Universität Freiburg
PD Dr. A. Jakoby
Sommer 2007

Übungsaufgaben zur Komplexitätstheorie

1. Zeigen Sie, dass alle regulären Sprachen in TIME(n) liegen.

2. Welche der folgenden Aussagen sind richtig? Geben Sie eine Begründung für Ihre Antwort.

   (a) Es gibt einen deterministischen polynomiell-platzbeschränkten Algorithmus, der entscheidet, ob eine natürliche Zahl eine Primzahl ist oder nicht.

   (b) Jede rekursiv aufzählbare Sprache liegt in $\mathcal{PSPACE}$.

   (c) Ist jede entscheidbare Sprache in $\mathcal{P}$, dann ist das Halteproblem in $\mathcal{NP}$.

3. Das Problem des 0/1-Integer Linear Programming (0/1-ILP) ist wie folgt definiert:

   - Gegeben sind $n \cdot m + m$ natürliche Zahlen $c_{i,j} \in \mathbb{N}$ und $b_j \in \mathbb{N}$ für $i \in \{1, \ldots, n\}$ und $j \in \{1, \ldots, m\}$. Wir interpretieren diese Zahlen als Koeffizienten eines Systems linearer Ungleichungen

     $$c_{1,j} \cdot x_1 + \ldots + c_{n,j} \cdot x_n \ \geq \ b_j \ .$$

     Wir können dieses auch über das Matrix-Vektor-Produkt wie folgt ausdrücken:

     $$C \cdot \vec{x} \ \geq \ \vec{b} \ ,$$

     wobei die Werte $c_{i,j} \in \mathbb{N}$ die Matrix C und die Werte b_j den Vektor $\vec{b}$ bilden.

   - Eine Eingabe $C, \vec{b}$ ist genau dann ein Element von 0/1-ILP, wenn es einen Belegung der Variablen $x_1, \ldots, x_n \in \{0, 1\}$ gibt, so dass alle Ungleichungen

     $$c_{1,j} \cdot x_1 + \ldots + c_{n,j} \cdot x_n \ \geq \ b_j$$

     durch diese Belegung erfüllt werden.

   Zeigen Sie, dass 0/1-ILP NP-vollständig ist.

4. Wir betrachten das Handlungsreisendenproblem (TSP). Gegeben ist ein gerichteter Graph $G = (V, E)$ mit einer Kostenfunktion $\gamma : E \to \mathbb{N}$. Unter einer Rundreise verstehen wir eine Knotenfolge $v_1, \ldots, v_n$, so dass aufeinanderfolgende Knoten verbunden sind (d.h. $(v_i, v_{i+1}) \in E$ für alle $i \in \{1, \ldots, n-1\}$ und $(v_n, v_1) \in E$) und jeder Knoten genau einmal besucht wird. Die Länge einer solchen Rundreise ergibt sich aus der Kostenfunktion über

   $$\gamma((v_n, v_1)) + \sum_{i \in \{1, \ldots, n-1\}} \gamma((v_i, v_{i+1})) \ .$$

   - Die Optimierungsvariante vom Typ 2 fragt nach einer kürzesten Rundreise.
   - Die Optimierungsvariante vom Typ 1 fragt nach den Kosten einer kürzesten Rundreise.
   - Die Entscheidungsvariante hat neben G und γ eine Zahl $c \in \mathbb{N}$ als Eingabe und fragt, ob es eine Rundreise mit den Kosten $\leq c$ gibt.

   Zeigen Sie, dass aus der effizienten Lösbarkeit einer der drei Varianten die effiziente Lösbarkeit der beiden anderen Varianten folgt. Wir nennen ein Problem effizient Lösbar, wenn es einen polynomiell-zeitbeschränkten Algorithmus gibt, der dieses Problem löst.

5. Sei $f : \mathbb{N} \to \mathbb{N}$ eine totale berechenbare Funktion und M eine NTM, so dass es zu jedem Wort $w \in L(M)$ eine akzeptierende Berechnung der Länge $\leq f(|w|)$ gibt. Zeigen Sie, dass $L(M)$ entscheidbar ist.

6. Welche der folgenden Aussagen sind richtig? Geben Sie eine Begründung für Ihre Antwort.
   (a) Das Rucksackproblem (RP) ist wie folgt definiert:
   $$\{ \ (p_1, \ldots, p_n, w_1, \ldots, w_n, Q, P) \mid \exists I \subseteq \{1, \ldots, n\} \ : \ \sum_{i \in I} w_i \leq K \ \wedge \ \sum_{i \in I} p_i \geq Q \ \} \ .$$
   Wenn RP $\in$ co-$\mathcal{NP}$, dann gilt $\mathcal{NP}$ = co-$\mathcal{NP}$. (Sie können als bekannt voraussetzen, dass RP $\mathcal{NP}$-vollständig ist.)
   (b) Wenn RP $\in \mathcal{P}$, dann ist $\mathcal{NP} = \mathcal{P}$.

