Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik
Vorlesung am 26. November 2019

INSTITUT FÜR THEORETISCHE INFORMATIK

Letzte Vorlesung

- Die Klasse $\mathcal{P}$ aller Probleme, die von deterministischen Turing-Maschinen in polynomialer Zeit entschieden werden.
- Die Klasse $\mathcal{NP}$ aller Probleme, die von nichtdeterministischen Turing-Maschinen in polynomialer Zeit entschieden werden.

Ist $\mathcal{P} = \mathcal{NP}$?

- **Polynomiale Transformation** $\Pi' \propto \Pi$
  - Instanzen von Π' $\longrightarrow$ Instanzen von Π
  - in polynomialer Zeit berechenbar
  - Ja-Instanz von Π' $\longrightarrow$ Ja-Instanz von Π
    Nein-Instanz von Π' $\longrightarrow$ Nein-Instanz von Π
- Problem Π ist $\mathcal{NP}$-vollständig, wenn
  - $\Pi \in \mathcal{NP}$ und
  - $\Pi' \propto \Pi$ für alle $\Pi' \in \mathcal{NP}$

Letzte Vorlesung

- Die Klasse $\mathcal{P}$ aller Probleme, die von deterministischen Turing-Maschinen in polynomialer Zeit entschieden werden.
- Die Klasse $\mathcal{NP}$ aller Probleme, die von nichtdeterministischen Turing-Maschinen in polynomialer Zeit entschieden werden.

Ist $\mathcal{P} = \mathcal{NP}$?

- **Polynomiale Transformation** $\Pi' \propto \Pi$
  - Instanzen von Π' $\longrightarrow$ Instanzen von Π
  - in polynomialer Zeit berechenbar
  - Ja-Instanz von Π' $\longrightarrow$ Ja-Instanz von Π
Nein-Instanz von Π' $\longrightarrow$ Nein-Instanz von Π
- Problem Π ist $\mathcal{NP}$-vollständig, wenn
  - $\Pi \in \mathcal{NP}$ und
  - ~~$\Pi' \propto \Pi$ für alle $\Pi' \in \mathcal{NP}$~~
  - $\Pi' \propto \Pi$ für ein $\mathcal{NP}$-vollständiges Π'

Letzte Vorlesung

- Die Klasse $\mathcal{P}$ aller Probleme, die von deterministischen Turing-Maschinen in polynomialer Zeit entschieden werden.
- Die Klasse $\mathcal{NP}$ aller Probleme, die von nichtdeterministischen Turing-Maschinen in polynomialer Zeit entschieden werden.

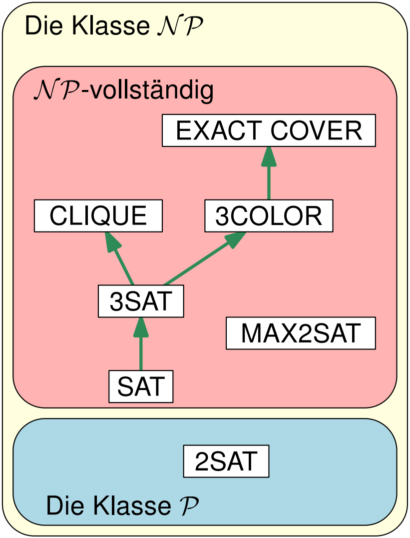
Ist $\mathcal{P} = \mathcal{NP}$?

- **Polynomiale Transformation** $\Pi' \propto \Pi$
  - Instanzen von $\Pi' \longrightarrow$ Instanzen von Π
  - in polynomialer Zeit berechenbar
  - Ja-Instanz von $\Pi' \longrightarrow$ Ja-Instanz von Π
    Nein-Instanz von $\Pi' \longrightarrow$ Nein-Instanz von Π
- Problem Π ist $\mathcal{NP}$-vollständig, wenn
  - $\Pi \in \mathcal{NP}$ und
  - ~~$\Pi' \propto \Pi$ für alle $\Pi' \in \mathcal{NP}$~~
  - $\Pi' \propto \Pi$ für ein $\mathcal{NP}$-vollständiges Π'

Satz von Cook. SAT ist $\mathcal{NP}$-vollständig.

Plan für heute

- 3SAT ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3SAT $\in \mathcal{NP}$
  - $\rightsquigarrow$ SAT $\propto$ 3SAT
- 2SAT ist in $\mathcal{P}$
- MAX2SAT ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ Übung
- CLIQUE ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ CLIQUE $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3SAT $\propto$ CLIQUE
- 3COLOR ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3COLOR $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3SAT $\propto$ 3COLOR
- EXACT COVER ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3COLOR $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER



Das Problem 3SAT

Problem 3SAT

Gegeben: Menge U von Variablen
Menge C von Klauseln über U
jede Klausel enthält genau drei Literale

Frage: Existiert eine erfüllende Wahrheitsbelegung für C?

Das Problem 3SAT

Problem 3SAT

Gegeben: Menge U von Variablen
Menge C von Klauseln über U
jede Klausel enthält genau drei Literale

Frage: Existiert eine erfüllende Wahrheitsbelegung für C?

Satz:
Das Problem 3SAT ist $\mathcal{NP}$-vollständig.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

3SAT $\in \mathcal{NP}$:

- Es existiert eine nichtdeterministische Turing-Maschine mit polynomialer Zeitkomplexitätsfunktion, die (immer hält und) in q_J hält bei Ja-Instanzen und in q_N hält bei Nein-Instanzen.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

3SAT $\in \mathcal{NP}$:

- Es existiert eine nichtdeterministische Turing-Maschine mit polynomialer Zeitkomplexitätsfunktion, die (immer hält und) in q_J hält bei Ja-Instanzen und in q_N hält bei Nein-Instanzen.
- Wir konstruieren eine Orakel-Turing-Maschine:
  - Das Orakelmodul schreibt eine Wahrheitsbelegung $t \colon U \to \{\texttt{wahr}, \texttt{falsch}\}$.
  - Die endliche Kontrolle überprüft, ob jede Klausel in C durch t erfüllt ist.
  - Wenn alle Klauseln erfüllt, gehe in q_J.
  - Wenn (mindestens) eine Klausel nicht erfüllt, gehe in q_N.
  - Laufzeit ist linear in der Größe der eingegebenen Klauselmenge C.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

3SAT $\in \mathcal{NP}$:

- Es existiert eine nichtdeterministische Turing-Maschine mit polynomialer Zeitkomplexitätsfunktion, die (immer hält und) in q_J hält bei Ja-Instanzen und in q_N hält bei Nein-Instanzen.
- Wir konstruieren eine Orakel-Turing-Maschine:
  - Das Orakelmodul schreibt eine Wahrheitsbelegung $t\colon U \to \{\texttt{wahr}, \texttt{falsch}\}$.
  - Die endliche Kontrolle überprüft, ob jede Klausel in C durch t erfüllt ist.
  - Wenn alle Klauseln erfüllt, gehe in q_J.
  - Wenn (mindestens) eine Klausel nicht erfüllt, gehe in q_N.
  - Laufzeit ist linear in der Größe der eingegebenen Klauselmenge C.
- Für eine feste Wahrheitsbelegung t kann in polynomialer Zeit $\mathcal{O}(|C|)$ überprüft werden, ob t alle Klauseln aus C erfüllt.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

SAT $\propto$ 3SAT:

- Wir geben eine polynomiale Transformation f von SAT zu 3SAT an.
- Gegeben sei eine SAT-Instanz I.

