϵ, a) fortfahren.

   Offensichtlich kann nun weder (ab, aba) noch (baa, aa) angewendet werden. Durch (aba, baa) erhalten wir aber $(aba, abaa)$, was wiederum zu (ϵ, a) reduziert werden kann. Damit kann also diese Instanz des Postschen Korrespondenzproblems keine Lösung besitzen.

3. Diese Instanz des Postschen Korrespondenzproblems ist ebenfalls lösbar.

   Wir untersuchen zunächst, welches Tupel das Anfangstupel einer Lösung bilden kann. In (bba, b) hat die erste Komponente ein b mehr als die zweite Komponente, jedoch hat keines der anderen Tupel von P_3 in der zweiten Komponente mehr b's als in der ersten, d.h., diese Karte kann nicht Anfang (und auch nicht irgendein anderer Teil) einer Lösung sein. Im Tupel (ab, aab) ist die erste Komponente kein Präfix der zweiten Komponente, und daher kann eine Lösung nicht damit beginnen. Mit dem Tupel (aaa, a) kann ebenfalls nicht begonnen werden, da danach nur wieder dieses Tupel anwendbar ist und dadurch die Länge des oberen Wortes schneller wächst als die des unteren Wortes, und dieser Unterschied kann nicht wieder ausgeglichen werden durch andere Tupel, denn diese sind niemals anwendbar. Das einzig mögliche Anfangstupel ist also (ba, baa).

   Von (ba, baa) beginnend kann nur (aaa, a) angewendet werden, denn insbesondere (ab, aab) passt nicht. Nach (ba, baa) und (aaa, a) kann nur (ab, aab) verwendet werden, und man erhält eine Lösung. Einzige Alternativen zu (ab, aab) im dritten Schritt ist nur (aaa, a), aber man sieht leicht, dass danach wiederum nur (aaa, a) anwendbar ist und darüberhinaus das obere Wort schneller länger wird als das untere. Daher kann dies nicht zu einer Lösung führen.

   Die Lösungsmenge ist also $\{(i_1 i_2 i_3)^n \mid n \in \mathbb{N}, i_1 = 2, i_2 = 4, i_3 = 3\}$.

Hausaufgabe 3 (4 Punkte)

Sei Σ ein Alphabet und seien $L_1, L_2 \subseteq \Sigma^*$ rekursiv aufzählbare Sprachen. Zeigen Sie:

1. $A := L_1 L_2$ ist rekursiv aufzählbar.
2. $B := L_1 \cap L_2$ ist rekursiv aufzählbar.

Hinweis: Die Cantorsche Paarfunktion bzw. die dazugehörigen Projektionen p_1 und p_2 könnten hilfreich sein.

Lösungsvorschlag

1. Der Fall $A = \varnothing$ ist trivial. Wir betrachten den Fall $A \neq \varnothing$. Seien also f_1 und f_2 Funktionen, die L_1 und L_2 aufzählen. Dann ist die Funktion $f(n) = f_1(p_1(n))f_2(p_2(n))$ total und berechenbar (denn f_1, f_2, p_1 und p_2 sind total und berechenbar) und zählt A auf:

   Da $f_1(p_1(n)) \in L_1$ und $f_2(p_2(n)) \in L_2$ für alle $n \in \mathbb{N}$, ist $f(n) \in L$ für alle $n \in \mathbb{N}$. Umgekehrt gilt für ein $x \in L$, dass $y \in L_1$ und $z \in L_2$ existieren mit $x = yz$. Da f_1 und f_2 die Sprachen L_1 und L_2 aufzählen, gibt es $k, l \in \mathbb{N}$ mit $f_1(k) = y$ und $f_2(l) = z$. Da die Cantor'sche Paarfunktion bijektiv ist, gibt es ein $n \in \mathbb{N}$, so dass $p_1(n) = k$ und $p_2(n) = l$ gibt; f zählt also auch alle Elemente von A auf.

2. Ist $B = \varnothing$, so ist es rekursiv aufzählbar nach Definition. Ist $B = L_1 \cap L_2 \neq \varnothing$, so gilt $L_1 \neq \varnothing$ und $L_2 \neq \varnothing$ und es gibt ein $d \in B$ und Funktionen f_1 und f_2, die L_1 und L_2 aufzählen Wir definieren nun f wie folgt.

   $$f(n) = \begin{cases} f_1(p_1(n)) & \text{wenn } f_1(p_1(n)) = f_2(p_2(n)) \\ d & \text{sonst} \end{cases}$$

   Die Funkion ist total und berechenbar (denn f_1 und f_2 sind ebenfalls total und berechenbar) und zählt B auf: Gilt $f_1(k) = f_2(l)$ für $k, l \in \mathbb{N}$, so ist $f_1(k)$ in L_1 und L_2, also auch in ihrem Schnitt enthalten. Weiterhin gilt auch $d \in B$. Alle von f aufgezählten Werte sind also in B enthalten.

