Lösungen zur dritten Miniklausur in Informatik III

Aufgabe 1 **5 Punkte**

Das Problem SUB-GRAPH ist wie folgt definiert:

- Eingabe: Zwei ungerichtete Graphen G und G'
- Frage: Ist G' ein Teilgraph von G?

Ein Graph $G' = (V', E')$ ist ein Teilgraph von $G = (V, E)$, wenn es eine injektive Abbildung $f : V' \to V$ gibt, so dass $\{u, v\} \in E' \Longrightarrow \{f(u), f(v)\} \in E$.

Zeigen Sie, dass SUB-GRAPH NP-vollständig ist.

SUB-GRAPH ist in NP:
Es gibt einen Verifzierer für SUB-GRAPH. Hierzu wird die Abbildung $f : V' \to V$ geraten. Der Verifizierer überprüft nun für jede Kante $\{u, v\}$ aus E', ob die Kante $\{f(u), f(v)\} in E$ ist. Die Laufzeit dieses Algorithmus ist höchstens quadratisch in der Eingabelänge und damit polynomiell zeitbeschränkt.

SUB-GRAPH ist NP-schwierig:
Laut Vorlesung ist CLIQUE NP-vollständig. Wenn nun CLIQUE $\leq_{m,p}$ SUB-GRAPH gilt, dann folgt, dass SUB-GRAPH NP-schwierig ist.

Beweis der Reduktion CLIQUE $\leq_{m,p}$ SUB-GRAPH
Reduktionsfunkion: $f(G, k) = (G_k, G)$, wobei G_k der vollständige Graph aus k Knoten ist.
Beweis der Korrektheit:

1. Fall $(G, k) \in$ CLIQUE. Dann besitzt G eine Clique der Größe k. Dann ist G_k ein Teilgraph von G und daraus folgt $(G_k, G) \in$ SUB-GRAPH.
2. Fall $(G_k, G) \in$ SUB-GRAPH. Dann besitzt G eine k-Clique und damit ist $(G, k) \in$ CLIQUE.

Laufzeit der Reduktion
Um einen vollständigen Graph der Größe k zu konstruieren, wird quadratische Laufzeit in Abhängigkeit $k \leq |V|$ benötigt.

Aufgabe 2

1. NPSPACE = $\bigcup_k$ NSPACE(n^k) ist abgeschlossen unter Komplement.
   2 Punkte

   - [x] Die Aussage ist korrekt.
   - [ ] Die Aussage ist falsch.
   - [ ] Bis jetzt ist die Richtigkeit unklar. Aus der Aussage folgte aber, dass ____ = ____, was dramatische Konsequenzen hätte.

   Beweis:

   Nach dem Satz von Savitch ist NPSPACE = PSPACE. Die Klasse PSPACE ist unter Komplement abgeschlossen. Um eine Komplementärsprache auf einer DTM mit den gleichen Platzbeschränkungen zu konstruieren, muss man nur den akzeptierenden mit dem verwerfenden Zustand vertauschen.

2. NP ist unter der Stern-Operation abgeschlossen, d.h. falls $L \in$ NP dann ist auch $L^* \in$ NP.
   4 Punkte

   - [x] Die Aussage ist korrekt.
   - [ ] Die Aussage ist falsch.
   - [ ] Bis jetzt ist die Richtigkeit unklar. Aus der Aussage folgte aber, dass ____ = ____, was dramatische Konsequenzen hätte.

Beweis:

Gegeben sei ein Polynomzeit-Verifizierer M für L.

$$L^* = \{w_1 w_2 \dots w_m \mid w_i \in L \text{ für alle } i\}$$

Folgender **Polynomzeit-Verifizierer** M' löst nun L'. Der Ratestring c wird erweitert mit der Folge $(n_1, c_1, n_2, c_2, \dots, n_m, c_3)$.
Für die Eingabe w überprüft M' zuerst ob $\sum_{i=1}^{m} n_i = |w|$ ist. Jetzt wird w gemäß dieser Folge in die Zeichenketten $w_1, w_2, \dots, w_m$ mit $|w_i| = n_i$ für alle i und $w = w_1 w_2 \dots w_m$ unterteilt. Dann verifiziert M' für alle $i \in \{1, \dots, m\}$ mit Hilfe von M und dem Ratestring c_i, ob das Teilwort w_i in L ist. Falls alle Tests positiv sind, wird das Wort w akzeptiert.
Die **Laufzeit** von M' ist höchstens um den Fakter $|w|$ größer als die von M' und damit noch polynomiell.
Zur **Korrektheit**: Falls $w \in L^*$ dann kann eine solche Unterteilung gefunden werden und die Korrektheit der Teilworte verifiziert werden. Falls M' ein Wort w verifiziert, dann ist $w \in L^*$, da M' eine Unterteilung gefunden hat, so dass jedes Teilwort in L ist.

Aufgabe 3

4 Punkte

Angenommen, es gilt P $=$ NP. Beweisen Sie, dass dann ein polynomiell zeitbeschränkter Algorithmus existiert, der für jede erfüllbare Boolesche Formel eine erfüllende Belegung findet.

Sei $F(x_1, \ldots, x_n)$ die gegebene erfüllbare Boolesche Funktion. Ferner sei M die DTM, welche SAT in polynomieller Zeit entscheidet.
Folgender Algorithmus löst nun das Problem:

for $i = 1$ to n do
    $c_i \leftarrow 0$
    Sei F' die Funktion F, wobei $x_1, \ldots, x_i$ durch $c_1, \ldots, c_{i-1}, \neg c_i$ ersetzt worden sind.
    Teste mit M, ob $F' \in$ SAT
    Falls F' nicht erfüllbar, dann $c_i \leftarrow 1$
end do

Falls der erste Test scheitert, dann muss F mit der Belegung $c_1, \ldots, \neg c_i$ erfüllbar sein.
Die Korrektheit folgt über eine Induktion über die Anzahl der Variablen.
Die Laufzeit des Algorithmus ist die Laufzeit von M multipliziert mit der Anzahl der Variablen. Daher ist dieser Algorithmus polynomiell zeitbeschränkt.