Wir konstruieren eine 3SAT-Instanz $f(I)$, indem wir jede Klausel c in I einzeln auf Klausel(n) $f(c)$ in $f(I)$ abbilden:

- Besteht die Klausel $c = x_1$ aus **einem** Literal, so wird c auf $x_1 \vee x_1 \vee x_1$ abgebildet.
- Besteht die Klausel $c = x_1 \vee x_2$ aus **zwei** Literalen, so wird c auf $x_1 \vee x_2 \vee x_1$ abgebildet.
- Besteht die Klausel c aus **drei** Literalen, so wird c auf sich selbst abgebildet.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

Wir konstruieren eine 3SAT-Instanz $f(I)$, indem wir jede Klausel c in I einzeln auf Klausel(n) $f(c)$ in $f(I)$ abbilden:

- Besteht die Klausel $c = x_1 \vee \cdots \vee x_k$ aus $k > 3$ Literalen, bilde c wie folgt ab:
  - Führe $k - 3$ neue Variablen $y_{c,3}, \ldots, y_{c,k-1}$ ein.
  - Bilde c auf die folgenden $k - 2$ Klauseln ab:

$$
\begin{array}{rcl}
x_1 \vee x_2 \vee y_{c,3} & & \text{„Falls } x_1, x_2 = \texttt{falsch} \rightsquigarrow y_{c,3} \text{ muss } \texttt{wahr}\text{“} \\
\overline{y_{c,3}} \vee x_3 \vee y_{c,4} & & \text{„}y_{c,3} = \texttt{wahr}, x_3 = \texttt{falsch} \rightsquigarrow y_{c,4} = \texttt{wahr}\text{“} \\
\vdots & & \vdots \\
\overline{y_{c,k-2}} \vee x_{k-2} \vee y_{c,k-1} & & \text{„}y_{c,k-2} = \texttt{wahr}, x_{k-2} = \texttt{falsch} \rightsquigarrow y_{c,k-1} = \texttt{wahr}\text{“} \\
\overline{y_{c,k-1}} \vee x_{k-1} \vee x_k & & \text{„}y_{c,k-1} = \texttt{wahr} \rightsquigarrow x_{k-1} \text{ oder } x_k \text{ muss } \texttt{wahr}\text{“}
\end{array}
$$

- Diese Klauseln lassen sich in Zeit $\mathcal{O}(|C| \cdot |U|)$ konstruieren.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

Noch zu zeigen:

- I ist erfüllbar $\Leftrightarrow$ $f(I)$ ist erfüllbar

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

I ist erfüllbar $\Rightarrow$ $f(I)$ ist erfüllbar

- Sei die SAT-Instanz I erfüllbar.
- Wir setzen eine erfüllende Wahrheitsbelegung t von I auf $f(I)$ fort.
  - Also wenn es Literal x in I und $f(I)$ gibt, lasse $t(x)$ wie gehabt.
- Wir untersuchen jede Klausel $c = x_1 \vee \cdots \vee x_k$ in I einzeln.
- Da c von t erfüllt ist, gilt für mindestens ein i, dass $t(x_i) = \texttt{wahr}$.
- Fall $k \leq 3$: Damit ist auch Klausel c in $f(I)$ erfüllt.
- Fall $k > 3$: Setze für $j = 3, \ldots, k-1$

$$t(y_{c,j}) = \begin{cases} \texttt{wahr} & \text{falls } t(x_i) = \texttt{wahr} \text{ für (mind.) ein } i \geq j \\ \texttt{falsch} & \text{falls } t(x_i) = \texttt{falsch} \text{ für alle } i \geq j \end{cases}$$

- Diese Erweiterung erfüllt alle Klauseln in $f(I)$, die zu c gehören.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3SAT

I ist erfüllbar $\Leftarrow f(I)$ ist erfüllbar

- Sei die 3SAT-Instanz $f(I)$ erfüllbar.
- Wir schränken eine erfüllende Wahrheitsbelegung t von $f(I)$ auf I ein.
  - Also nehme für jedes Literal x in I die Belegung $t(x)$ wie in $f(I)$.
- Wir untersuchen jede Klausel $c = x_1 \vee \cdots \vee x_k$ in I einzeln.
- Fall $k \leq 3$: Dann ist Klausel c auch in $f(I)$, also durch t erfüllt.
- Fall $k > 3$: Alle Klauseln in $f(I)$ zu Klausel c in I sind erfüllt:

$$x_1 \vee x_2 \vee y_{c,3}$$

Falls $t(x_1), t(x_2) = \texttt{falsch}$, dann $t(y_{c,3}) = \texttt{wahr}$.

$$\overline{y_{c,j}} \vee x_j \vee y_{c,j+1}$$

Falls $t(y_{c,j}) = \texttt{wahr}$, $t(x_j) = \texttt{falsch}$, dann $t(y_{c,j+1}) = \texttt{wahr}$.

$$\overline{y_{c,k-1}} \vee x_{k-1} \vee x_k$$

Falls $t(y_{c,k-1}) = \texttt{wahr}$, dann $t(x_{k-1})$ oder $t(x_k) = \texttt{wahr}$.

- Also ist $t(x_i) = \texttt{wahr}$ für (mind.) ein i, und demnach c erfüllt durch t.

Das Problem 2SAT

Problem 2SAT

Gegeben: Menge U von Variablen
Menge C von Klauseln über U
wobei jede Klausel genau zwei Literale enthält

Frage: Existiert eine erfüllende Wahrheitsbelegung für C?

Satz:
Das Problem 2SAT liegt in $\mathcal{P}$.