7. Das rationale Rucksackproblem ist wie folgt definiert:
   $$\{ \ (p_1, \ldots, p_n, w_1, \ldots, w_n, Q, P) \mid \exists x_1, \ldots, x_n \in [0,1] \ : \ \sum_{i=1}^{n} w_i \cdot x_i \leq K \ \wedge \ \sum_{i=1}^{n} p_i \cdot x_i \geq Q \ \} \ .$$
   Betrachten Sie nun den folgenden Algorithmus:

   Algorithmus rationales-RP($p_1, \ldots, p_n, w_1, \ldots, w_n, Q, K \in \mathbb{N}$)
   Eingabe: $p_1, \ldots, p_n, w_1, \ldots, w_n, Q, K \in \mathbb{N}$ mit $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}$
   Ausgabe: $x_1, \ldots, x_n \in \mathbb{R}$
   FOR $i := 1$ to n **DO** $x_i := 0$ **END FOR**
   for $i := 1$ to n **do**
   $\quad h \ := \ \sum_{j=1}^{i-1} x_j \cdot w_j$
   $\quad$ **if** $w_i + h \leq K$ **then**
   $\qquad x_i := 1$
   $\quad$ **else**
   $\qquad x_i := \frac{K-h}{w_i}$
   $\quad$ **end if**
   end for
   return $x_1, \ldots, x_n$

   (a) Zeigen Sie, dass der Algorithmus rationales-RP eine optimale Lösung für das rationale Rucksackproblem liefert.
   (b) Zeigen Sie, dass dieser Algorithmus das ursprüngliche Rucksackproblem nicht löst.

8. Welche der folgenden Aussagen sind richtig und welche falsch? Begründen Sie Ihre Antwort.

(a) $2n = O(n)$	(b) $n = o(2n)$	(c) $n^2 = O(n)$	(d) $2n = o(n^2)$
(e) $n^2 = O(n \log^2 n)$	(f) $n \log n = O(n^2)$	(g) $n = o(\log n)$	(h) $3^n = 2^{O(n)}$
(i) $2^{2^n} = O(2^{2^n})$	(j) $2^n = o(3^n)$	(k) $1 = o(n)$	(l) $1 = o(1/n)$

9. Zeigen Sie, dass $\mathcal{P}$ abgeschlossen ist unter Vereinigung, Konkatenation, Komplement und Stern-Operator.

10. Zeigen Sie, dass $\mathcal{NP}$ abgeschlossen ist unter Vereinigung, Konkatenation und Stern-Operator.

11. Definiere die Sprache TRIANGLE als die Menge aller ungerichteten Graphen, die eine 3-Clique beinhalten, d.h.
   $$\text{TRIANGLE} \ := \ \{ \ \langle G \rangle \mid G \text{ besitzt eine 3-Clique } \} \ .$$
   Zeigen Sie, dass TRIANGLE in $\mathcal{P}$ ist.

12. Zeigen Sie, dass die Sprache
   $$\text{ALL}_{\text{DFA}} \ := \ \{ \ \langle M \rangle \mid M \text{ ist ein DFA mit } L(M) = \{0,1\}^* \ \}$$
   in $\mathcal{P}$ ist.

13. Wir nennen zwei Graphen G und H isomorph, wenn die Knoten in G so umbenannt weder können, dass wir H erhalten. Wir definieren
   $$\text{ISO} \ := \ \{ \ \langle G, H \rangle \mid G \text{ und } H \text{ sind isomorph } \} \ .$$
   Zeigen Sie, dass ISO in $\mathcal{NP}$ ist.

14. Definiere

$$\text{MODEXP} := \{ \langle a,b,c,p \rangle \mid a,b,c,p \text{ sind binäre Zahlen aus } \mathbb{N} \text{ mit } a^b \equiv c(\text{mod} p) \} .$$

Zeigen Sie, dass MODEXP in $\mathcal{P}$.

15. Definiere

$$\text{SUBSET-SUM} := \{ \langle S,t \rangle \mid S = \{x_1,\ldots,x_n\} \text{ und für eine Teilmenge } Y \subseteq S \text{ gilt } \sum_{x\in Y} x = t \} .$$

Es ist bekannt, dass SUBSET-SUM NP-vollständig ist, wenn die Werte aus S binär gegeben sind. Zeigen Sie, dass das Problem in $\mathcal{P}$ gelöst werden kann, wenn die Zahlen aus S unär, d.h. als 1^{x_i}, gegeben sind.

16. Zeigen Sie, dass für $\mathcal{P} = \mathcal{NP}$ alle Probleme aus $\mathcal{P}$ bis auf $\emptyset$ und Σ^* $\mathcal{NP}$-vollständig bezüglich der $\leq_{m,p}$-Reduktion sind.

