



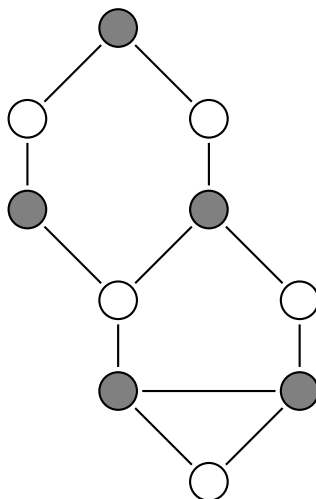
Hausaufgaben zur Vorlesung »Analyse von Algorithmen und Komplexität«

Blatt 11

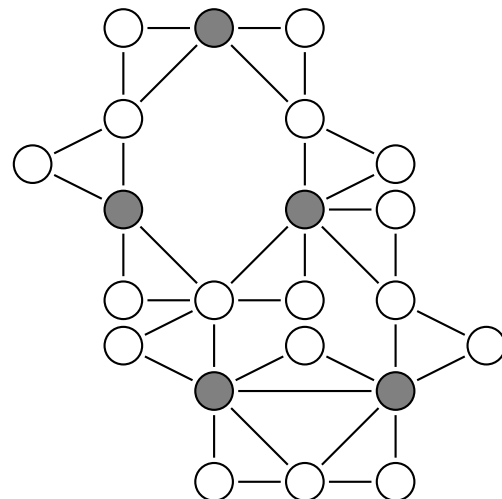
Definition 1 (Dreiecksüberdeckung eines Graphen (Δ COVER)).

Gegeben Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{Z}_{\geq 0}$.

Entscheide Besitzt G eine *Dreiecksüberdeckung* $C_\Delta \subseteq V$ mit $|C_\Delta| \leq k$,
d.h. gilt $D \cap C_\Delta \neq \emptyset$ für jede 3-Clique $D \subseteq V$ in G ?



(a) Ein Vertex Cover der Größe 5



(b) Eine Dreiecksüberdeckung der Größe 5

Abbildung 1: Zwei Beispielgraphen

Hausaufgabe 11.1

(5 Punkte)

Zeigen Sie: Δ COVER ist **NP**-vollständig.

Lösung. Wir zeigen zunächst, Δ COVER $\in$ **NP** mittels eines polynomiellen Verifizierers.

Der polynomielle Verifizierer erhält als Zertifikat eine Menge von Knoten $C_\Delta \subseteq V$. Zunächst überprüft er, ob $|C_\Delta| \leq k$. Falls dies nicht der Fall ist, verwirft die Instanz. Ansonsten überprüft für alle Teilgraphen $D = (V_D, E_D) \subseteq (V, E)$ mit $|V_D| = 3$ und $E_D = \{ \{v_1, v_2\} \in V_D \times V_D \mid v_1 \neq v_2 \}$ von G , ob für mindestens ein $v \in V_D$ auch $v \in C_\Delta$ gilt. Falls dies gilt, akzeptiere. Ansonsten verwirft.

Korrektheit: Der Verifizierer überprüft die geforderten Eigenschaften. Falls ein Graph eine Dreiecksüberdeckung besitzt, so existiert ein entsprechendes Zertifikat C_Δ und der Verifizierer akzeptiert die Instanz. Falls ein Graph keine Dreiecksüberdeckung besitzt, existiert kein valides entsprechendes Zertifikat C_Δ und der Verifizierer verwirft die Instanz.

Laufzeit: Es gibt maximal $|V|^3$ verschiedene Teilmengen, die ein Dreieck bilden könnten und daher überprüft werden müssen. Die Überprüfung geht in $O(|V|)$ Zeit. Da alle anderen Schritte des Verifizierers in $O(1)$ Zeit erledigt werden können, ergibt sich somit insgesamt eine Laufzeit von $O(|V|^4)$, was einer polynomiellen Laufzeit entspricht.

Also gilt $\Delta \text{ COVER} \in \mathbf{NP}$.

Nun reduzieren wir VERTEXCOVER auf $\Delta \text{ COVER}$. Betrachte folgende Reduktion:

Sei $I = (G = (V, E), k)$ eine Instanz von VERTEXCOVER. Bilde den Graphen $G' = (V', E')$ mit $V' = V \cup \{v_e \mid e \in E\}$ und $E' = E \cup \{\{v_e, e_1\}, \{v_e, e_2\} \mid e = \{e_1, e_2\} \in E\}$. Gebe $I' = (G', k)$ als $\Delta \text{ COVER}$ Instanz aus.

Korrektheit:

$\Rightarrow$ (Falls (G, k) eine Ja-Instanz von VERTEXCOVER ist, dann ist (G', k) eine Ja-Instanz von $\Delta \text{ COVER}$.):

Sei C ein Vertex Cover mit Größe kleiner oder gleich k von G . Dann ist C ebenfalls eine Dreiecksüberdeckung mit Größe kleiner oder gleich k . Offensichtlich gilt $|C| \leq k$. Da C ein Vertex Cover von G ist, gilt für jede Kante $\{u, v\} \in E$, dass $u \in C$ oder $v \in C$. Für alle Teilgraphen $D = (V_D, E_D) \subseteq (V, E)$ mit $|V_D| = 3$ und $E_D = \{\{v_1, v_2\} \in V_D \times V_D \mid v_1 \neq v_2\}$ von G gilt damit direkt, dass mindestens ein $v \in V_D$ auch in C enthalten ist. Der Graph G' besitzt neben allen solchen Teilgraphen in G zusätzlich solche Teilgraphen (Dreiecke), die durch die Knotenmenge $\{v_e \mid e \in E\}$ und die entsprechenden Kanten $\{e, \{v_e, e_1\}, \{v_e, e_2\} \mid e = \{e_1, e_2\} \in E\}$ entstehen. Für diese Dreiecke gilt aber auch, dass mindestens ein Knoten, nämlich e_1 oder e_2 , in C enthalten sein muss, da C ein Vertex Cover ist.