Beweis: Übung

Das Problem MAX2SAT

Problem MAX2SAT

Gegeben: Menge U von Variablen
Menge C von Klauseln über U
wobei jede Klausel genau zwei Literale enthält
Zahl $K \in \mathbb{N}$

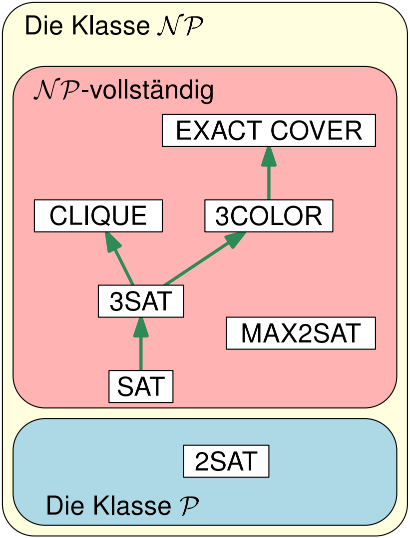
Frage: Existiert eine Wahrheitsbelegung, die mindestens K Klauseln erfüllt?

Satz:
Das Problem MAX2SAT ist $\mathcal{NP}$-vollständig.

Beweis: Übung

Plan für heute

- 3SAT ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3SAT $\in \mathcal{NP}$
  - $\rightsquigarrow$ SAT $\propto$ 3SAT
- 2SAT ist in $\mathcal{P}$
- MAX2SAT ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ Übung
- CLIQUE ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ CLIQUE $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3SAT $\propto$ CLIQUE
- 3COLOR ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3COLOR $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3SAT $\propto$ 3COLOR
- EXACT COVER ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3COLOR $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER



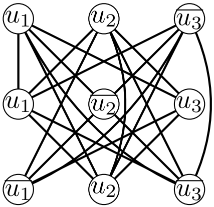
Das Problem CLIQUE

Eine **Clique** in einem Graphen $G = (V, E)$ ist eine Menge $V' \subseteq V$, sodass für alle $i, j \in V'$, $i \neq j$, gilt: $\{i, j\} \in E$.

Problem CLIQUE

Gegeben: Graph $G = (V, E)$ und ein Parameter $K \leq |V|$

Frage: Gibt es in G eine Clique der Größe mindestens K?



Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

Satz:
Das Problem CLIQUE ist $\mathcal{NP}$-vollständig.

CLIQUE $\in \mathcal{NP}$:

- Für eine gegebene Menge $V' \subseteq V$ kann in polynomieller Zeit überprüft werden, ob
  - für alle $i, j \in V'$, $i \neq j$ gilt: $\{i, j\} \in E$,
  - $|V'| \geq K$.

Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

3SAT $\propto$ CLIQUE

- Sei $C = \{c_1, \ldots, c_n\}$ eine 3SAT-Instanz mit $c_i = x_{i1} \vee x_{i2} \vee x_{i3}$ und $x_{ij} \in \{u_1, \ldots, u_m, \overline{u_1}, \ldots, \overline{u_m}\}$.

Wir transformieren C in eine CLIQUE-Instanz $(G = (V, E), K)$.

- V enthält $3n$ Knoten v_{ij} für $1 \leq i \leq n$, $1 \leq j \leq 3$.
- v_{ij} und $v_{k\ell}$ sind genau dann durch Kanten aus E verbunden, wenn:
  - $i \neq k$ (Literale sind in verschiedenen Klauseln)
  - $x_{ij} \neq \overline{x_{k\ell}}$ (Literale sind gleichzeitig erfüllbar)
- Wir setzen $K := n$.

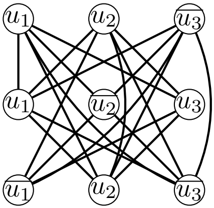
Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

Karlsruher Institut für Technologie

- V enthält $3n$ Knoten v_{ij} für $1 \leq i \leq n$, $1 \leq j \leq 3$.
- v_{ij} und $v_{k\ell}$ sind genau dann durch Kanten aus E verbunden, wenn:
  - $i \neq k$ (Literale sind in verschiedenen Klauseln)
  - $x_{ij} \neq \overline{x_{k\ell}}$ (Literale sind gleichzeitig erfüllbar)

Beispiel: Sei $C = \{u_1 \vee u_2 \vee \overline{u_3},\ u_1 \vee \overline{u_2} \vee u_3,\ \overline{u_1} \vee u_2 \vee \overline{u_3}\}$.

Knotennummer	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	v_{31}	v_{32}	v_{33}
Literal	u_1	u_2	$\overline{u_3}$	u_1	$\overline{u_2}$	u_3	$\overline{u_1}$	u_2	$\overline{u_3}$



Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

- Die Transformation kann in polynomieller Zeit berechnet werden.

Noch zu zeigen:

- 3SAT-Instanz C ist Ja-Instanz $\Leftrightarrow$ CLIQUE-Instanz (G, K) ist Ja-Instanz
  - C ist Ja-Instanz:
    Es existiert Wahrheitsbelegung $t\colon U \to \{\texttt{wahr}, \texttt{falsch}\}$, sodass alle Klauseln in C unter t erfüllt sind.
  - (G, K) ist Ja-Instanz:
    Es existiert Knotenmenge $V' \subseteq V$, sodass $|V'| \geq K$ und alle Knoten in V' paarweise miteinander durch Kanten in G verbunden sind.

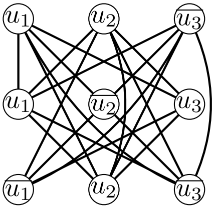
Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

3SAT-Instanz C ist Ja-Instanz $\Rightarrow$ CLIQUE-Instanz (G, K) ist Ja-Instanz:

- Wähle eine beliebige erfüllende Wahrheitsbelegung t von C.
- Wähle in jeder Klausel ein wahres Literal.
- Diese Knoten in G bilden eine Clique V' der Größe $K = n$.

Beispiel: Sei $C = \{u_1 \vee u_2 \vee \overline{u_3},\ u_1 \vee \overline{u_2} \vee u_3,\ \overline{u_1} \vee u_2 \vee \overline{u_3}\}$.

Knotennummer	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	v_{31}	v_{32}	v_{33}
Literal	u_1	u_2	$\overline{u_3}$	u_1	$\overline{u_2}$	u_3	$\overline{u_1}$	u_2	$\overline{u_3}$.



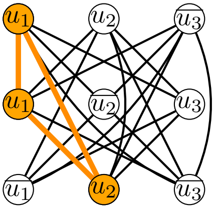
Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

3SAT-Instanz C ist Ja-Instanz $\Rightarrow$ CLIQUE-Instanz (G, K) ist Ja-Instanz:

- Wähle eine beliebige erfüllende Wahrheitsbelegung t von C.
- Wähle in jeder Klausel ein wahres Literal.
- Diese Knoten in G bilden eine Clique V' der Größe $K = n$.