17. Definiere

$$\begin{aligned} \text{CNF}_k := \{ \langle \phi \rangle \mid\ & \phi \text{ ist eine erfüllbare Boolesche Formel in konjunktiver Normalform, wobei} \\ & \text{jede Variable maximal } k\text{-mal vorkommt} \} . \end{aligned}$$

Zeigen Sie,

(a) dass CNF_2 in $\mathcal{P}$ ist.

(b) dass CNF_3 $\mathcal{NP}$-vollstängig ist.

18. Definiere

$$U := \{ \langle M \rangle \# w \#^t \mid \text{die NTM } M \text{ akzeptiert } w \text{ auf einem Berechnungspfad in maximal } t \text{ Schritten} \} .$$

Zeigen Sie, dass U $\mathcal{NP}$-vollständig ist.

19. Zeigen Sie, dass eine polynomiell-zeitbeschränkte DTM zum Faktorisieren von natürlichen Zahlen existiert, wenn $\mathcal{P} = \mathcal{NP}$ ist.

20. Eine Boolesche 2-CNF Formel ist eine über AND verknüpfte Folge von Klauseln, wobei jede Klausel ein OR von maximal zwei Literalen ist. Ein Literal ist eine Variable oder deren Negation. Zeigen Sie, dass

$$\text{2-SAT} := \{ \langle \phi \rangle \mid \phi \text{ ist eine erfüllbare Boolesche 2-CNF Formel} \}$$

in $\mathcal{P}$ ist.

21. Einen regulären Ausdruck nennen wir Stern-frei, wenn er keine Stern-Operation beinhaltet. Sei

$$\text{EQ}_{\text{sf-rex}} := \{ \langle R,S \rangle \mid R \text{ und } S \text{ sind zwei äquivalente Stern-freie reguläre Ausdrücke} \} .$$

(a) Zeigen Sie, dass $\text{EQ}_{\text{sf-rex}}$ in co-$\mathcal{NP}$ ist.

(b) Warum funktioniert Ihr Ansatz nicht, wenn wir uneingeschränkte reguläre Ausdrücke betrachten.

22. Welche der folgenden Aussagen ist korrekt (für $\Sigma = \{0,1\}$):

(a) $\Sigma^* \leq_{m,p}$ COPY

(b) COPY $\leq_{m,p} \Sigma^*$

(c) COPY $\leq_{m,p}$ PALINDROM

(d) COPY $\leq_{m,p} \Sigma^* \setminus \{0,1\}$

Geben Sie, wenn möglich, die Reduktionsfunktion an. Hierbei ist

$$\begin{aligned} \text{COPY} &:= \{ ww \mid w \in \{0,1\}^* \} \\ \text{PALINDROM} &:= \{ ww^{\text{rev}} \mid w \in \{0,1\}^* \} . \end{aligned}$$

23. Zeigen Sie, dass $\mathcal{PSPACE}$ abgeschlossen ist unter Vereinigung, Konkatenation, Komplement und Stern-Operator.

24. Zeigen Sie, dass jede $\mathcal{PSPACE}$-schwere Sprache auch $\mathcal{NP}$-schwer ist.

25. Zeigen Sie, dass $\mathcal{PSPACE} = \mathcal{NP}$ ist, wenn jede $\mathcal{NP}$-schwere Sprache auch $\mathcal{PSPACE}$-schwer ist.

26. Zeige Sie, dass $\mathcal{NP} = \text{co-}\mathcal{NP}$ genau dann gilt, wenn es eine $\mathcal{NP}$-vollständige Sprache L gibt, so dass $\overline{L} \in \mathcal{NP}$ ist.

27. Angenommen $\mathcal{P} = \mathcal{NP}$. Zeigen Sie, dass es einen Algorithmus mit polynomieller Laufzeit gibt, der bei Eingabe einer Booleschen Formel ϕ eine erfüllende Belegung der Formel ϕ findet.

28. Welche der folgenden Aussagen sind korrekt? Beweisen Sie die Korrektheit oder geben Sie Gegenbeispiele an.

    (a) Wenn $L \in \mathcal{NP}$ und $L \leq_{m,p} L'$, dann ist
    - $L' \in \mathcal{NP}$,
    - L' ist $\mathcal{NP}$-schwierig,
    - L' ist $\mathcal{NP}$-vollständig.

    (b) Wenn L $\mathcal{NP}$-vollständig ist und $L \leq_{m,p} L'$, so ist
    - $L' \in \mathcal{NP}$,
    - L' ist $\mathcal{NP}$-schwierig,
    - L' ist $\mathcal{NP}$-vollständig.

