Lehrstuhl für Informatik 1
Prof. Dr. Gerhard Woeginger
Jan Böker, Tim Hartmann

WS 2018/19
18.01.2019

Übung zur Vorlesung
Berechenbarkeit und Komplexität
Blatt 12

Tutoriumsaufgabe 12.1

Das Makespan-Scheduling-Problem ist das folgende Optimierungsproblem:

> Makespan-Scheduling
>
> **Eingabe:** m Maschinen, n Jobs mit Laufzeiten $p_1, \dots, p_n$.
>
> **zulässige Lösungen:** Jede Zuteilung $s\colon \{1, \dots, n\} \to \{1, \dots, m\}$ der Jobs auf die Maschinen.
>
> **Zielfunktion:** Minimiere den Makespan, d.h. minimiere $\max_{1 \le i \le m} \sum_{j : s(j)=i} p_j$.

(a) Definieren Sie die Entscheidungsvariante des Makespan-Scheduling-Problems.

(b) Beschreiben Sie eine polynomielle Reduktion von Subset-Sum auf die Entscheidungsvariante von Makespan-Scheduling und beweisen Sie ihre Korrektheit.

Tutoriumsaufgabe 12.2

Zeigen Sie, dass BinPacking stark NP-schwer ist.

Tutoriumsaufgabe 12.3

Wir betrachten sogenannte *primitive* Programme, die durch die Hintereinanderausführung von Zuweisungen "$x_i := x_j + c$" und If-Befehlen "IF $(x_i = c')$ THEN $x_j := x_k + c''$ ENDIF" (mit Konstanten $c, c', c'' \in \mathbb{N}$) entstehen. If-Befehle können nicht ineinander verschachtelt werden, und es gibt keine ELSE-Klauseln. Die Eingabe des Programms steht in den Variablen $x_1, \dots, x_k$, und die Ausgabe des Programms steht am Ende in der Variable x_0. Die Variable x_0 enthält keinen Eingabewert und wird mit 0 initialisiert. Beweisen Sie, dass das folgende Problem coNP-schwer ist:

> PrimitiveEq
>
> **Eingabe:** Zwei primitive Programme P_1 und P_2 mit Variablen $x_0, \dots, x_k$.
>
> **Frage:** Berechnen diese beiden Programme dieselbe Funktion $f : \mathbb{N}^k \to \mathbb{N}$?

— bitte wenden —

Hausaufgabe 12.1 **(4 Punkte + 3 Bonuspunkte)**

PARTITION-INTO-THREE-SETS ist das folgende Entscheidungsproblem:

> PARTITION-INTO-THREE-SETS
>
> **Eingabe:** Positive ganze Zahlen $a_1, \dots, a_n$.
>
> **Frage:** Gibt es paarweise disjunkte Mengen $I, J, K \subseteq \{1, \dots, n\}$ mit $I \cup J \cup K = \{1, \dots, n\}$, so dass
> $$\sum_{i \in I} a_i = \sum_{j \in J} a_j = \sum_{k \in K} a_k \;?$$

Zeigen Sie, dass PARTITION-INTO-THREE-SETS NP-vollständig ist. Um die Bonuspunkte zu erhalten, müssen Sie zudem zeigen, dass PARTITION-INTO-THREE-SETS in pseudopolynomieller Zeit gelöst werden kann.

Hausaufgabe 12.2 **(5 Punkte)**

Zeigen Sie, dass das folgende Problem stark NP-schwer ist:

> TRIPLE-PARTITION
>
> **Eingabe:** Positive ganze Zahlen $d_1, \dots, d_{3n}$ mit $\sum_{i=1}^{3n} d_i = nD$.
>
> **Frage:** Können diese Zahlen in n Tripel partitioniert werden, sodass sich in jedem Tripel die Elemente zur Summe D aufaddieren?

Hinweis: Konstruieren Sie eine Reduktion von THREE-PARTITION. Ersetzen Sie dabei die Zahlen a_i durch $a_i + X$, die Zahlen b_i durch $b_i + Y$ und die Zahlen c_i durch $c_i + Z$, wobei die Zahlen X, Y, Z geeignet von S abhängen.

Hausaufgabe 12.3 **(4 Punkte)**

Beweisen Sie, dass das folgende Problem coNP-vollständig ist:

> AT-MOST-THREE-SAT
>
> **Eingabe:** Boole'sche Formel φ in CNF über den Variablen $x_1, \dots, x_n$.
>
> **Frage:** Besitzt φ höchstens drei verschiedene erfüllende Variablenbelegungen?

Abgabe bis Mittwoch, den 30.01.2019 um 12:15 Uhr
im Sammelkasten am Lehrstuhl i1, in Ihrem Tutorium oder am Anfang der Globalübung.