GTI Übungsblatt 12
Tutor: Marko Schmellenkamp
ID: MS1
Übung: Mi 16-18

Max Springenberg, 177792

12.1

12.1.1

z.z.: $G_{RAPH}I_{SOMORPHIE} \in NP$

Nach Vorlesung hatten wir die Zugehörigkeit von Problemen wie folgt definiert:

1. es existiert ein Suchraum von Lösungen
2. jede Lösung ist polynomiell groß in ihrer Eingabe
3. jede Lösung kann in polynomieller Zeit überprüft werden

Der Suchraum umfasst Bijektionen von Graph G nach G', oder TUringmaschinen die eine Bijektion umsetzten.

Diese Funktionen, oder auch Turingmaschinen sind polynomiell in ihrer Eingabegröße, da für jede Kante und jeden Knoten je ein Fall, oder ein Zustand benötigt wird, der die Jeweiligen variablen umbenennt.
Insbesondere ergibt sich daraus, dass Die Funktion oder auch TM durch den Betrag an Kanten und Knoten mit $O(|V| + E|)$ eingeschränkt werden kann.

Der Test eines Lösungskandidaten kann wie folgt getestet werden.
$G = (V, E), G' = (V', E')$

1. $marked = \emptyset$
2. FOR $v \in V$ DO
3. $\quad marked = marked \cup \{v\}$
4. $\quad$ FOR $u \in adj(v) \cap marked$ DO
5. $\quad\quad$ IF $\neg(f(u) \in adj(f(v)))$ THEN
6. $\quad\quad\quad$ return false
7. $\quad\quad marked = marked \cup \{u\}$
8. return trü

Der algorithmus testet für jeden Knoten, allen adjazenten Knoten in adj auch nach anwenden der Bijektion noch adjazent sind. Und umgekehrt.
Wenn dies der Fall ist wurde die Kantenrelation eingehalten sonst nicht.
Die umkehrung der codierung einer bijektiven Funktion, oder einer TM kann polynomiell berechnet werden.
Insbesondere ist der Test durch die Anzahl an Kanten begrenzt.
Der Algorithmus hat also eine Laufzeitschranke von $O(|E|)$.

12.1.2

Eine aussagenlogische Formel φ bestehe aus k Klauseln und n Literalen, mit $k, n \in \mathbb{N}$
Da eine Klausel aus Verorderungen besteht ist diese Bereits erfüllt, sobald eine Belegung eine in der Klausel enthaltenem Variable so belegt, dass das Literal positiv ist, ist die KLausel erfüllt.
Bereits erfüllte Klauseln sind nicht mehr ausschlaggeben für die Erfüllbarkeit von φ.

f Sei gegeben durch

$$f(\varphi, \alpha) = \varphi'$$

Wobei φ' (i) nur Klauseln aus φ enthält, die unter Belegung α nicht schon erfüllt sind und (ii) deren Klauseln nur noch Variablen enthalten, die nicht im Definitionsbereich von α liegen.

12.2

12.2.1

Betrachte $I = \{2, 3, 4\}$ mit:
$\bigcup_{i \in I} S_i = \{a, b, c, d, h\}$
$S_2 \cap S_3 = \emptyset$
$S_2 \cap S_4 = \emptyset$
$S_3 \cap S_4 = \emptyset$

I ist eine entsprechende Indexmenge.

12.2.2

$f(M, (S_1, \ldots, S_4)) =$

$(i)(c_{1,e}x_1 + \ldots + c_{k,e}x_k \geq 1):$
$e = a:$
$x_1 + x_4 \geq 1$
$e = b:$
$x_2 + x_2 \geq 1$
$e = c:$
$x_2 \geq 1$
$e = d:$
$x_3 \geq 1$
$e = h:$
$x_3 \geq 1$

$(ii)(x_i + x_j \leq 1, \forall i \neq j : S_i \cap S_j \neq \emptyset):$
$i = 1, j = 2:$
$x_1 + x_2 \leq 1$
$i = 2, j = 1:$
$x_2 + x_1 \leq 1$
$i = 1, j = 4:$
$x_1 + x_4 \leq 1$
$i = 4, j = 1:$
$x_4 + x_1 \leq 1$

12.2.3

(i) $(c_{1,e}x_1 + \ldots + c_{k,e}x_k \geq 1)$:
Es muss für jede Variable eine Menge gewählt werden, in der diese Variable enthalten ist.
(ii) $(x_i + x_j \leq 1, \forall i \neq j : S_i \cap S_j \neq \emptyset)$:
Es darf von Menge, die zueinander einen nicht leeren Schnitt haben nur eine der jeweiligen Mengen genommen werden.

12.2.4

12.2.5

(i) $(c_{1,e}x_1 + \ldots + c_{k,e}x_k \geq 1)$:
Es gibt $|M|$ Ungleichungen, deren Länge je durch $O(k)$ eingegrenzt werden kann.
Damit also in $O(|M| * k)$.
(ii) $(x_i + x_j \leq 1, \forall i \neq j : S_i \cap S_j \neq \emptyset)$:
Es kann maximal jede Menge zu jeder anderen Menge nicht disjunkt sein.
Daraus ergeben sich dann maximal $2 * k^2$ Ungleichungen deren Länge jeweils durch $O(1)$ beschränkt ist.
Damit also in $O(k^2)$

Insgesamt ist die Laufzeit der Funktion damit in $O(k^2 + |M| * k)$ und damit insbesondere polynomiell zur Größe der Eingaben.