   Da f_1 die Menge L_1 und f_2 die Menge L_2 aufzählt, existierem umgekehrt für jedes $x \in B$ Zahlen $k, l \in \mathbb{N}$, so dass $f_1(k) = f_2(l) = x$ gilt. Da die Cantor'sche Paarfunktion bijektiv ist, gibt es ein $n \in \mathbb{N}$, so dass $p_1(n) = k$ und $p_2(n) = l$ gibt; f zählt also auch alle Elemente von B auf.

Hausaufgabe 4 (4 Punkte)

Zeigen Sie durch Rückführung auf die Definition, dass die folgenden Funktionen primitiv rekursiv sind. Sie dürfen das erweiterte Schema der Rekursion sowie die bereits definierten Funktionen, insbesondere auch den beschränkten Maximumsoperator, verwenden.

1. $f(n) = \begin{cases} \sqrt[3]{n} & \text{falls } \sqrt[3]{n} \in \mathbb{N} \\ 0 & \text{sonst} \end{cases}$

2. $\text{even}(n) = \begin{cases} 1 & \text{falls } n \text{ gerade, d.h. } 2|n \\ 0 & \text{sonst} \end{cases}$

Lösungsvorschlag

1. Wir setzen $f(n) = \max\{x \leq n \mid x * x * x = n\}$
2. Die einfachste Lösung ist:

$$\text{even}(0) = 1 \qquad\qquad \text{even}(n+1) = 1 \dot{-} \text{even}(n)$$

Alternativ kann man auch die verschränkte Rekursion aus Übung 8 verwenden um die Funktionen even und odd zu definieren:

$$\begin{aligned} \text{even}(0) &= 1 & \text{odd}(0) &= 0 \\ \text{even}(n+1) &= \text{odd}(n) & \text{odd}(n+1) &= \text{even}(n) \end{aligned}$$

Zusatzaufgabe 1 (3 Zusatzpunkte)

Sei nun $v \in \Sigma^*$ ein beliebiges, aber fest gewähltes Wort. Überlegen und begründen Sie, was man zur Entscheidbarkeit der folgenden Menge aussagen kann:

$$L = \{w \in \Sigma^* \mid \varphi_v(w) = 110\}$$

Lösungsvorschlag

Die Entscheidbarkeit von L hängt von der Wahl von v ab. Ist beispielsweise φ_v total, dann ist L entscheidbar. Wählt man allerdings v als die Codierung der Turingmaschine, die eine Turingmaschine nimmt und diese auf der Eingabe 001 simuliert, so ist $\varphi_v(w) = \varphi_w(001)$. Damit ist aber $L = L_1$ und daher unentscheidbar.

Zusatzaufgabe 2 (3 Zusatzpunkte)

Sei $\Sigma = \{0, 1, \$\}$. Wir betrachten eine Variante des PCP namens $\text{PCP}_\$$, bei der die Wörter auf den Karten immer mit \$ beginnen:

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$ mit $x_i, y_i \in L(\$(0|1)^+)$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$ mit $n > 0$, so dass $x_{i_1} \cdots x_{i_k} = y_{i_1} \cdots y_{i_k}$?

Zeigen oder widerlegen Sie: $\text{PCP}_\$$ ist entscheidbar.

Lösungsvorschlag

$\text{PCP}_\$$ ist entscheidbar. Wir zeigen dazu: Es gibt eine Lösung für eine Instanz des $\text{PCP}_\$$ genau dann, wenn es ein $i \in \{1, \ldots, k\}$ gibt, so dass $x_i = y_i$ gilt. Offensichtlich können wir dies entscheiden.
Gibt es ein i mit $x_i = y_i$, so ist i eine Lösung des Problems. Nehmen wir umgekehrt an, dass es eine Lösung $i_1 \ldots i_m$ des $\text{PCP}_\$$ gibt. Wir betrachten die erste Karte der Lösung.

