

Informatik CI: – Blatt 11

Rasmus Diederichsen

15. Juli 2014

Aufgabe 11.1

Der Beweis ist Bullshit. Es wird reduziert von dem in Polynomialzeit lösbaren Problem EULERKREIS auf das Problem HAMILTONKREIS. Dies zeigt aber nur, dass HAMILTONKREIS mindestens so schwer ist wie EULERKREIS, was nicht überrascht, da HAMILTONKREIS *NP*-vollständig ist. Um zu zeigen, dass EULERKREIS *NP*-vollständig ist, müsste man aber zeigen, dass sich HAMILTONKREIS in polynomiell-deterministischer Zeit auf EULERKREIS reduzieren lässt. Dann wäre bewiesen, dass EULERKREIS mindestens so schwer ist wie HAMILTONKREIS.

Aufgabe 11.2

a)

Gegeben einen Zeugen für die Lösbarkeit einer Instanz (also einen Weg, dessen Länge angeblich $\leq k$ ist und der alle Knoten genau einmal enthält), ist es einfach in polynomieller Zeit, nachzuprüfen ob

1. er wirklich kumulierte Kosten $\leq k$ hat (in Linearzeit prüfbar)
2. er alle Knoten enthält (ebenfalls in Linearzeit prüfbar)

b)

Wie wissen (s.o), dass HAMILTONKREIS *NP*-vollständig ist. Wir müssen also zeigen, dass sich jede Problem Instanz von HAMILTONKREIS auf eine Instanz von TSP reduzieren lässt.

Gegeben einen Graphen $G = (V, E)$, $|V| = n$, für den ein Hamilton-Kreis gefunden werden soll, können wir die Problemstellung umformulieren. Sei $G' = (V', E')$ ein vollständiger Graph mit $V' = V$ und $E' = \{(u, v) \mid u, v \in V'\}$. Es sind also alle Vertices miteinander verbunden, auch wenn es zwischen ihnen in G keine Kante gab. Die Kantenkosten seien durch die Funktion

$$d(e = (u, v) \in E') = \begin{cases} 1 & \text{falls } (u, v) \in E \\ 2 & \text{sonst} \end{cases}$$

gegeben. Die Fragestellung lautet nun

Gibt es einen Weg in G' , der jeden Knoten genau einmal besucht und dessen kumulierte Kosten (\leq) n sind?

c)

Es bleibt zu zeigen

G enthält einen Hamiltonkreis $\iff G'$ enthält eine Rundtour $\leq n$.

“ \Rightarrow ” Falls G einen Hamiltonkreis besitzt, dann enthält dieser jeden Knoten genau einmal und derselbe Kreis existiert in G' . Da in G' alle Kanten aus G die Kosten 1 besitzen, hat die Rundtour in G' die Kosten maximal n , da maximal n Kanten zwischen n Knoten besucht werden können.

“ \Leftarrow ” Wenn es in G' eine Rundtour mit Kosten $\leq n$ gibt, so sind die kumulierten Kosten die Summe aus n Kantenkosten (damit ein Kreis entstehen kann). Es muss also jede Kante die Kosten 1 haben, da ansonsten $\sum_{i=1}^n c_i \leq n$ nicht erfüllbar ist mit $c \in \{1, 2\}$. Dies wiederum bedeutet aber (nach Definition von d oben), dass derselbe Kreis auch in G existiert.

Aufgabe 11.3

a)

Das Problem lässt sich in Polynomialzeit per Brute-Force-Search lösen.

Algorithmus für CLIQUE-Lösung

```
1 def find_clique(graph, k):
2     for g in graph.subgraphs:
3         for (v1, v2) in [(u,v) for u in g.vertices for v in g.
4                           vertices if u != v]:
5             if !g.hasEdge(v1, v2):
6                 return false
7         return true, g
```

b)

Es gibt in einem Graphen mit Maximalgrad k und Knotenzahl n nur n^k viele Teilgraphen, die theoretisch eine Clique bilden können, da diese nur k Knoten enthalten kann (weil sie vollständig verbunden sein muss). In der Clique muss jeder der k Knoten k Kanten besitzen und diese müssen ihn mit den anderen Knoten in der Clique verbinden. Um die Verbundenheit zu prüfen, muss man

also k^2 Kanten betrachten. Der Algorithmus läuft also in $\mathcal{O}(n^k k^2)$, in diesem Fall $\mathcal{O}(n^5)$.

Für jede Konstante k läuft der Algorithmus aber immer in Polynomialzeit.