Beispiel: Sei $C = \{u_1 \vee u_2 \vee \overline{u_3},\ u_1 \vee \overline{u_2} \vee u_3,\ \overline{u_1} \vee u_2 \vee \overline{u_3}\}$.

Knotennummer	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	v_{31}	v_{32}	v_{33}
Literal	u_1	u_2	$\overline{u_3}$	u_1	$\overline{u_2}$	u_3	$\overline{u_1}$	u_2	$\overline{u_3}$.



$t(u_1) = \texttt{wahr}$

$t(u_2) = \texttt{wahr}$

$t(u_3) = \texttt{falsch}$

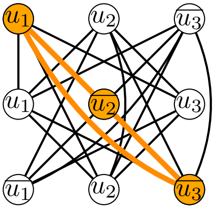
Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

3SAT-Instanz C ist Ja-Instanz $\Rightarrow$ CLIQUE-Instanz (G, K) ist Ja-Instanz:

- Wähle eine beliebige erfüllende Wahrheitsbelegung t von C.
- Wähle in jeder Klausel ein wahres Literal.
- Diese Knoten in G bilden eine Clique V' der Größe $K = n$.

Beispiel: Sei $C = \{u_1 \vee u_2 \vee \overline{u_3},\ u_1 \vee \overline{u_2} \vee u_3,\ \overline{u_1} \vee u_2 \vee \overline{u_3}\}$.

Knotennummer	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	v_{31}	v_{32}	v_{33}
Literal	u_1	u_2	$\overline{u_3}$	u_1	$\overline{u_2}$	u_3	$\overline{u_1}$	u_2	$\overline{u_3}$.



$t(u_1) = \texttt{wahr}$

$t(u_2) = \texttt{falsch}$

$t(u_3) = \texttt{falsch}$

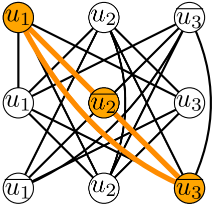
Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

3SAT-Instanz C ist Ja-Instanz $\Leftarrow$ CLIQUE-Instanz (G, K) ist Ja-Instanz:

- Wähle eine Clique V' der Größe $K = n$ in G.
- Dies ist ein Knoten pro Klausel. Wir setzen dieses Literal in t auf `wahr`.
- Dann ist $t(u) \neq t(\overline{u})$ und alle Klauseln in C sind erfüllt.

Beispiel: Sei $C = \{u_1 \vee u_2 \vee \overline{u_3},\ u_1 \vee \overline{u_2} \vee u_3,\ \overline{u_1} \vee u_2 \vee \overline{u_3}\}$.

Knotennummer	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	v_{31}	v_{32}	v_{33}
Literal	u_1	u_2	$\overline{u_3}$	u_1	$\overline{u_2}$	u_3	$\overline{u_1}$	u_2	$\overline{u_3}$.



$t(u_1) = \texttt{wahr}$

$t(u_2) = \texttt{falsch}$

$t(u_3) = \texttt{falsch}$

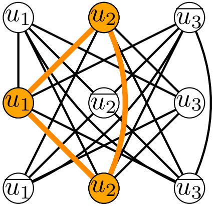
Beweis: $\mathcal{NP}$-Vollständigkeit von CLIQUE

3SAT-Instanz C ist Ja-Instanz $\Leftarrow$ CLIQUE-Instanz (G, K) ist Ja-Instanz:

- Wähle eine Clique V' der Größe $K = n$ in G.
- Dies ist ein Knoten pro Klausel. Wir setzen dieses Literal in t auf `wahr`.
- Dann ist $t(u) \neq t(\overline{u})$ und alle Klauseln in C sind erfüllt.

Beispiel: Sei $C = \{u_1 \vee u_2 \vee \overline{u_3},\ u_1 \vee \overline{u_2} \vee u_3,\ \overline{u_1} \vee u_2 \vee \overline{u_3}\}$.

Knotennummer	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	v_{31}	v_{32}	v_{33}
Literal	u_1	u_2	$\overline{u_3}$	u_1	$\overline{u_2}$	u_3	$\overline{u_1}$	u_2	$\overline{u_3}$.



$t(u_1) = \texttt{wahr}$

$t(u_2) = \texttt{wahr}$

$t(u_3) = $ beliebig

Das Problem COLOR

Problem COLOR

Gegeben: Graph $G = (V, E)$ und ein Parameter $K \in \mathbb{N}$.

Frage: Gibt es eine Knotenfärbung von G mit höchstens K Farben, sodass je zwei adjazente Knoten verschiedene Farben besitzen?

3COLOR bezeichnet das Problem COLOR mit festem Parameter $K = 3$.

Satz:
Das Problem 3COLOR ist $\mathcal{NP}$-vollständig.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3COLOR

3COLOR $\in \mathcal{NP}$

- Es kann in Zeit $\mathcal{O}(|E|)$ überprüft werden, ob eine Färbung von Graph $G = (V, E)$ mit drei Farben zulässig ist.

Beweis: $\mathcal{NP}$-Vollständigkeit von 3COLOR

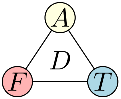
3SAT $\propto$ 3COLOR

- Sei I eine 3SAT-Instanz mit Variablen $U = \{u_1, \ldots, u_m\}$ und Klauseln $\{c_1, \ldots, c_n\}$.
- Wir konstruieren in Polynomialzeit eine 3COLOR-Instanz G.
- Es soll gelten: I ist erfüllbar $\Leftrightarrow$ G ist 3-färbbar.

Konstruktion von 3COLOR-Instanz G

Der Graph G enthält:

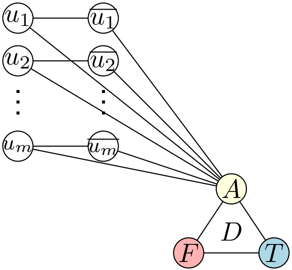
- Hauptdreieck D aus Knoten $\{T, F, A\}$ und Kanten $\{\{T, F\}, \{F, A\}, \{T, A\}\}$
- Interpretation: T, F, A sind die drei Farben, mit denen G gefärbt wird.
- Interpretation: $T \longleftrightarrow \texttt{wahr}$, $F \longleftrightarrow \texttt{falsch}$



Konstruktion von 3COLOR-Instanz G

Der Graph G enthält:

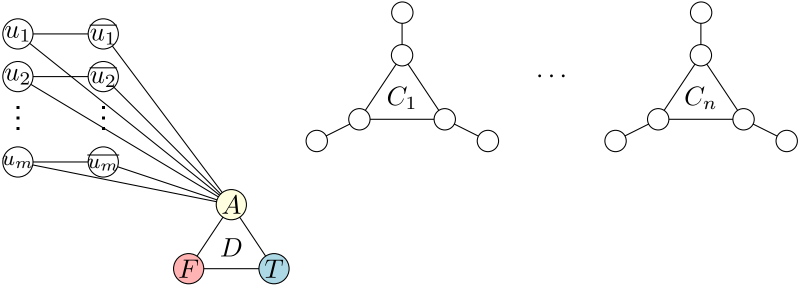
- Hauptdreieck D aus Knoten $\{T, F, A\}$ und Kanten $\{\{T, F\}, \{F, A\}, \{T, A\}\}$
- Interpretation: T, F, A sind die drei Farben, mit denen G gefärbt wird.
- Interpretation: $T \longleftrightarrow \texttt{wahr}$, $F \longleftrightarrow \texttt{falsch}$
- Für jede Variable $u_i \in U$ zwei Knoten $u_i, \overline{u_i}$ und ein Dreieck $\{u_i, \overline{u_i}, A\}$.
- Interpretation: Falls u_i in Farbe T, muss $\overline{u_i}$ in Farbe F.



