FGI-1 – Formale Grundlagen der Informatik I

Logik, Automaten und Formale Sprachen

Aufgabenblatt 7 : Komplexitätstheorie

Präsenzaufgabe 7.1: Sei $G = (V, E)$ ein ungerichteter Graph. Bei einer k-Färbung von G wird jedem Knoten $v \in V$ eine Farbe $c \in [k] = \{1, 2, \ldots, k\}$ zugewiesen. Die Färbung ist korrekt, wenn zwei benachbarte Knoten nie die gleiche Farbe haben. Formal ist eine *k-Färbung* dann also eine Abbildung $f : V \to [k]$ und eine *korrekte k-Färbung* f erfüllt $f(x) \neq f(y)$ für jede Kante $\{x, y\} \in E$. Wir definieren folgendes Problem:

$$k\text{-col} = \{\langle G\rangle \mid G \text{ besitzt eine korrekte } k\text{-Färbung}\}$$

1. Geben Sie einen Algorithmus an der, gegeben einen ungerichteten Graphen $G = (V, E)$ und eine Färbung $f : V \to \{1, 2, \ldots, k\}$, deterministisch in Polynomialzeit entscheidet, ob f den Graphen korrekt mit k Farben färbt.

2. Geben Sie nun einen Algorithmus an, der *nichtdeterministisch* in Polynomialzeit für einen gegebenen Graphen G und eine gegebene Zahl k bestimmt, ob G eine korrekte k-Färbung besitzt oder nicht. Wieviel Platz benötigt Ihr Algorithmus?

Präsenzaufgabe 7.2: Sie dürfen nutzen, dass 3-col NP-vollständig ist. Susi Sorglos will nun zeigen, dass auch 2-col NP-vollständig ist und geht dafür wie folgt vor: Zu einem gegebenen Graphen $G = (V, E)$ fügt sie einen neuen Knoten v_n hinzu sowie Kanten $\{v_n, v\}$ für jedes $v \in V$, d.h. der neue Knoten wird mit jedem anderen Knoten verbunden. Sei der neue Graph mit G_n bezeichnet. G_n besitzt nun eine 3-Färbung gdw. G eine 2-Färbung besitzt. Da die Konstruktion in Polynomialzeit möglich ist, folgt damit dass 2-col NP-vollständig ist.

Klappt Susis Beweis?

von
8

Übungsaufgabe 7.3: Gegeben sei ein gerichteter Graph $G = (V, E)$ (ohne Kantengewichte) sowie zwei Knoten $s, t \in V$. Das zu lösende Problem ist die Frage, ob t von s aus in G erreichbar ist, d.h. ob ein (gerichteter) Pfad zwischen s und t existiert.

1. Geben Sie einen Algorithmus an, der nichtdeterministisch und in Linearzeit (d.h. mit nicht mehr als $k \cdot |V|$ Schritten, wobei k eine feste, nicht von der Eingabe abhängige Konstante ist) und nur unter Nutzung von linear viel Platz (d.h. abermals mit nicht mehr als $k' \cdot |V|$ zusätzlichem Platzbedarf) das Problem entscheidet.

2. Geben Sie nun einen Algorithmus an, der das Problem abermals nichtdeterministisch und in Linearzeit, aber nur unter Nutzung von *logarithmisch viel Platz* (d.h. es darf nur $k \cdot \log |V|$ zusätzlicher Speicher benutzt werden) löst.

von
4

Übungsaufgabe 7.4: Betrachten Sie das folgende Problem:

Gegeben: Zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$.

Frage: Gibt es Teilmengen $V \subseteq V_1$ und $E \subseteq E_1$ derart, dass $|V| = |V_2|$ und $|E| = |E_2|$ gilt und eine bijektive Abbildung $f : V_2 \to V$ existiert mit $\{u, v\} \in E_2$ genau dann, wenn $\{f(u), f(v)\} \in E$.

1. Zeigen Sie, dass das Problem in NP liegt, indem Sie einen NP-Algorithmus angeben, der das Problem löst.

2. Beweisen Sie, dass das Problem NP-hart (und damit insgesamt NP-vollständig) ist, indem Sie eine Reduktion von einem Ihnen bekannten NP-vollständigen Problem angeben.