- $\underline{|x_{i_1}| = |y_{i_1}|}$. Gilt $x_{i_1} = y_{i_1}$, so sind wir fertig. Andernfalls kann i_1 nicht der Beginn einer Lösung sein, denn zwei Wörter, die mit $\$x_{i_1}$ und $\$y_{i_1}$ anfangen, haben ein unterschiedliches Präfix.

- $|x_{i_1}| < |y_{i_1}|$. Da x_{i_1} und x_{i_2} unterschiedliche Länge haben, muss $m > 1$ gelten, also die Lösung aus mindestens zwei Karten bestehen. Betrachten wir jetzt $x = x_{i_1} \cdots x_{i_m}$ und $y = y_{i_1} \cdots y_{i_m}$. Dann steht an $(|x_{i_1}|+1)$-ter Stelle in x ein \$, in y aber ein Zeichen aus $\{0, 1\}$. Also kann $i_1 \dots i_m$ keine Lösung sein.

- $|x_{i_1}| > |y_{i_1}|$. Analog zum vorherigen Fall.

Also folgt aus der Existenz einer Lösung , dass es ein i mit $x_i = y_i$ gibt. Aus unserer Vorüberlegung folgt damit die Entscheidbarkeit des $\mathrm{PCP}_{\$}$.

Tutoraufgabe 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), also 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.

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

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. Wir betrachten das Postsche Korrespondenzproblem $P = (\,(1, c1), (abc, ab)\,)$.

   Bestimmen Sie *alle* Lösungen von P!

2. Sei $P = (p_1, p_2)$ ein Postsches Korrespondenzproblem über einem beliebigem Alphabet Σ mit $p_i = (x_i, y_i)$ und $|\,|x_i| - |y_i|\,| = 1$ für $i = 1, 2$.

   Zeigen Sie, dass P entscheidbar ist!

Lösungsvorschlag

1. Für die Menge L der Lösungen gilt $L = \{(i_1 i_2)^n \mid n \in \mathbb{N}, n \neq 0, i_1 = 2, i_2 = 1\}$.

2. Es gibt genau dann eine Lösung von P, wenn $1, 2$ oder $2, 1$ eine Lösung ist, d. h. wenn entweder $x_1 x_2 = y_1 y_2$ oder $x_2 x_1 = y_2 y_1$ gilt.

   Beweis:

   Wir nehmen an, dass $i_1 i_2 \dots i_k$ eine Lösung von P ist. Zunächst gilt sicher $k \geq 2$, denn aus der Längenbedingung für die p_i folgt, dass weder p_1 noch p_2 selbst schon eine Lösung ist.

   Falls $i_1 \neq i_2$, dann folgt $|x_{i_1} x_{i_2}| = |y_{i_1} y_{i_2}|$. Mithin ist bereits $i_1 i_2$ eine Lösung, d. h. $1, 2$ oder $2, 1$ ist Lösung.

   Sei nun $i_1 = i_2$ und o. B. d. A. $i_1 = i_2 = 1$. Wir nehmen ebenfalls o. B. d. A. an, dass y_1 ein Prefix von x_1 ist, d. h. $x_1 = u_1 u_2 \dots u_n u_{n+1}$ und $y_1 = u_1 u_2 \dots u_n$ mit $u_1, \dots, u_{n+1} \in \Sigma$. Da $i_1 i_2$ Teil einer Lösung ist, gilt

   $$\begin{aligned} x_1 x_1 &= y_1 y_1 u_n u_{n+1} \quad \text{und} && (1) \\ u_1 u_2 \dots u_n u_{n+1} u_1 u_2 \dots u_n u_{n+1} &= u_1 u_2 \dots u_n u_1 u_2 \dots u_n u_n u_{n+1} && (2) \end{aligned}$$

Durch buchstabenweisen Vergleich folgt

$$u_{n+1} = u_1 = u_2 = \ldots = u_{n-1} = u_n \,, \tag{3}$$

Wir erhalten $p_1 = (a^{n+1}, a^n)$ für irgendeinen Buchstaben $a \in \Sigma$.

In der Sequenz $i_1 i_2 \ldots i_k$ muss es eine Teilsequenz $i_1 i_2 \ldots i_l$ geben, die selbst Lösung ist und für die gilt $i_{l-1} = 2, i_l = 2$. Dann zeigt man analog $p_2 = (b^m, b^{m+1})$.

Da nun p_1 und p_2 irgendwo gematcht werden müssen, folgt $a = b$. Mithin folgt, dass $1, 2$ eine Lösung ist. Es folgt sogar, dass $2, 1$ ebenfalls eine Lösung ist.