Prof. Dr. M. Chimani
Dipl.-Inf. S. Beyer
Dipl.-Math. I. Hedtke

Universität Osnabrück
Theoretische Informatik
Sommersemester 2014

Übungsblatt 11 zur
Informatik Ⴀ: Einführung in die Theoretische Informatik

Ausgabe: 11. Juli — Besprechung: 21.–23. Juli

In der Vorlesung am 17. Juli 2014 werden Sie das Problem HAMILTONKREIS und den Beweis für dessen *NP*-Vollständigkeit gesehen haben. Da das Resultat für dieses Übungsblatt bereits benötigt wird, finden Sie hier die Definition des Problems:

> **Gegeben:** ein ungerichteter Graph $G = (V, E)$
> **Gefragt:** Gibt es eine Rundtour, die jeden Knoten genau einmal besucht?

Aufgabe 11.1. Die Kreise von Euler und Hamilton

Wir betrachten das Entscheidungsproblem EULERKREIS:

> **Gegeben:** ein ungerichteter Graph $G = (V, E)$
> **Gefragt:** Gibt es einen Kantenzug, der alle Kanten von G genau einmal besucht und am Anfangsknoten endet? (Knoten dürfen dabei mehrmals besucht werden.)

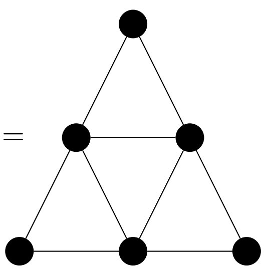
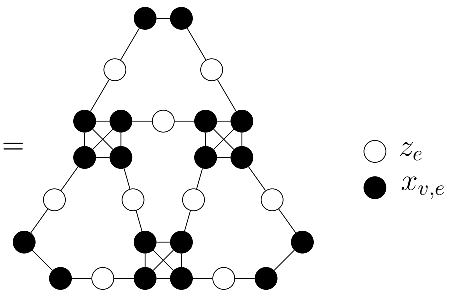
Sie kennen aus der Vorlesung *Informatik A* bereits einen Algorithmus, der einen Eulerkreis in einem beliebigen Graphen in polynomieller Zeit finden kann.

Betrachten Sie den folgenden Beweis:

Theorem. EULERKREIS ist *NP*-vollständig.

Beweis. Es ist trivial, dass EULERKREIS in *NP* liegt. Wir wissen bereits, dass HAMILTONKREIS *NP*-vollständig ist.

Sei G ein beliebiger Eingabegraph für EULERKREIS. Wir konstruieren einen Graphen $G' = (V', E')$ für HAMILTONKREIS wie folgt: Für jede Kante $e \in E$ mit $e = \{u, w\}$ erzeugen wir in G' drei Knoten $\{x_{u,e}, z_e, x_{w,e}\}$ und die Kanten $\{x_{u,e}, z_e\}$ und $\{z_e, x_{w,e}\}$. Für alle Knoten $x_{v,e}, x_{v,f} \in V'$ mit $e \neq f$ erzeugen wir eine Kante $\{x_{v,e}, x_{v,f}\}$.

Beispiel: $G =$  $\longrightarrow$ $G' =$ 

G' kann deterministisch in polynomieller Zeit erzeugt werden.

Wir müssen noch die Korrektheit der Reduktion zeigen, also das G einen Eulerkreis hat, gdw. G' einen Hamiltonkreis hat.

„$\Rightarrow$“: Sei K der Eulerkreis in G. Wir konstruieren ein K' für G': für jede Kante $e = \{u, w\}$ in K wählen wir die Kanten $\{x_{u,e}, z_e\}, \{z_e, x_{w,e}\}$ in K' hinein; folgt auf eine Kante $e = (u, v)$ direkt die Kante $f = (v, w)$ in K, so wählen wir die Kante $\{x_{v,e}, x_{v,f}\}$ in K' hinein. Damit ist K' ein Hamiltonkreis in G'.

„$\Leftarrow$“: Sei K' ein Hamiltonkreis in G'. Sei K der Kantenzug, der entsteht, wenn wir die zum Knoten z_e in K' zugehörige Kante e in K hineinwählen. Weil K' Kanten der Form $\{x_{v,e}, x_{v,f}\}$ benutzt, ist v der gemeinsame Knoten der Kanten e und f in K. Deshalb ist K ein gültiger Kantenzug. Da in K' jeder Knoten z_e genau einmal besucht wird, wird in K jede Kante genau einmal besucht. Damit ist K ein Eulerkreis in G. □

Was ist hier passiert? Folgt $\mathsf{P} = \mathsf{NP}$? Ist der Algorithmus aus Informatik A nur eine Heuristik? Wurde in Informatik A ein anderes Maschinenmodell benutzt? ...oder ist der Beweis etwa falsch? Erklären Sie die Situation.

Aufgabe 11.2. *NP*-Vollständigkeit: TSP

Das Traveling Salesperson Problem (TSP) ist wie folgt definiert:

Gegeben: ein ungerichteter, vollständiger Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{N}^+$ und eine natürliche Zahl K

Gefragt: Gibt es eine Rundtour mit Kosten $\leq K$, die jeden Knoten aus V genau einmal besucht?

Beachten Sie, dass Kantenkosten positiv sind, also nicht 0 sein dürfen! Zeigen Sie, dass TSP *NP*-vollständig ist, indem Sie

(a) beschreiben, warum TSP in *NP* liegt,

(b) eine geeignete Reduktion angeben, aus der folgt, dass TSP *NP*-schwer ist, und

(c) die Korrektheit der angegebenen Reduktion beweisen.