$\Leftarrow$ (Falls (G', k) eine Ja-Instanz von $\Delta \text{ COVER}$ ist, dann ist (G, k) eine Ja-Instanz von VERTEXCOVER.):

Sei C_Δ eine Dreiecksüberdeckung von G' der Größe $|C_\Delta| \leq k$. Ohne Beschränkung der Allgemeinheit können wir davon ausgehen, dass $C_\Delta \subseteq V$ gilt: Falls für einen Knoten $v_e \in C_\Delta$ gilt, können wir e_1 zu C_Δ hinzufügen und v_e aus C_Δ löschen. Dadurch bleibt die Überdeckungseigenschaft erhalten und die Menge wird höchstens kleiner. Nun gilt also $C_\Delta \subseteq V$. Die Menge C_Δ ist auch ein Vertex Cover mit Größe kleiner oder gleich k von G . Offensichtlich gilt $|C_\Delta| \leq k$. Da für alle Teilgraphen $D' = (V'_D, E'_D) \subseteq (V', E')$ mit $|V'_D| = 3$ und $E'_D = \{\{v_1, v_2\} \in V'_D \times V'_D \mid v_1 \neq v_2\}$ von G' gilt, dass mindestens ein Knoten $v \in V'_D$ ebenfalls in C_Δ enthalten sein muss, gilt dies auch für die Dreiecke $\{v_e, e_1, e_2\}$. Da außerdem $C_\Delta \subseteq V$ gilt, gilt $e_1 \in C_\Delta$ oder $e_2 \in C_\Delta$ für alle Kanten $e \in E$.

Laufzeit: Das Hinzufügen von einem neuen Knoten und zwei neuen Kanten pro Kante geht in $O(|E|)$. Damit geht die gesamte Reduktion in $O(|E|)$, also in polynomieller Zeit.

Also existiert eine polynomielle Reduktion von VERTEXCOVER auf $\Delta \text{ COVER}$. Da VERTEXCOVER $\mathbf{NP}$ -vollständig aus den Präsenzaufgaben bekannt ist und $\Delta \text{ COVER} \in \mathbf{NP}$ gilt, ist somit $\Delta \text{ COVER}$ $\mathbf{NP}$ -vollständig. $\square$

Bewertung (a) und (b) jeweils 1 Punkt, (c) gibt 8 Punkte: 1 Punkt Angabe Verifizierer/NDTM, 0,5 Punkte Korrektheit Verifizierer/NDTM, 0,5 Punkte Laufzeit Verifizierer/NDTM, 1 Punkt Angabe Reduktion, 3 Punkte Korrektheit (je Richtung 1,5 Punkte), 1 Punkt Laufzeit Reduktion, 1 Punkt Beweisführung

Definition 2 (Halteproblem auf DTMs). Eine DTM M hält auf dem Wort w , wenn die Berechnung von M auf w eine Stopkonfiguration erreicht. Für eine geeignete Kodierung $\langle \bullet, \bullet \rangle$ definieren wir:

$$\text{HALT}_{\text{TM}} := \{ \langle M, w \rangle \mid M \text{ ist DTM und } M \text{ hält auf dem Wort } w \}$$

Hausaufgabe 11.2

(5 Punkte)

Zeigen Sie: HALT_{TM} ist NP-schwer.

Lösung. Wir zeigen $\text{SAT} \leq \text{HALT}_{\text{TM}}$.

Sei φ ein Boolescher Ausdruck in KNF. Aus φ konstruieren wir eine DTM M_φ , die für eine Eingabe w wie folgt arbeitet:

1. Lösche das Band, d.h. ignoriere die Eingabe w .
2. Enumeriere systematisch alle Belegungen der Variablen von φ
3. Prüfe für jede dieser Belegungen, ob sie φ erfüllt.
4. Sofern darunter eine erfüllende Belegung gefunden wird, halte.
5. Ansonsten laufe für immer.

Die Transformation f definieren wir nun so, dass $f(\varphi) = \langle M_\varphi, \varepsilon \rangle$, wobei ε das leere Wort ist.

Die Maschine M_φ läuft zwar nicht in polynomieller Zeit - aber sie lässt sich in polynomieller Zeit konstruieren, weshalb f in polynomieller Zeit berechenbar ist.

Wir zeigen: $\varphi \in \text{SAT}$ gdw. $\langle M_\varphi, \varepsilon \rangle \in \text{HALT}_{\text{TM}}$

“ $\Rightarrow$ ”. Sei $\varphi \in \text{SAT}$. Dann ist φ erfüllbar, d.h. es existiert mindestens eine erfüllende Belegung von φ . Da M_φ alle Belegungen ausprobiert, findet sie eine erfüllende Belegung und hält, d.h. $\langle M_\varphi, \varepsilon \rangle \in \text{HALT}_{\text{TM}}$.

“ $\Leftarrow$ ”. Sei $\langle M_\varphi, \varepsilon \rangle \in \text{HALT}_{\text{TM}}$, d.h. M_φ hält auf ε . Die Maschine M_φ hält ausschließlich dann, wenn sie eine erfüllende Belegung für φ gefunden hat. Also ist φ erfüllbar, d.h. $\varphi \in \text{SAT}$. $\square$