Info III Tutorium

Thomas Pajor

Fakultät für Informatik

ITI Sanders

6. Februar 2007

Aufgabe

Aufgabe 1: Multiprocessor Scheduling (MPS)

Gegeben: $M := \{1, \ldots, m\}$ Prozessoren, $J := \{1, \ldots, n\}$ Jobs, Bearbeitungsdauern für jeden Job: $p : J \to \mathbb{N}$. Außerdem eine Deadline $D \in \mathbb{N}$.

Frage: Existiert ein Schedule s mit Abarbeitungslänge höchstens D? Formal: $\exists s : J \to M$ mit

$$\max_{1 \leq j \leq m} \sum_{i \in s^{-1}(j)} p(i) \leq D?$$

Zeigen Sie: MPS ist $\mathcal{NP}$-vollständig.

Wiederholung: PARTITION

PARTITION

Gegeben: n Gegenstände $\{1, \ldots, n\}$ mit Gewicht $w_i \in \mathbb{N}$.

Frage: Gibt es eine Teilmenge $M \subseteq \{1, \ldots, n\}$ mit

$$\sum_{i \in M} w_i = \sum_{i \notin M} w_i?$$

PARTITION ist $\mathcal{NP}$–vollständig. Beweis siehe Vorlesung. $\square$

Aufgabe 2.

Zeigen Sie: Das Halteproblem ist $\mathcal{NP}$–hart. Warum ist es nicht $\mathcal{NP}$–vollständig?

Aufgabe 3.

Gegeben Sei folgender (naiver) Algorithmus, der für eine Zahl $n \in \mathbb{N}$ entscheidet, ob sie prim ist:

Für jedes $k \in [2, n-1]$ überprüfe ob $k|n$. Falls das für kein k gilt, ist n prim, sonst ist n nicht prim. Dieser Algorithmus führt offenbar $\mathcal{O}(n)$ Divisionen durch.

Warum konnte dennoch erst 2004 gezeigt werden, dass das Problem $\textsc{Primes}$ in $\mathcal{P}$ liegt[1]?

[1] http://www.math.princeton.edu/%7Eannals/issues/2004/Sept2004/Agrawal.pdf