    (c) Wenn L $\mathcal{NP}$-schwierig ist und $L \leq_{m,p} L'$, so ist
    - $L' \in \mathcal{NP}$,
    - L' ist $\mathcal{NP}$-schwierig,
    - L' ist $\mathcal{NP}$-vollständig.

29. In einem ungerichteten Graphen $G = (V, E)$ heißt eine Knotenmenge $V' \subseteq V$ unabhängig, wenn keine Kante zwischen zwei Knoten aus V' existiert. Das Independent Set Problem IP enthält dann alle Paare (G, k) von ungerichteten Graphen G und natürlichen Zahlen k, für die G eine unabhängige Knotenmenge mit k Knoten enthält. Zeigen Sie, dass IP $\mathcal{NP}$-vollständig ist.

30. Betrachten Sie das Optimierungsproblem von 3-SAT. Gegeben ist eine 3-CNF Formel, in der in jeder Klausel drei unterschiedliche Literale vorkommen. Ziel ist es, eine Variablenbelegung zu finden, die möglichst viele Klauseln zu erfüllen. Finden Sie einen Algorithmus der (im Erwartungswert) eine Approximationsgüte von 8/7 hat. Gehen Sie hierbei wie folgt vor:

    (a) Betrachten Sie eine erfüllbare 3-CNF und eine zufällige Belegung der Variablen. Bestimmen Sie die erwartete Anzahl der hierbei erfüllten Klauseln.

    (b) Betrachten Sie eine nicht erfüllbare 3-CNF-Instanz und bestimmen sie die erwartete Anzahl der erfüllten Klauseln bei zufälliger Belegung in Abhängigkeit der optimalen Anzahl.

    (c) Geben Sie nun den Approximationsalgorithmus an.

31. Welche der folgenden Aussagen sind korrekt? Beweisen Sie die Korrektheit oder geben Sie Gegenbeispiele an.

    (a) Wenn $L' \in \mathcal{PSPACE}$ und $L \leq_{m,p} L'$ ist, dann gilt
    - $L \in \mathcal{NPSPACE}$,
    - L ist $\mathcal{NP}$-schwierig,
    - L ist $\mathcal{NP}$-vollständig.

    (b) Wenn L' $\mathcal{NPSPACE}$-vollständig und $L \leq_{m,p} L'$ ist, so ist
    - $L \in \text{TIME}(2^{O(1)}) = \text{EXPTIME}$,
    - L ist $\mathcal{NP}$-schwierig,
    - L ist $\mathcal{NP}$-vollständig.

32. Betrachten Sie das Problem k-CLIQUE: Gegeben ist ein Graph $G = (V, E)$. Hat G eine Clique der Größe $\geq k$, d.h. gibt es einen Teilgraphen $C = (V_C, E_C)$ von G, der vollständig ist und für den $|V_C| \geq k$ gilt. Zeigen Sie, dass man k-CLIQUE in Zeit $O(n^{k+3})$ entscheiden kann.

33. Betrachten Sie das Zwei-Personen-Spiel NIM. Im Spielbrett befinden sich k Reihen von $x_1, \ldots, x_k$ Steinen (natürlich ist $x_i \in \mathbb{N}$). Abwechselnd wählt jeder Spieler eine Reihe j mit $x_j > 0$ und entnimmt beliebig viele Steine, aber mindestens einen. Gewonnen hat der Spieler, der den letzten Zug ausführen kann. Betrachten Sie die Sprache

$$\mathrm{NIM}_k \quad := \quad \{\ 1^{x_1}\#1^{x_2}\#\cdots\#1^{x_k} \mid (x_1, ..., x_k) \text{ ist eine Gewinnstellung für den ersten Spieler } \}\ .$$

   (a) Zeichnen Sie den Spielbaum für die Startposition 1#111#1. Wer gewinnt?

   (b) Geben Sie (quantifizierte) Prädikate für

   $$1^{x_1}\#1^{x_2}\#\cdots\#1^{x_k} \in \mathrm{NIM}_k$$

   mit $k = \{1, 2, 3, 4\}$ an.

   (c) Zeigen Sie, dass NIM_k in $\mathcal{PSPACE}$ ist.

   (d) Betrachten Sie nun die Variante NIM-NICHT-7, in der man nicht mehr beliebige Anzahlen von Steinen entfernen darf, sondern nur noch eine Anzahl, die nicht durch sieben teilbar ist und in deren Dezimaldarstellung keine sieben vorkommt. Zeigen Sie, dass $\bigcup_{k \in \mathbb{N}} \text{NIM-NICHT-7}_k$ in $\mathcal{PSPACE}$ ist.

Ein großer Teil dieser Aufgaben ist dem Büchern

- A. Asteroth, C. Baier, *Theoretische Informatik*, Pearson Studium, 2002
- M. Sipser, *Introduction to the Theory of Computation*, Thomson Course Technology, 2004 (2. Auflage)

entnommen.