Technische Universität München
Fakultät für Informatik
Lehrstuhl Informatik II
Prof. Dr. Helmut Seidl
Dr. Werner Meixner

Sommersemester 2013
Übungsblatt 12
8. Juli 2013

Einführung in die Theoretische Informatik

Abgabetermin: 15. Juli, 13 Uhr in die **THEO Briefkästen**

Hausaufgabe 1 (4 Punkte)

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

1. Wir betrachten das spezielle Halteproblem $K = \{w \in \Sigma^* \mid M_w[w]\downarrow\}$ und das Halteproblem auf leerem Band $H_0 = \{w \in \Sigma^* \mid M_w[\epsilon]\downarrow\}$.

   Zeigen Sie durch hinreichend genaue Spezifikation und Begründung einer Reduktionsabbildung (wie in den entsprechenden Beweisen der Vorlesung), dass H_0 reduzierbar ist auf K, d.h. $H_0 \leq K$.

2. Zeigen Sie, dass die Menge $R = \{w \in \Sigma^* \mid \varphi_w(0) = \perp\}$ unentscheidbar ist.
   Dabei sei φ_w diejenige (partielle) Funktion $\varphi_w : \mathbb{N} \to \mathbb{N}$, die von der Turingmaschine M_w berechnet wird.

Hausaufgabe 2 (4 Punkte)

1. Sei $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ für alle $m, n \in \mathbb{N}$ gegeben durch $f(m, n) = m^2 \dot{-} n$.

   Zeigen Sie, dass μf primitiv rekursiv ist.

2. Sei $g : \mathbb{N} \to \mathbb{N}$ total und μ-rekursiv, und sei $f : \mathbb{N} \to \mathbb{N}$ definiert durch die Startwerte $f(0) = 1$ und $f(1) = 2$ zusammen mit der Rekursion

   $$f(n) = g(n)\cdot f(n-1)\cdot f(n-2) \quad \text{für alle} \quad n \in \mathbb{N} \text{ mit } n \geq 2\,.$$

   Zeigen Sie durch Anwendung der Erzeugungsregeln für μ-rekursive Funktionen mit Hilfe der Paarfunktion $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ und deren Umkehrfunktionen p_1 und p_2, dass f μ-rekursiv ist.

   Hinweis:
   Sie dürfen zusätzlich zu den Basisfunktionen der primitiven Rekursion die folgenden Funktionen als primitiv rekursiv annehmen: $plus(m, n)$ $(+)$, $times(m, n)$ $(\cdot)$, $pred(n)$, $c(m, n)$, $p_1(n)$, $p_2(n)$, $ifthen(n, a, b)$ und die konstante k-stellige Funktion c_n^k. Sie dürfen die erweiterte Komposition und das erweiterte rekursive Definitionsschema benützen. LOOP- und WHILE-Programme sind nicht erlaubt.

Hausaufgabe 3 (4 Punkte)

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

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

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

Hausaufgabe 4 (4 Punkte)

Sei $\Sigma = \{0, 1\}$. Geben Sie jeweils ein Beispiel für die folgenden Objekte an. Falls kein solches Objekt existiert, begründen Sie dies.

1. Eine Funktion $f : \mathbb{N} \to \mathbb{N}$, die primitiv rekursiv ist, aber deren Definitionsbereich (also $\{n \in \mathbb{N} \mid f(n) \neq \bot\}$) endlich ist.
2. Eine Funktion $f : \Sigma^* \to \Sigma^*$, die total ist und für die $\{w \in \Sigma^* \mid \varphi_w = f\}$ entscheidbar ist.
3. Eine Menge $A \subseteq \Sigma^*$, die nicht rekursiv aufzählbar ist und deren Komplement ebenfalls nicht rekursiv aufzählbar ist.
4. Ein unentscheidbarer Wertebereich einer berechenbaren Funktion.

Hausaufgabe 5 (4 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. $H_{\Sigma^*} = \{w \,|\, M_w$ hält für mindestens eine Eingabe$\}$.
2. $C = \{w \mid M_w$ berechnet die Funktion g mit $g(n) = 0$ für alle n$\}$.

