Lehrstuhl für Informatik I
Informationsstrukturen und
wissensbasierte Systeme
Universität Würzburg

Prof. Dr. H. Noltemeier
Dr. S. O. Krumke

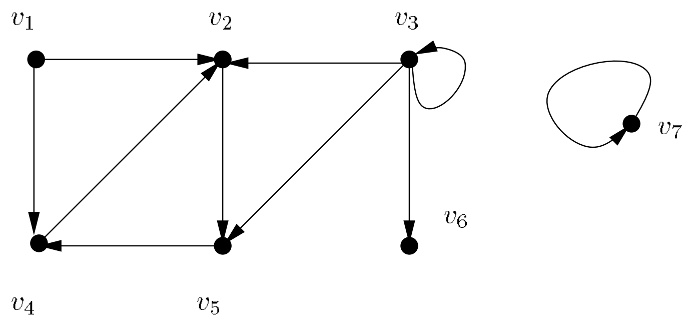
Würzburg, den 21. November 1997

Informationsstrukturen und wissensbasierte Systeme
(WS 97/98)

1. Übung

Aufgabe 1: (Grundlegende graphentheoretische Konzepte)

Gegeben sei der folgende gerichtete Graph $G = (V, R)$.



(a) Wenden Sie den *Depth-First-Search-Algorithmus* auf den oben gezeichneten Graphen G an. Beginnen Sie dabei mit der Ecke v_1. Geben Sie an, in welcher Reihenfolge die Ecken durch Depth-First-Search betrachtet werden und geben sie den bei Abbruch entstandenen Depth-First-Search-Wald an. **(2 Punkte)**

(b) In welcher Reihenfolge werden die Ecken von G durch den *Breadth-First-Search-Algorithmus* besucht, wenn man dabei ebenfalls mit v_1 startet? **(2 Punkte)**

(c) Geben Sie für den oben gezeichneten Graphen G alle starken Zusammenhangskomponenten an. **(1 Punkt)**

(d) Wie sehen die schwachen Zusammenhangskomponenten des Graphen G aus? **(1 Punkt)**

Aufgabe 2: (Erfüllbarkeitsprobleme in der Aussagenlogik)

Es sei $X = \{A_1, \dots, A_n\}$ eine Menge von atomaren Formeln. Man nennt A_i und $\neg A_i$ *Literale*. Das Literal A_i ist *positiv*, das Literal $\neg A_i$ *negativ*.

Eine *Belegung* $\mathcal{B}$ für die Atomformeln in X ist eine Funktion $\mathcal{B} \colon X \to \{0, 1\}$. Gilt $\mathcal{B}(A) = 1$, so sagen wir, daß A *unter* $\mathcal{B}$ *wahr* ist, ansonsten nennen wir A *unter* $\mathcal{B}$ *falsch*. Das Literal A ist genau dann wahr unter $\mathcal{B}$, wenn A unter $\mathcal{B}$ wahr ist. Das Literal $\neg A$ ist genau dann wahr unter $\mathcal{B}$, wenn A unter $\mathcal{B}$ falsch ist.

Eine *Klausel* ist eine Menge von Literalen über X, z.B. $C = \{A_1, \neg A_2, A_5\}$. Eine Klausel C repräsentiert die Disjunktion dieser Literale. Eine Klausel C wird durch die Belegung $\mathcal{B}$ *erfüllt* genau dann, wenn mindestens eines der Literale aus C unter $\mathcal{B}$ wahr ist. Eine Menge M von Klauseln heißt *erfüllbar*, wenn es eine *Belegung* gibt, die alle Klauseln in M erfüllt. Solch eine Belegung nennt man dann eine *erfüllende Belegung* für M.

Eine Instanz (X, M) des *Erfüllbarkeitsproblems* SAT ist durch eine Menge X von Atomformeln und eine Menge M von Klauseln über X gegeben. Das Problem besteht darin zu entscheiden, ob M erfüllbar ist. Mit k-SAT bezeichnen wir die Teilmenge der Instanzen von SAT, in der jede Klausel *genau* k Literale besitzt.

Das Problem SAT ist ein klassisches NP-vollständiges Problem. Bereits die Einschränkung 3-SAT ist NP-vollständig. Von 2-SAT weiß man hingegen, daß dieses Problem in Polynomialzeit gelöst werden kann.

Das Problem MAX SAT besteht darin, zu einer Menge X von Atomformeln und einer Menge M von Klauseln über X eine Belegung $\mathcal{B}$ zu finden, die die größtmögliche Anzahl von Klauseln erfüllt. MAX k-SAT besteht aus den Instanzen von MAX SAT, bei denen jede Klausel genau k Literale enthält.