Konstruktion von 3COLOR-Instanz G

Der Graph G enthält:

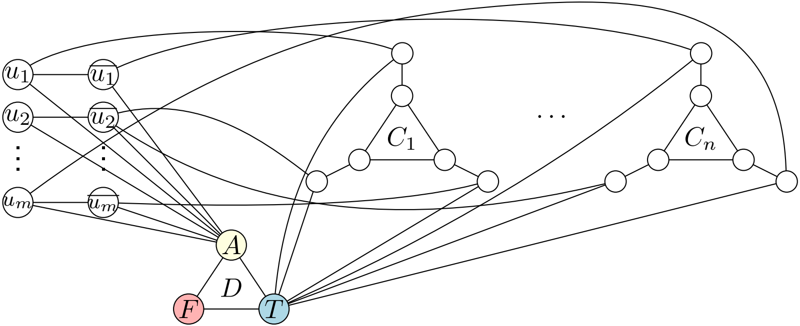
- Für jede Klausel $c_j = x \vee y \vee z$ eine Komponente C_j wie folgt:
  - C_j besteht aus sechs Knoten, einem „inneren Dreieck“ und drei „Satelliten“.



Konstruktion von 3COLOR-Instanz G

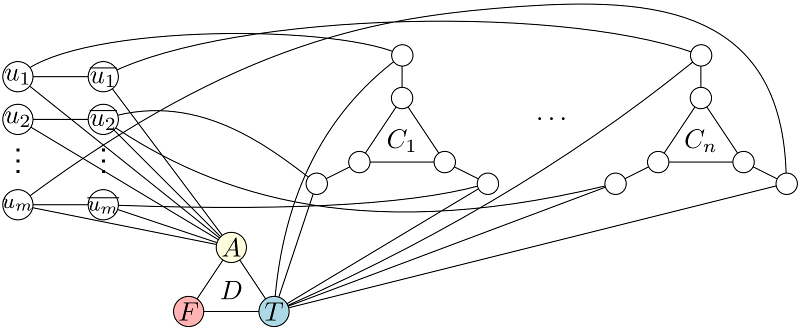
Der Graph G enthält:

- Für jede Klausel $c_j = x \vee y \vee z$ eine Komponente C_j wie folgt:
  - C_j besteht aus sechs Knoten, einem „inneren Dreieck“ und drei „Satelliten“.
- Jeder der drei Satelliten wird mit einem der Literale x, y, z verbunden.
- Alle drei Satelliten werden mit dem Knoten T in D verbunden.



Polynomialität der Reduktion

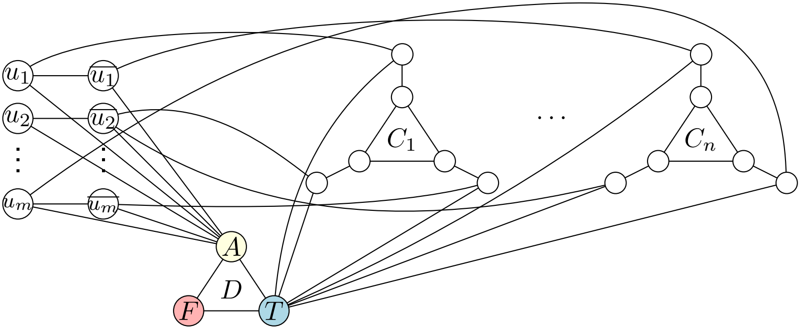
- Die Knotenanzahl von G liegt in $\mathcal{O}(n+m)$.
- Deswegen ist die Transformation polynomial.



Zu zeigen:
3SAT-Instanz I ist Ja-Instanz $\Leftrightarrow$ 3COLOR-Instanz G ist Ja-Instanz.

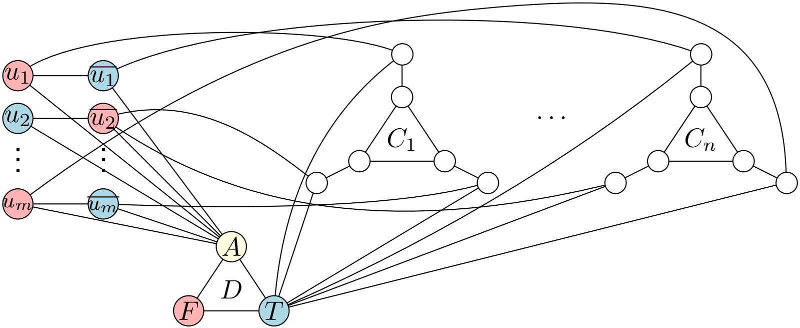
Instanz I erfüllbar $\Rightarrow$ Instanz G 3-färbbar

- Betrachte zulässige Wahrheitsbelegung t für I.
- Färbe wahre Literale wie Knoten T, falsche Literale wie Knoten F.



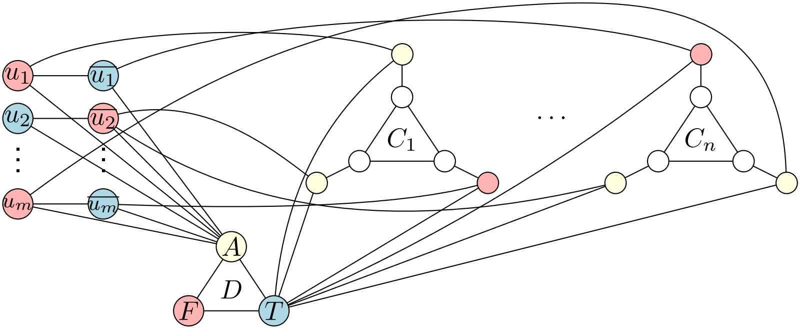
Instanz I erfüllbar $\Rightarrow$ Instanz G 3-färbbar

- Betrachte zulässige Wahrheitsbelegung t für I.
- Färbe wahre Literale wie Knoten T, falsche Literale wie Knoten F.
- Färbe Satelliten zu einem beliebigen wahren Literal mit F.



