AG Theorie der künstlichen Intelligenz
FB Mathematik und Informatik, Universität Bremen
Prof. Dr. Carsten Lutz

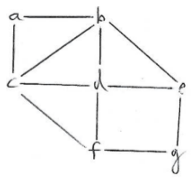
MZH 3090
clu@uni-bremen.de
Tel.: 0421/218-64431

1. Aufgabenblatt für die Vorlesung „Komplexitätstheorie“

Aufgabe 1: 20%

Betrachte den Algorithmus b-clique aus Kapitel 1 der Vorlesung.

(a) Wende den Algorithmus auf folgenden Graphen G und Cliquengröße 3 an:



(b) Zeige durch Angeben eines Gegenbeispieles, dass das Eliminieren adjazenter Knoten notwendig ist, weil der Algorithmus sonst nicht korrekt ist.

Aufgabe 2: 10%

Das Rucksackproblem ist als Optimierungsproblem wie folgt formuliert, wobei alle Zahlen *binär kodiert* sind:

Gegeben:

- eine Menge $A = \{a_1, \ldots, a_n\}$ von Gegenständen, und für jeden Gegenstand a_i ein Gewicht $g_i \in \mathbb{N}$ und einen Nutzen $n_i \in \mathbb{N}$ sowie
- eine Gewichtsgrenze $G \in \mathbb{N}$ für einen zu packenden Rucksack.

Ausgabe: nutzenmaximale Rucksackfüllung d.h. Teilmenge $R \subseteq M$ so dass $\Sigma_{a_i \in R}\, g_i \leq G$ und $\Sigma_{a_i \in R}\, n_i$ maximal.

Berechne die Lösung für folgende Eingabe:

$$A = \{a_1, a_2, a_3, a_4, a_5\} \qquad (n_1, n_2, n_3, n_4, n_5) = (3, 8, 3, 6, 2) \qquad (g_1, g_2, g_3, g_4, g_5) = (2, 6, 2, 4, 3) \qquad G = 8.$$

Argumentiere, dass die Lösung wirklich optimal ist.

Aufgabe 3: 25%

In der Variante als Berechnungsproblem ist beim Rucksackproblem zusätzlich ein Zielnutzen N gegeben, der mindestens erreicht werden muß. Ausgegeben wird eine Rucksackfüllung, die diesen Zielnutzen realisiert.
Zeige: wenn das Berechnungsproblem in Polynomialzeit lösbar ist, dann auch das Optimierungsproblem.
(Hinweis: es ist nicht möglich, alle Werte von 0 bis $\sum_{a_i \in A} n_i$ durchzugehen, da die Werte n_i in binär gegeben sind.)

Aufgabe 4: 25%

In der Variante als Entscheidungsproblem ist beim Rucksackproblem ebenfalls ein Zielnutzen N gegeben und die Frage ist, ob eine Rucksackfüllung *existiert*, die diesen Zielnutzen realisiert. Die Ausgabe ist (wie bei jedem Entscheidungsproblem) “ja” oder “nein”.
Zeige: wenn das Entscheidungsproblem in Polynomialzeit lösbar ist, dann auch das Berechnungsproblem.

Aufgabe 5: 20%

Entwickle eine (deterministische) Turingmaschine, die als Eingabe $\$\mathsf{bin}(n)$ erhält, wobei $\mathsf{bin}(n)$ die binäre Kodierung der Zahl n ist, und die diese dann inkrementiert. Es wird angenommen, dass das höchstwertige Bit ganz links steht und das niederstwertige ganze rechts. Beim Start steht der Kopf der Maschine auf dem Symbol \$. Dieses darf beim Inkrementieren wenn nötig überschrieben werden.

Gib die Übergänge in graphischer Form an (wie in der Vorlesung). Erkläre die Konstruktion. Gib die Berechnung der TM auf der Eingabe \$111 an.

Aufgabe 6: 25% (Zusatzaufgabe)

Beweise die Korrektheit des Algorithmus aus Aufgabe 1. Zeige, dass für alle Knoten $v \in V$ folgendes gilt: v wird markiert gdw. v von v_0 erreichbar ist.

Hinweis: Verwende für die “$\Rightarrow$” Richtung Induktion über die Anzahl Schritte der while Schleife und für die “$\Leftarrow$” Richtung Induktion über die Länge des kürzesten Pfades von v nach v_0.