Aufgabe 11.3. Clique

In der Vorlesung wurde gezeigt, dass Clique ein *NP*-vollständiges Problem ist. Sei $\mathcal{G}_c$ die Menge aller Graphen mit Maximalgrad c für eine Konstante c.

(a) Geben Sie einen Algorithmus an, der das Entscheidungsproblem mit der Eingabe (G, k) in *polynomieller Zeit* lösen kann, wobei G ein beliebiger Graph aus $\mathcal{G}_5$ ist.

(b) Wird das Problem wieder *NP*-vollständig, wenn wir es auf Graphen aus $\mathcal{G}_6$, $\mathcal{G}_7$ oder $\mathcal{G}_c$ für eine beliebige Konstante c beschränken?

Aufgabe 11.4. Nichterfüllbarkeit

Wir betrachten das Problem Non-Satisfiability:

Gegeben: eine Formel F der Aussagenlogik (in KNF)
Gefragt: Ist $F = \mathit{false}$ für alle Variablenbelegungen?

Der folgende Beweis, dass dieses Problem in *NP* liegt, ist falsch. Warum? Begründen Sie!

Beweis. Wir zeigen, dass es einen nicht-deterministischen polynomiellen Algorithmus `NonSat` gibt, der dieses Problem entscheidet. Sei `Sat` ein solcher Algorithmus für das Sat-Problem. Der folgende Algorithmus löst unser Problem:

```
NonSat(𝓘):
  bool result = Sat(𝓘)
  return !result
```

Ist `result` $= \mathit{false}$, so war `Sat` nicht in der Lage einen Zeugen für $F = \mathit{true}$ zu finden, also gilt für alle Variablenbelegungen $F = \mathit{false}$. Schließlich hätte `Sat` ja richtig geraten, wenn es eine *true*-Belegung geben würde. □

Aufgabe 11.5. Asterix und *NP*

Wir betrachten das folgende praxisnahe Problem:

> Obelix wirft Hinkelsteine auf n verschiedene Haufen. Ein Haufen bietet Platz für maximal drei Hinkelsteine. Miraculix färbt diese Hinkelsteine während des Fluges mit je einer aus m verschiedenen Farben so ein, dass keine Farbe in einem Haufen mehrfach vorkommt. Asterix stellt sich die Frage: Gibt es eine Teilmenge $F_{\text{schön}}$ der Farben, sodass auf jedem Haufen genau ein Hinkelstein schöner Farbe liegt?

(a) Formalisieren Sie das Problem. Vervollständigen Sie dazu den letzten Satz:

Sei $F = \{f_1, \ldots, f_m\}$ die Menge aller Farben. Seien $H_1, \ldots, H_n \subseteq F$ Mengen mit je höchstens drei Elementen. Gibt es eine Teilmenge $F_{\text{schön}} \subseteq F$ der Farben, sodass ...

(b) Wir wollen das Problem aus **(a)** als SAT-Instanz darstellen. Ob eine Farbe f_i in $F_{\text{schön}}$ liegt, können Sie durch eine boolesche Variable x_i modellieren. Wir führen den neuen Klauseltyp $[\![x \vee y \vee z]\!]$ ein, der $(x \vee y \vee z)$ entspricht, wobei aber nur eines von x, y, z gleich *true* sein darf. Das gesamte Problem lässt sich modellieren, indem für jede Menge H_i eine Klausel des neuen Typs aufgestellt wird, und alle am Ende durch Konjunktion verbunden werden.

Ihre Aufgabe: Wie sieht die Klausel des neuen Typs für den Haufen $H_i = \{f_j, f_k, f_\ell\}$ aus?

(c) Zeigen Sie, dass das Problem aus **(a)**, also das Asterix-Problem, *NP*-vollständig ist.

Hinweise:

- Verwenden Sie, dass 3-SAT *NP*-vollständig ist.
- Sie können negierte Variablen in Termen entfernen, indem Sie für jede vorkommende Variable x_i eine weitere Variable x_i' einführen, jedes $\neg x_i$ durch x_i' ersetzen und mittels $[\![x_i \vee x_i']\!]$ dafür sorgen, dass nur eines von x_i und $\neg x_i$ auf true gesetzt werden kann.
- Ein Term $(x_1 \vee x_2 \vee x_3)$, der keine negierten Variablen enthält, kann mittels 4 neuer Variablen x_a, x_b, x_c, x_d so dargestellt werden, dass in jedem neuen Term nur genau eine Variable true ist: $[\![x_1' \vee x_a \vee x_b]\!] \wedge [\![x_2 \vee x_b \vee x_c]\!] \wedge [\![x_3' \vee x_c \vee x_d]\!]$.

Aufgabe 11.6. KGB (Kursgruppenbesetzung)

Das vereinfachte KGB-Problem ist wie folgt definiert:

Gegeben n Studenten, m Kurstermine und eine Zahl k. Jeder Student gibt eine nicht-leere Teilmenge der Termine an, zu denen er einen Kurs besuchen könnte. Die Frage ist, ob es eine Auswahl von k Terminen gibt, so dass es jedem Studenten möglich ist (mindestens) einen der ausgewählten Kurse zu besuchen. In der Realität haben die Kurse noch eine maximale Teilnehmeranzahl. Diese Einschränkung fällt hier weg.

Im Gegensatz zu Ihren (armen) Übungsleitern müssen sie das Problem nicht lösen, sondern nur dessen Komplexität ermitteln:

(a) Recherchieren Sie das Problem SETCOVER. Beschreiben Sie das Entscheidungsproblem und geben Sie seine Komplexität an.

(b) Ermitteln und beweisen Sie die Komplexität vom vereinfachten KGB unter Verwendung von (a).

จงพากเพียรไปเถิดจักเกิดผล
งานเลี้ยงต้องมีวันเลิกรา