AuD – Algorithmen und Datenstrukturen

Aufgabenblatt 4 : P, NP und NPC

Abgabe bis 23. November, 1600 Uhr – Besprechung am 25.-27. November 2015

von
3

Übungsaufgabe 4.1: Zeigen Sie, dass die folgenden beiden Probleme in NP liegen, indem Sie jeweils einen geeigneten Algorithmus angeben, der das Problem löst.

1. **Eingabe**: Eine endliche Menge U und eine Größe $s(u) \in \mathbb{N}$ für jedes $u \in U$, eine Kapazität $K \in \mathbb{N}$ und eine Zahl $n \in \mathbb{N}$.
   Frage: Gibt es eine Partitionierung von U in paarweise disjunkte Teilmengen $U_1, U_2, \ldots, U_n$ (d.h. die U_i enthalten zusammen alle Elemente aus U und jedes Element aus U ist in genau einem der U_i) derart, dass $\sum_{u \in U_i} s(u) \leq K$ für jedes i gilt?

2. **Eingabe:** Ein endliches Alphabet Σ, eine endliche Menge $S \subseteq \Sigma^*$ von Worten und eine Zahl $n \in \mathbb{N}$.
   Frage: Gibt es ein Wort $w \in \Sigma^*$ mit $|w| \geq n$ derart, dass w ein Teilwort von jedem $x \in S$ ist?

von
7

Übungsaufgabe 4.2:

1. Zeigen Sie, dass P und NP jeweils gegenüber Konkatenation abgeschlossen sind (d.h. mit $L_1, L_2 \in P$ ist auch $L_1 \cdot L_2 \in P$ und entsprechend für NP).

2. Geben Sie unter der Annahme $P = NP$ einen Algorithmus an, der in polynomieller Zeit zu einer aussagenlogischen Formel eine erfüllende Belegung bestimmt, sofern eine existiert. Begründen Sie kurz, warum Ihr Algorithmus das Gewünschte leistet.

3. Zeigen Sie, dass unter der Annahme $P = NP$ jedes $L \in P$ außer $L = \emptyset$ und $L = \Sigma^*$ NP-vollständig ist. Warum gilt die Aussage nicht für die beiden ausgeschlossenen Fälle?

von
6

Übungsaufgabe 4.3:

1. Sei `2-SAT` die Menge aller (sinnvoll codierten) aussagenlogischen Formeln, die mindestens zwei erfüllende Belegungen haben. Zeigen Sie, dass `2-SAT` NP-vollständig ist.

2. Die folgende Beschreibung verallgemeinert das Spiel *Minesweeper* auf einen ungerichteten Graphen: Sei G ein ungerichteter Graph. Jeder Knoten von G enthält entweder eine einzelne *Mine* oder ist leer. Der Spieler kann eine Knoten wählen. Ist es eine Mine, hat er verloren. Ist der Knoten leer, dann wird dieser mit der Anzahl der direkt benachbarten Knoten beschriftet, die eine Mine enthalten. Der Spieler gewinnt, wenn alle leeren Knoten gewählt wurden. Das Problem ist nun folgendes: Gegeben ein Graph zuzüglich einiger beschrifteter Knoten, ist es möglich Minen so auf die verbleibenden Knoten abzulegen, dass jeder Knoten v, der mit k beschriftet ist, genau k direkt benachbarte Knoten besitzt, die eine Mine enthalten? Formulieren Sie dieses Problem sinnvoll als formale Sprache (und füllen Sie dadurch die Lücken in obiger Beschreibung) und zeigen Sie dann, dass dieses Problem NP-vollständig ist.

Informationen und Unterlagen zur Veranstaltung unter:

http://www.informatik.uni-hamburg.de/TGI/lehre/vl/WS1516/AuD

Version vom 10. November 2015 *Bisher erreichbare Punktzahl*: 48