Hinweis: *Auf den Übungsblättern in diesem Semester wird es grundsätzlich die drei Aufgabentypen Vorbereitungsaufgabe, Tutoraufgabe und Hausaufgabe geben. Die als Vorbereitung bezeichneten Aufgaben dienen der häuslichen Vorbereitung der Tutoraufgaben. Tutoraufgaben werden in den Übungsgruppen bearbeitet. Dabei wird die Lösung der Vorbereitungsaufgaben vorausgesetzt und entsprechend abgefragt werden. Ab Blatt 2 werden zusätzlich Hausaufgaben gestellt, die selbstständig bearbeitet und zur Korrektur und Bewertung abgegeben werden sollen.*

Vorbereitung 1

Zeigen Sie, dass die polynomielle Reduzierbarkeit $\leq_p$ eine transitive Relation ist.

Vorbereitung 2

Beantworten Sie kurz die folgenden Fragen:

1. Seien A, B Sprachen mit $A \leq_p B$ und B NP-vollständig. Gilt dann A NP-vollständig?
2. Ist $time_M$ für jede deterministische Turingmaschine M berechenbar?

Vorbereitung 3

Beweisen Sie:

1. P ist abgeschlossen unter Komplement.
2. Das Problem, zu entscheiden, ob ein gegebener Graph ein Dreieck enthält, ist in P.

Tutoraufgabe 1

1. Ist $ntime_M$ für jede deterministische Turingmaschine M berechenbar? Begründung!
2. Sei $f : \mathbb{N} \to \mathbb{N}$. Falls $NTIME(f(n))$ eine nichtentscheidbare Sprache enthält, dann ist f nicht berechenbar. Beweis!
3. Wir betrachten die Komplexitätsklasse P. Dann gibt es für jede DTM M mit $L(M) \in$ P ein Polynom p, so dass $time_M(w) \leq p(|w|)$ für alle $w \in \Sigma^*$ gilt.

   Richtig oder Falsch? Begründung!
4. Ist jede in polynomieller Zeit berechenbare Funktion $f : \mathbb{N} \to \mathbb{N}$ auch polynomiell beschränkt ($\exists$ Polynom $p.\ \forall n.\ f(n) \leq p(n)$)? Begründung!
5. Sei p ein Polynom und M eine DTM mit $time_M(w) \leq p(|w|)$ für alle $w \in \Sigma^*$. Konstruieren Sie einen einfachen polynomiell beschränkten Verifikator für $L(M)$.

Tutoraufgabe 2

Seien $A, B \in \Sigma^*$ Sprachen in NP. Zeigen Sie, dass dann auch $A \cup B$, AB und A^* in NP liegen.
Hinweis: Überlegen Sie sich hierzu, wie geeignete Zertifikate aussehen.

Tutoraufgabe 3

Das STUNDENPLAN-Problem, das in der Praxis allen bekannt ist, lässt sich vereinfacht wie folgt formal beschreiben:

Gegeben: Endliche Mengen S (Studierende), V (Vorlesungen) und T (Termine) und eine Relation $R \subseteq S \times V$. Dabei bedeutet $(s, v) \in R$, dass s die Vorlesung v besuchen möchte.

Problem: Gibt es eine Abbildung $f : V \to T$, so dass alle Studierenden einen Überschneidungsfreien Stundenplan haben, also

$$(s, v_1) \in R \wedge (s, v_2) \in R \wedge v_1 \neq v_2 \implies f(v_1) \neq f(v_2)$$

1. Zeigen Sie, dass STUNDENPLAN in NP liegt.
2. Zeigen Sie, dass STUNDENPLAN NP-hart ist, indem sie eine polynomielle Reduktion von 3COL auf STUNDENPLAN angeben.
3. Geben Sie nun eine Reduktion von STUNDENPLAN auf SAT an, die es erlaubt, das Stundenplanproblem mit Hilfe eines SAT-Solvers zu lösen.