Zentralübung 7

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 17. Juli 2024
Basierend auf Folien von PD Dr. David Sabel

Plan für heute

1. Das CLIQUE-Problem
2. Das GRAPH-COLORING-Problem
3. Das KNAPSACK-Problem (nur FSK)

1. Das CLIQUE-Problem

Cliquen in Graphen

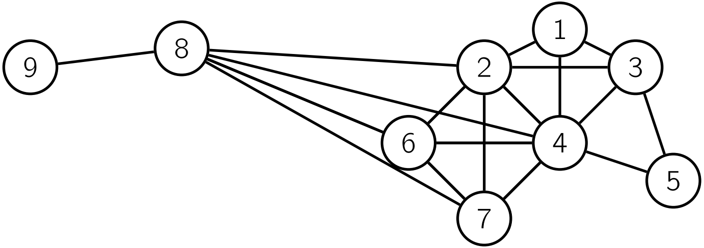
Definition

Für einen ungerichteten Graphen $G = (V, E)$ ist eine Clique der Größe k eine Menge $V' \subseteq V$, sodass $|V'| = k$ und für alle $u, v \in V'$ mit $u \neq v$ gilt: $\{u, v\} \in E$.

Quiz

Hat der Graph eine Clique der Größe

a) 1?
b) 2?
c) 3?
d) 4?
e) 5?
f) 6?
g) 7?

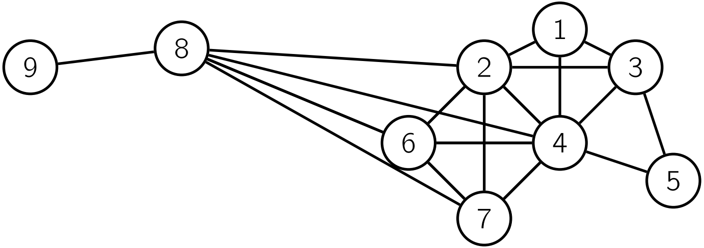


Quiz

Hat der Graph eine Clique der Größe

a) 1?
b) 2?
c) 3?
d) 4?
e) 5?
f) 6?
g) 7?

Antwort: a)–e).



Das CLIQUE-Problem

Definition

Das CLIQUE-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$

gefragt: Besitzt G eine Clique der Größe k?

Satz

CLIQUE ist $\mathcal{NP}$-vollständig.

Nachweis der $\mathcal{NP}$-Vollständigkeit

Nachweis der $\mathcal{NP}$-Vollständigkeit einer Sprache L:

1. Zugehörigkeit zu $\mathcal{NP}$:
   Gib eine Polynomialzeit-beschränkte NTM an, die L entscheidet.
2. $\mathcal{NP}$-Schwere:
   Wähle ein $\mathcal{NP}$-schweres Problem L_0 und zeige $L_0 \leq_p L$.

1. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Definition

Das FULL-CLIQUE-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$

gefragt: Besitzt G eine Clique der Größe $|V|$?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist FULL-CLIQUE $\mathcal{NP}$-vollständig?

1. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Definition

Das FULL-CLIQUE-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$

gefragt: Besitzt G eine Clique der Größe $|V|$?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist FULL-CLIQUE $\mathcal{NP}$-vollständig?

Antwort:

Nein. Man kann in deterministischer Polynomialzeit prüfen, ob der Graph vollständig ist. Daher ist das problem in $\mathcal{P}$. Unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ ist das Problem nicht $\mathcal{NP}$-vollständig.

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Definition

Das TWIN-CLIQUES-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$

gefragt: Besitzt G zwei Knotendisjunkte Cliquen der Größe k?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist TWIN-CLIQUES $\mathcal{NP}$-vollständig?

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Antwort:

Ja. Beweis:

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Antwort:

Ja. Beweis:

1. TWIN-CLIQUES $\in \mathcal{NP}$:
   NTM rät nichtdeterministisch disjunkte Knotenmengen $V_1, V_2 \subseteq V$ mit $|V_i| = k$ und prüft in deterministischer Polynomialzeit, ob V_1 und V_2 Cliquen sind.

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TWIN-CLIQUES ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TWIN-CLIQUES.

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TWIN-CLIQUES ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TWIN-CLIQUES.
   Sei $f((V, E), k) = ((V \cup V', E \cup E', k))$, wobei
   $V' = \{v' \mid v \in V\}$ und $E' = \{\{u', v'\} \mid \{u, v\} \in E\}$.