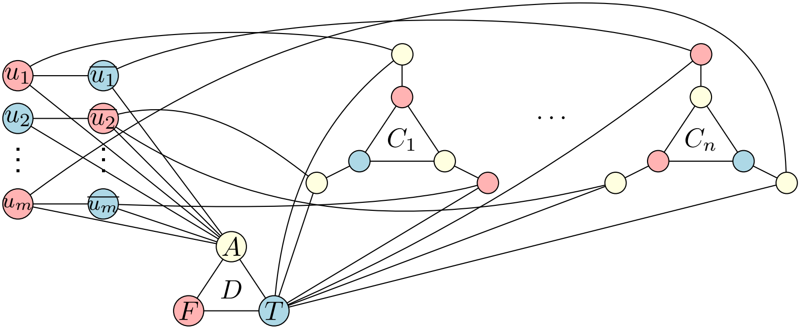
Instanz I erfüllbar $\Rightarrow$ Instanz G 3-färbbar

- Betrachte zulässige Wahrheitsbelegung t für I.
- Färbe wahre Literale wie Knoten T, falsche Literale wie Knoten F.
- Färbe Satelliten zu einem beliebigen wahren Literal mit F.
- Färbe die beiden anderen Satelliten mit A.



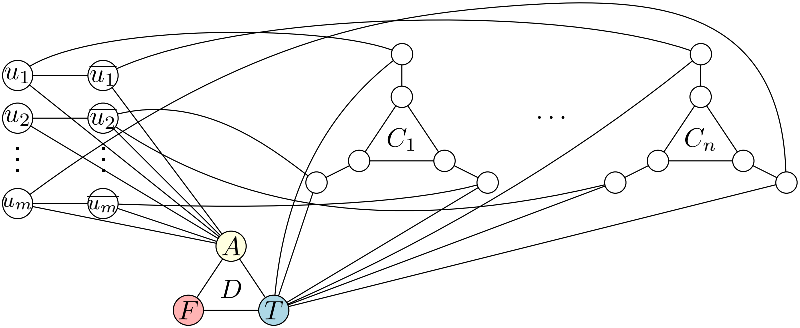
Instanz I erfüllbar $\Rightarrow$ Instanz G 3-färbbar

- Betrachte zulässige Wahrheitsbelegung t für I.
- Färbe wahre Literale wie Knoten T, falsche Literale wie Knoten F.
- Färbe Satelliten zu einem beliebigen wahren Literal mit F.
- Färbe die beiden anderen Satelliten mit A.
- Inneres Dreieck kann dann zulässig gefärbt werden.



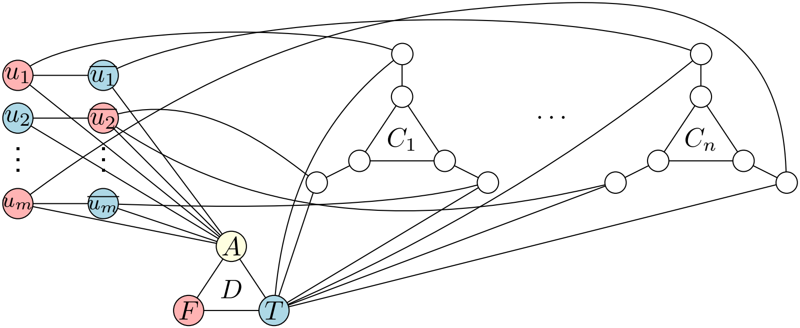
Instanz I erfüllbar $\Leftarrow$ Instanz G 3-färbbar

- Betrachte 3-Färbung von G. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.



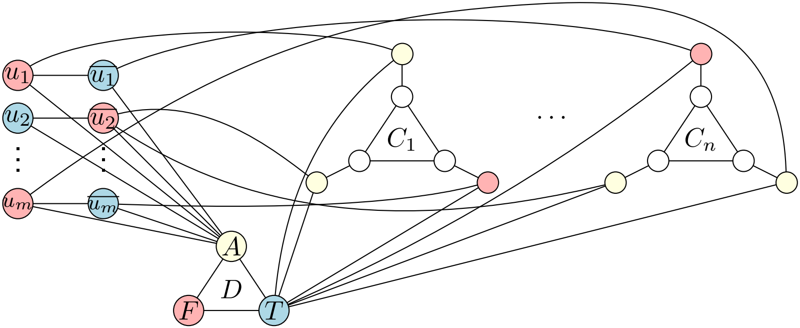
Instanz I erfüllbar $\Leftarrow$ Instanz G 3-färbbar

- Betrachte 3-Färbung von G. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.
- Färbung von Literal-Knoten induziert eine Wahrheitsbelegung t von I.



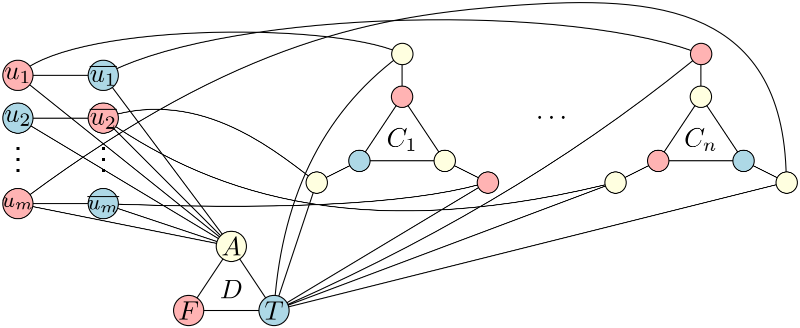
Instanz I erfüllbar $\Leftarrow$ Instanz G 3-färbbar

- Betrachte 3-Färbung von G. Interpretiere $T \longleftrightarrow \texttt{wahr}$, $F \longleftrightarrow \texttt{falsch}$.
- Färbung von Literal-Knoten induziert eine Wahrheitsbelegung t von I.
- Kein Satellit ist in Farbe von T.