(a) Ist die Menge $M = \{C_1, C_2, C_3, C_4\}$ mit $C_1 = \{A_1, \neg A_1, \neg A_2\}$, $C_2 = \{A_3, A_2, A_4\}$, $C_3 = \{\neg A_1, \neg A_3, \neg A_4\}$, $C_4 = \{A_2, \neg A_1, \neg A_3\}$ erfüllbar? **(2 Punkte)**

(b) Sei (X, M) eine beliebige Instanz von MAX k-SAT mit $M = \{C_1, \dots, C_m\}$. Wir betrachten eine zufällige Belegung $\mathcal{B} \colon X \to \{0, 1\}$ der Variablen aus X. Wie groß ist die Wahrscheinlichkeit, daß $\mathcal{B}$ die Klausel C_i erfüllt? Wie groß ist der Erwartungswert der Anzahl der erfüllten Klauseln? **(3 Punkte)**

(c) Zeigen Sie, daß es zu jeder Klauselmenge $M = \{C_1, \dots, C_m\}$ mit m Klauseln eine Belegung $\mathcal{B}$ gibt, die mindestens $m/2$ Klauseln erfüllt. Geben Sie einen Algorithmus an, der in $\mathcal{O}(n + \sum_{i=1}^{m} |C_i|)$ Zeit eine solche Belegung findet. **(2 Punkte)**

Aufgabe 3: (Hillclimbing für MAX CUT)

Eine Instanz des Problems MAX CUT ist durch einen endlichen ungerichteten Graphen $G = (V, E)$ gegeben. Das Problem besteht darin, die Eckenmenge V in zwei Mengen S und $V \setminus S$ zu partitionieren, so daß die Anzahl der Kanten, die einen Endpunkt in S und den anderen in $V \setminus S$ haben, maximiert wird.

Wir betrachten nun das Hillclimbing-Verfahren aus der Vorlesung zur approximativen Lösung von MAX CUT. Als Zustandsmenge Z wählen wir

$$Z := \{ S : S \subseteq V \}.$$

Für einen Zustand $z = S$ besteht die Menge seiner Nachfolger aus alle denjenigen Zuständen, die durch Hinzufügen oder Entfernen einer einzelnen Ecke aus S entstehen:

$$N(z) := \{ S \cup \{v\} : v \in V \setminus S \} \cup \{ S \setminus \{v\} : v \in S \}.$$

Die Bewertung $\varphi(z)$ eines Zustandes $z = S$ ist die Anzahl der Kanten zwischen S und $V \setminus S$.

Der Hillclimbing-Algorithmus werde für diese Aufgabe so modifiziert, daß er vom aktuellen Zustand z nur dann in einen Folgezustand $z' \in N(z)$ übergeht, wenn $\varphi(z') > \varphi(z)$ gilt.

(a) Zeigen Sie, daß der (modifizierte) Hillclimbing-Algorithmus nach endlich vielen Schritten terminiert, aber nicht notwendigerweise eine optimale Lösung findet. **(2 Punkte)**

(b) Sei OPT der Wert der optimalen Lösung $(S^*, V \setminus S^*)$. Beweisen Sie, daß für den Zustand z_t, der vom (modifizierten) Hillclimbing-Algorithmus bei Abbruch geliefert wird, gilt: $\varphi(z_t) \geq \mathrm{OPT}/2$. **(4 Punkte)**

Hinweis: Sei $z_t = S$ die vom Algorithmus gelieferte Lösung. Betrachten sie die Partitionierung von V in $V = V_1 \cup V_2 \cup V_3 \cup V_4$, so daß $S = V_1 \cup V_2$, $V \setminus S = V_3 \cup V_4$ und $(S^*, V \setminus S^*) = (V_1 \cup V_3, V_2 \cup V_4)$ gilt. Seien e_{ij} die Anzahl der Kanten mit einem Endpunkt in V_i und dem anderen in V_j. Es gilt dann

$$\begin{aligned} \mathrm{opt} &= e_{12} + e_{34} + e_{14} + e_{23} \\ \varphi(z_t) &= e_{13} + e_{14} + e_{23} + e_{24}. \end{aligned}$$

Zeigen Sie zunächst die Ungleichung $2e_{11} + e_{12} \leq e_{13} + e_{14}$ und folgern sie daraus $e_{12} \leq e_{13} + e_{14}$. Beweisen Sie dann analoge Ungleichungen für e_{34}, e_{14} und e_{23}.

Abgabetermin: Dienstag, 18. November 1997, bis 10^{00} Uhr in den dafür vorgesehenen Kästen vor der Teilbibliothek Mathematik/Informatik

Hinweise: Bis zu zwei Übungsteilnehmer dürfen gemeinsam ein Übungsblatt abgeben.

Die erste Übung findet am Freitag, den 21. November, statt. Die Gruppeneinteilung entnehmen Sie bitte den Aushängen.