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TWIN-CLIQUES ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TWIN-CLIQUES.

   Sei $f((V, E), k) = ((V \cup V', E \cup E', k))$, wobei
   $V' = \{v' \mid v \in V\}$ und $E' = \{\{u', v'\} \mid \{u, v\} \in E\}$.

   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

2. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TWIN-CLIQUES ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TWIN-CLIQUES.

   Sei $f((V,E),k) = ((V \cup V', E \cup E', k))$, wobei
   $V' = \{v' \mid v \in V\}$ und $E' = \{\{u', v'\} \mid \{u, v\} \in E\}$.

   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

   f ist korrekt:

   $((V,E),k) \in$ CLIQUE
   g.d.w. (V,E) hat eine Clique der Größe k
   g.d.w. (V,E) und (V',E') haben jeweils eine Clique der Größe k
   g.d.w. $f((V,E),k) = ((V \cup V', E \cup E'), k) \in$ TWIN-CLIQUES

3. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Definition

Das TWO-CLIQUE-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$

gefragt: Besitzt G eine Clique der Größe 2?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist TWO-CLIQUE $\mathcal{NP}$-vollständig?

3. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Definition

Das TWO-CLIQUE-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$

gefragt: Besitzt G eine Clique der Größe 2?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist TWO-CLIQUE $\mathcal{NP}$-vollständig?

Antwort:

Nein. Man kann in deterministischer Polynomialzeit prüfen, ob $E \neq \emptyset$. Daher ist das problem in $\mathcal{P}$. Unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ ist das Problem nicht $\mathcal{NP}$-vollständig.

4. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Definition

Das TRIPLE-CLIQUE-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$ und drei Zahlen $k_1, k_2, k_3 \in \mathbb{N}$

gefragt: Besitzt G Cliquen der Größen k_1, k_2 und k_3?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist TRIPLE-CLIQUE $\mathcal{NP}$-vollständig?

4. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

Antwort:

Ja. TRIPLE-CLIQUE ist eigentlich das gleiche wie CLIQUE mit $\max\{k_1, k_2, k_3\}$.
Wir bieten trotzdem einen $\mathcal{NP}$-Vollständigkeitsbeweis an:

1. TRIPLE-CLIQUE $\in \mathcal{NP}$:
   NTM rät nichtdeterministisch Knotenmengen $V_1, V_2, V_3 \subseteq V$ mit $|V_i| = k_i$ und prüft in deterministischer Polynomialzeit, ob alle V_i Cliquen sind.

4. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TRIPLE-CLIQUE ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TRIPLE-CLIQUE.

4. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TRIPLE-CLIQUE ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TRIPLE-CLIQUE.
   Sei $f((V, E), k)) = ((V, E), k, k, k)$.

4. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TRIPLE-CLIQUE ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TRIPLE-CLIQUE.
   Sei $f((V, E), k)) = ((V, E), k, k, k)$.
   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

4. Aufgabe: $\mathcal{NP}$-vollständig oder nicht

2. TRIPLE-CLIQUE ist $\mathcal{NP}$-schwer:
   Wir zeigen CLIQUE $\leq_p$ TRIPLE-CLIQUE.

   Sei $f((V, E), k)) = ((V, E), k, k, k)$.

   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

   f ist korrekt:

   $((V, E), k) \in$ CLIQUE
   g.d.w. (V, E) hat eine Clique der Größe k
   g.d.w. (V, E) hat Cliquen der Größen k, k und k
   g.d.w. $((V, E), k, k, k) \in$ TRIPLE-CLIQUE

2. Das GRAPH-COLORING-Problem

Das GRAPH-COLORING-Problem

Definition

Das GRAPH-COLORING-Problem lässt sich wie folgt formulieren.

gegeben: ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$

gefragt: Gibt es eine Färbung der Knoten in V mit höchstens k Farben (eine sogenannte k-Färbung), sodass keine zwei benachbarten Knoten in G die gleiche Farbe erhalten?

Satz

GRAPH-COLORING ist $\mathcal{NP}$-vollständig.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

Definition

Das EXAM-ASSIGN-Problem lässt sich wie folgt formulieren.

gegeben: eine Menge P von (studierende Person, Klausur)-Paaren, die erfassen, welche Studierenden welche Klausure schreiben und eine Menge von Terminen T

gefragt: Gibt es eine Zuordnung aller Klausuren auf Termine, sodass keine studierende Person zur gleichen Zeit mehrere Klausuren schreiben muss?