Instanz I erfüllbar $\Leftarrow$ Instanz G 3-färbbar

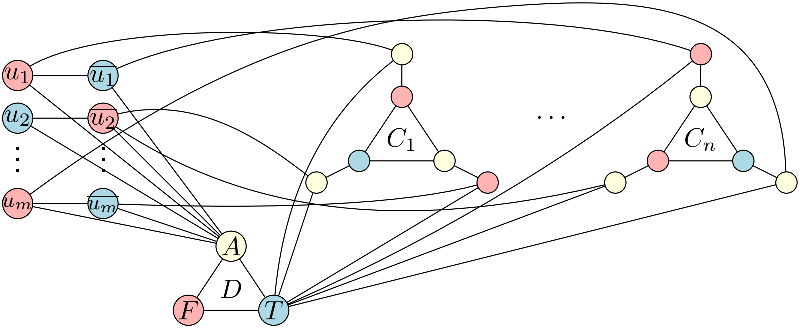
- Betrachte 3-Färbung von G. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.
- Färbung von Literal-Knoten induziert eine Wahrheitsbelegung t von I.
- Kein Satellit ist in Farbe von T.
- Nicht alle Satelliten sind in Farbe von A wegen des inneren Dreiecks.



Instanz I erfüllbar $\Leftarrow$ Instanz G 3-färbbar

- Betrachte 3-Färbung von G. Interpretiere $T \longleftrightarrow \texttt{wahr}$, $F \longleftrightarrow \texttt{falsch}$.
- Färbung von Literal-Knoten induziert eine Wahrheitsbelegung t von I.
- Kein Satellit ist in Farbe von T.
- Nicht alle Satelliten sind in Farbe von A wegen des inneren Dreiecks.

  $\Rightarrow$ Mindestens ein Literal pro Klausel in Farbe T, also I erfüllbar.



Zwischenstand Polynomiale Reduktion

Π'	$\propto$	Π
bekannt $\mathcal{NP}$-vollständig (wähle ähnlich zu Π)		$\mathcal{NP}$-vollständig zu zeigen
beliebige Instanzen	$\xrightarrow{f}$	spezielle Instanzen (größer, aber noch polynomial)
Ja-Instanzen	$\longleftrightarrow$	Ja-Instanzen

Zwischenstand Polynomiale Reduktion

Π'	$\propto$	Π
bekannt $\mathcal{NP}$-vollständig (wähle ähnlich zu Π)		$\mathcal{NP}$-vollständig zu zeigen
beliebige Instanzen	$\xrightarrow{f}$	spezielle Instanzen (größer, aber noch polynomial)
Ja-Instanzen	$\longleftrightarrow$	Ja-Instanzen

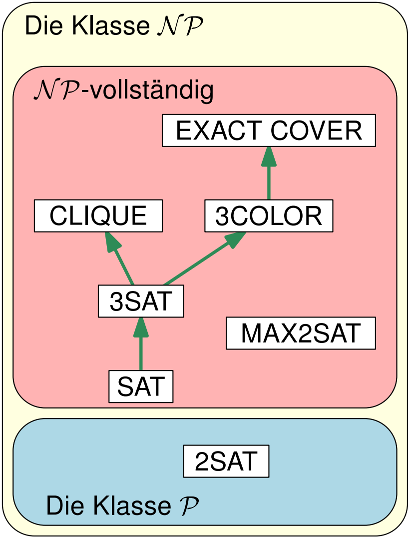
Testen Sie sich:

Können Sie zeigen, dass folgende Probleme $\mathcal{NP}$-vollständig sind?

Geg.: Graph G, Zahl K **Frage:** $\exists V' \subseteq V, \lvert V' \rvert = k$, keine zwei Knoten in V' verbunden?	**Geg.:** Graph $G = (V, E)$ **Frage:** $\exists$ 4-Färbung von V, keine zwei Knoten gleichfarbig verbunden?

Plan für heute

- 3SAT ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3SAT $\in \mathcal{NP}$
  - $\rightsquigarrow$ SAT $\propto$ 3SAT
- 2SAT ist in $\mathcal{P}$
- MAX2SAT ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ Übung
- CLIQUE ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ CLIQUE $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3SAT $\propto$ CLIQUE
- 3COLOR ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3COLOR $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3SAT $\propto$ 3COLOR
- EXACT COVER ist $\mathcal{NP}$-vollständig
  - $\rightsquigarrow$ 3COLOR $\in \mathcal{NP}$
  - $\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER



Das Problem EXACT COVER

Problem EXACT COVER

Gegeben: Eine endliche Menge X und eine Familie $\mathcal{S}$ von Teilmengen von X.

Frage: Existiert eine Menge $\mathcal{S}' \subseteq \mathcal{S}$, sodass jedes Element aus X in genau einer Menge aus $\mathcal{S}'$ liegt?

Das Problem EXACT COVER

Problem EXACT COVER

Gegeben: Eine endliche Menge X und eine Familie $\mathcal{S}$ von Teilmengen von X.

Frage: Existiert eine Menge $\mathcal{S}' \subseteq \mathcal{S}$, sodass jedes Element aus X in genau einer Menge aus $\mathcal{S}'$ liegt?

Beispiel:

$$\begin{aligned} X &= \{1, 2, \ldots, 7\} \\ \mathcal{S} &= \{\{1,2,3\}, \{1,2,4\}, \{2,3,4\}, \{1,3,4\}, \{1,5\}, \{3,5\}, \{1,3\}, \\ &\quad \{5,6,7\}, \{4,5,6\}, \{4,5,7\}, \{4,6,7\}, \{5,6\}, \{5,7\}, \{6,7\}\} \end{aligned}$$

Ist $(X, \mathcal{S})$ eine Ja-Instanz?

Das Problem EXACT COVER

Problem EXACT COVER

Gegeben: Eine endliche Menge X und eine Familie $\mathcal{S}$ von Teilmengen von X.

Frage: Existiert eine Menge $\mathcal{S}' \subseteq \mathcal{S}$, sodass jedes Element aus X in genau einer Menge aus $\mathcal{S}'$ liegt?

Beispiel:

$$\begin{aligned} X &= \{1,2,\ldots,7\} \\ \mathcal{S} &= \{\{1,2,3\},\{1,2,4\},\{2,3,4\},\{1,3,4\},\{1,5\},\{3,5\},\{1,3\}, \\ &\quad \{5,6,7\},\{4,5,6\},\{4,5,7\},\{4,6,7\},\{5,6\},\{5,7\},\{6,7\}\} \\ \mathcal{S}' &= \{\{1,5\},\{2,3,4\},\{6,7\}\} \end{aligned}$$

Ja.

Das Problem EXACT COVER

Problem EXACT COVER

Gegeben: Eine endliche Menge X und eine Familie $\mathcal{S}$ von Teilmengen von X.

Frage: Existiert eine Menge $\mathcal{S}' \subseteq \mathcal{S}$, sodass jedes Element aus X in genau einer Menge aus $\mathcal{S}'$ liegt?

Satz:
Problem EXACT COVER ist $\mathcal{NP}$-vollständig.