Beweisen Sie, dass EXAM-ASSIGN $\mathcal{NP}$-vollständig ist.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

Antwort:

1. EXAM-ASSIGN ist in $\mathcal{NP}$:
   Rate nichtdeterministisch eine Zuordnung Klausur zu Termin und prüfe anschließend pro studierende Person, ob ein Konflikt vorliegt.
2. EXAM-ASSIGN ist $\mathcal{NP}$-schwer:
   Wir zeigen GRAPH-COLORING $\leq_p$ EXAM-ASSIGN.

Quiz

Welche der beiden folgenden Anweisungen ergibt eine korrekte Reduktionsfunktion f?

a) Pro Klausur erzeuge Knoten, pro studierende Person erzeuge Knoten.
Pro (studierende Person, Klausur)-Paar erzeuge eine Kante im Graph.
Pro Termin erzeuge eine Farbe.

b) Pro Farbe erzeuge einen Termin.
Pro Knoten v erzeuge eine Klausur v.
Pro Kante $e = \{u, v\}$ erzeuge studierende Person e, sodass e Klausur u und Klausur v schreibt, d.h. füge Paare (e, u) und (e, v) hinzu.

Quiz

Welche der beiden folgenden Anweisungen ergibt eine korrekte Reduktionsfunktion f?

a) Pro Klausur erzeuge Knoten, pro studierende Person erzeuge Knoten.
Pro (studierende Person, Klausur)-Paar erzeuge eine Kante im Graph.
Pro Termin erzeuge eine Farbe.

b) Pro Farbe erzeuge einen Termin.
Pro Knoten v erzeuge eine Klausur v.
Pro Kante $e = \{u, v\}$ erzeuge studierende Person e, sodass e Klausur u und Klausur v schreibt, d.h. füge Paare (e, u) und (e, v) hinzu.

Antwort: b).

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. EXAM-ASSIGN ist $\mathcal{NP}$-schwer:
   Wir zeigen GRAPH-COLORING $\leq_p$ EXAM-ASSIGN.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. EXAM-ASSIGN ist $\mathcal{NP}$-schwer:
   Wir zeigen GRAPH-COLORING $\leq_p$ EXAM-ASSIGN.
   $f((V, E), k) = (P, T)$, wobei
   $P = \bigcup \{\{(e, u), (e, v)\} \mid e = \{u, v\} \in E\}$ und
   $T = \{t_1, \ldots, t_k\}$, wobei die t_i paarweise disjunkt sind.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. EXAM-ASSIGN ist $\mathcal{NP}$-schwer:
   Wir zeigen GRAPH-COLORING $\leq_p$ EXAM-ASSIGN.

   $f((V,E),k) = (P,T)$, wobei
   $P = \bigcup\{\{(e,u),(e,v)\} \mid e = \{u,v\} \in E\}$ und
   $T = \{t_1, \ldots, t_k\}$, wobei die t_i paarweise disjunkt sind.

   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. EXAM-ASSIGN ist $\mathcal{NP}$-schwer:
   Wir zeigen GRAPH-COLORING $\leq_p$ EXAM-ASSIGN.

   $f((V,E),k) = (P,T)$, wobei
   $P = \bigcup\{\{(e,u),(e,v)\} \mid e = \{u,v\} \in E\}$ und
   $T = \{t_1,\ldots,t_k\}$, wobei die t_i paarweise disjunkt sind.

   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

   f ist korrekt:

   $((V,E),k) \in$ GRAPH-COLORING
   g.d.w. es gibt eine k-Färbung, sodass keine benachbarten Knoten dieselbe Farbe haben
   g.d.w. es gibt eine Zuordnung Klausur zu Termin, sodass keine studierende Person zwei Klausuren am gleichen Termin haben
   g.d.w. $f((V,E),k) = (P,T) \in$ EXAM-ASSIGN

3. Das KNAPSACK-Problem (nur FSK)

Das KNAPSACK-Problem

Definition

Das KNAPSACK-Problem lässt sich wie folgt formulieren.

gegeben: k Gegenstände mit Gewichten $w_1, \ldots, w_k \in \mathbb{N}$ und
Nutzenwerten $n_1, \ldots, n_k \in \mathbb{N}$,
sowie zwei Schwellenwerte $s_w, s_n \in \mathbb{N}$

gefragt: Gibt es eine Teilmenge $I \subseteq \{1, \ldots, k\}$, sodass $\sum_{i \in I} w_i \leq s_w$ und $\sum_{i \in I} n_i \geq s_n$?