Beweis: $\mathcal{NP}$-Vollständigkeit von EXACT COVER

EXACT COVER $\in \mathcal{NP}$

- Es kann in Polynomialzeit überprüft werden, ob eine Teilmenge $\mathcal{S}' \subseteq \mathcal{S}$ aus disjunkten Mengen besteht und X überdeckt.

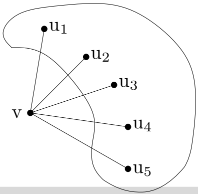
Beweis: $\mathcal{NP}$-Vollständigkeit von EXACT COVER

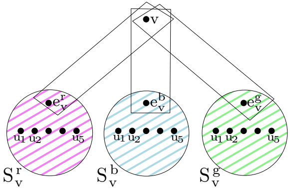
3COLOR $\propto$ EXACT COVER

- Sei $G = (V, E)$ eine 3COLOR-Instanz.
- Wir konstruieren in Polynomialzeit eine EXACT COVER-Instanz $(X, \mathcal{S})$.
- Es soll gelten: G ist 3-färbbar $\Leftrightarrow$ $(X, \mathcal{S})$ hat exakte Überdeckung

Konstruktion von (X, S)

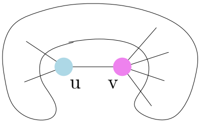
- Sei $C = \{r, b, g\}$ (steht für rot, blau, grün).
- Sei $N(v) := \{u \in V : \{u, v\} \in E\}$ die Nachbarschaft von v.
- Für jedes $v \in V$ enthalte X ein „Element“ v und jeweils $3 \cdot |N(v)| + 3$ zusätzliche Elemente.
- Zu jedem $v \in V$ gebe es in $\mathcal{S}$ drei disjunkte Mengen S_v^r, S_v^b, S_v^g mit jeweils $|N(v)| + 1$ Elementen.
- Außerdem enthalte $\mathcal{S}$ für jedes v drei zweielementige Mengen $\{v, e_v^r\}, \{v, e_v^b\}$ und $\{v, e_v^g\}$ mit $e_v^r \in S_v^r$, $e_v^b \in S_v^b$ und $e_v^g \in S_v^g$.
- **Interpretation:** S_v^r entspricht der „Farbe“ r, enthält für jeden Knoten aus $N(v)$ eine Kopie und einen zusätzlichen Knoten e_v^r.

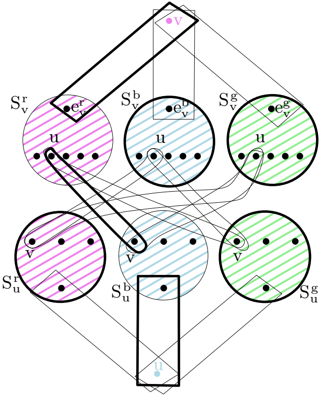




Konstruktion von (X, S)

- Außerdem enthält $\mathcal{S}$ für jede Kante $\{u, v\} \in E$ und je zwei $c, c' \in C$, $c \neq c'$, die zweielementigen Mengen $\{u_v^c, v_u^{c'}\}$, $u_v^c \in S_v^c$ „Kopie“ von u, $v_u^{c'} \in S_u^{c'}$ „Kopie“ von v.





Konstruktion von (X, S)

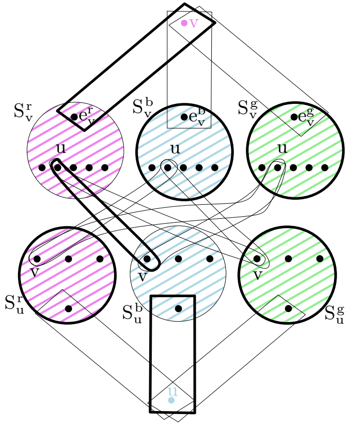
- Die Konstruktion ist polynomial.

Noch zu zeigen:

- G ist 3-färbbar $\Leftrightarrow (X, \mathcal{S})$ hat exakte Überdeckung

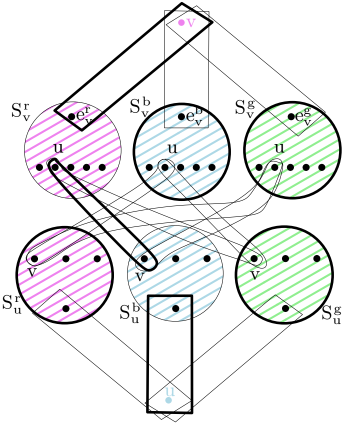
G 3-färbbar $\Rightarrow (X, S)$ hat exakte Überdeckung

- Sei $\chi : V \to C$ eine zulässige Dreifärbung.
- $\mathcal{S}'$ enthalte für jedes $v \in V$ die Mengen $\{v, e_v^{\chi(v)}\}$ und S_v^c mit $c \neq \chi(v)$.
- Diese Mengen überdecken alle Elemente exakt, außer den Elementen der Form $u_v^{\chi(v)}, v_u^{\chi(u)}$ für $\{u, v\} \in E$.
- Daher enthalte $\mathcal{S}'$ für jede Kante $\{u, v\} \in E$ die Menge $\{u_v^{\chi(v)}, v_u^{\chi(u)}\}$.
- Diese Menge existiert, da $\chi(u) \neq \chi(v)$, und damit überdeckt $\mathcal{S}'$ jedes Element aus X genau einmal.



G 3-färbbar $\Leftarrow (X, S)$ hat exakte Überdeckung

- Sei also $\mathcal{S}'$ eine exakte Überdeckung.
- Jedes Element v muss von genau einer Menge der Form $\{v, e_v^c\}$ überdeckt sein.
- Dies induziert eine Färbung χ von G mit den Farben r, b und g.
- Wir müssen beweisen, dass diese Färbung zulässig ist.
- Da für jedes v bereits $\{v, e_v^{\chi(v)}\} \in S'$, kann e_v^c mit $c \neq \chi(v)$ nur durch die Menge S_v^c überdeckt werden.



G 3-färbbar $\Leftarrow (X, S)$ hat exakte Überdeckung

Karlsruher Institut für Technologie

- Da für jedes v bereits $\{v, e_v^{\chi(v)}\} \in S'$, kann e_v^c mit $c \neq \chi(v)$ nur durch die Menge S_v^c überdeckt werden.
- Da die Mengen der Form $\{v, e_v^{\chi(v)}\}$ und S_v^c, $c \neq \chi(v)$, alle Elemente außer den $u_v^{\chi(v)}$ mit $\{u, v\} \in E$ überdecken, müssen auch die Mengen $\{u_v^{\chi(v)}, v_u^{\chi(u)}\}$ für $\{u, v\} \in E$ in $\mathcal{S}'$ enthalten sein.
- Für diese gilt per Konstruktion $\chi(v) \neq \chi(u)$.