Satz

KNAPSACK ist $\mathcal{NP}$-vollständig.

Das TASK-MACHINE-Problem

Definition

Das TASK-MACHINE-Problem lässt sich wie folgt formulieren.

gegeben: Zielwert W und n Aufgaben mit

- t_i ist die Bearbeitungszeit (in Stunden) von Aufgabe i
- d_i ist die Deadline (in Stunden) von Aufgabe i
- val_i ist der Wert von Aufgabe i.

gefragt: Es gibt eine Maschine, die immer eine Aufgabe vollständig (ohne Unterbrechung) bearbeiten kann, und dann die nächste usw. Nur wenn eine Aufgabe vor der Deadline fertigstellt wird, wird der Wert val_i erzielt, ansonsten wird für diese Aufgabe kein Wert erzielt.
Gibt es eine Abarbeitungsreihenfolge der Aufgaben, sodass der Zielwert W mindestens erreicht wird?

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

Beweisen Sie, dass TASK-MACHINE $\mathcal{NP}$-vollständig ist.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

Beweisen Sie, dass TASK-MACHINE $\mathcal{NP}$-vollständig ist.

Antwort:

1. TASK-MACHINE $\in \mathcal{NP}$:
   NTM rät nichtdeterministisch eine Abarbeitungsreihenfolge und prüft in deterministischer Polynomialzeit, ob der Wert W erreicht wird.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. TASK-MACHINE ist $\mathcal{NP}$-schwer:
   Wir zeigen KNAPSACK $\leq_p$ TASK-MACHINE.
   $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) = (W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k)$,
   wobei
   - $W = ?$
   - $t_i = ?$
   - $d_i = ?$
   - $val_i = ?$

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. TASK-MACHINE ist $\mathcal{NP}$-schwer:
   Wir zeigen KNAPSACK $\leq_p$ TASK-MACHINE.
   $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) = (W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k)$,
   wobei
   - $W = ?$
   - $t_i = w_i$
   - $d_i = ?$
   - $val_i = ?$

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. TASK-MACHINE ist $\mathcal{NP}$-schwer:
   Wir zeigen KNAPSACK $\leq_p$ TASK-MACHINE.

   $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) = (W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k)$,
   wobei

   - $W = ?$
   - $t_i = w_i$
   - $d_i = s_w$ (für alle i gleich)
   - $val_i = ?$

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. TASK-MACHINE ist $\mathcal{NP}$-schwer:
   Wir zeigen KNAPSACK $\leq_p$ TASK-MACHINE.
   $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) = (W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k)$,
   wobei
   - $W = ?$
   - $t_i = w_i$
   - $d_i = s_w$ (für alle i gleich)
   - $val_i = n_i$

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. TASK-MACHINE ist $\mathcal{NP}$-schwer:
   Wir zeigen KNAPSACK $\leq_p$ TASK-MACHINE.

   $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) = (W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k)$,
   wobei

   - $W = s_n$
   - $t_i = w_i$
   - $d_i = s_w$ (für alle i gleich)
   - $val_i = n_i$

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

2. TASK-MACHINE ist $\mathcal{NP}$-schwer:
   Wir zeigen KNAPSACK $\leq_p$ TASK-MACHINE.

   $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) = (W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k)$, wobei

   - $W = s_n$
   - $t_i = w_i$
   - $d_i = s_w$ (für alle i gleich)
   - $val_i = n_i$

   Die Funktion f ist total und in deterministischer Polynomialzeit berechenbar.

Aufgabe: $\mathcal{NP}$-Vollständigkeit beweisen

f ist korrekt:

$(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) \in \text{KNAPSACK}$

g.d.w. es gibt eine Teilmenge $I \subseteq \{1, \ldots, k\}$, sodass $\sum_{i \in I} w_i \leq s_w$ und $\sum_{i \in I} n_i \geq s_n$

g.d.w. es gibt eine Abarbeitungsreihenfolge, sodass Aufgaben $I \subseteq \{1, \ldots, k\}$ vor der Deadline s_w mit einem Gesamtwert $\sum_{i \in I} n_i \geq s_n$ abgearbeitet werden

g.d.w. $f(w_1, \ldots, w_k, n_1, \ldots, n_k, s_w, s_n) =$
$(W, t_1, \ldots, t_k, d_1, \ldots, d_k, val_1, \ldots, val_k) \in \text{TASK-MACHINE}$