Theoretische Grundlagen der Informatik

Vorlesung am 13.12.2022

Torsten Ueckerdt | 13. Dezember 2022

Letzte Vorlesung

Entscheidungsproblem Π ist $\mathcal{NP}$-vollständig wenn

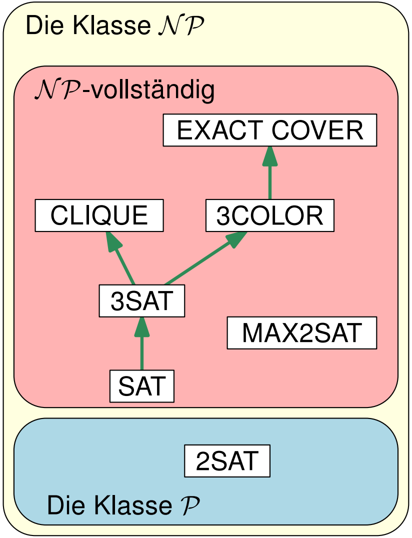
- $\Pi \in \mathcal{NP}$
- $\Pi' \propto \Pi$ für ein $\mathcal{NP}$-vollständiges Problem Π'

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

- Die $\mathcal{NP}$-vollständigen Probleme sind die “schwierigsten” Probleme in $\mathcal{NP}$.
- Solange $\mathcal{P} \neq \mathcal{NP}$ (wovon wir ausgehen), kann ein $\mathcal{NP}$-vollständiges Problem nicht von einer DTM in polynomialer Laufzeit entschieden werden.

Der Plan

- 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
  $\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$
  $\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER

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, so dass 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

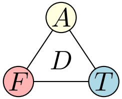
Karlsruher Institut für Technologie

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 $G = f(I)$.
- Es soll gelten: I ist erfüllbar $\Leftrightarrow$ $G = f(I)$ ist 3-färbbar.

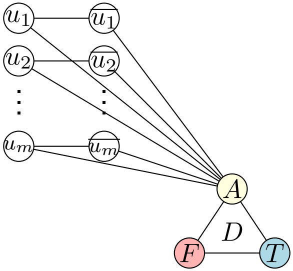
Konstruktion von 3COLOR-Instanz $G = f(I)$

- Ein 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: $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`



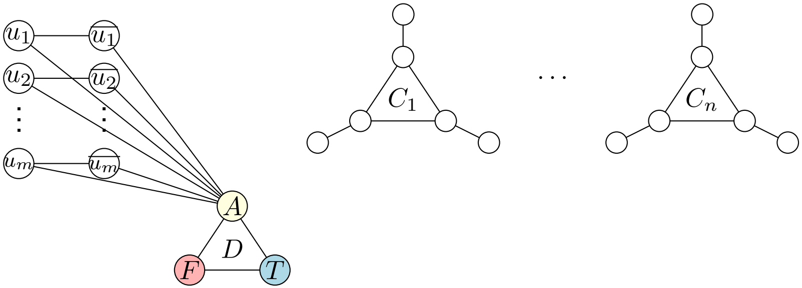
Konstruktion von 3COLOR-Instanz $G = f(I)$

- Ein 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: $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `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.



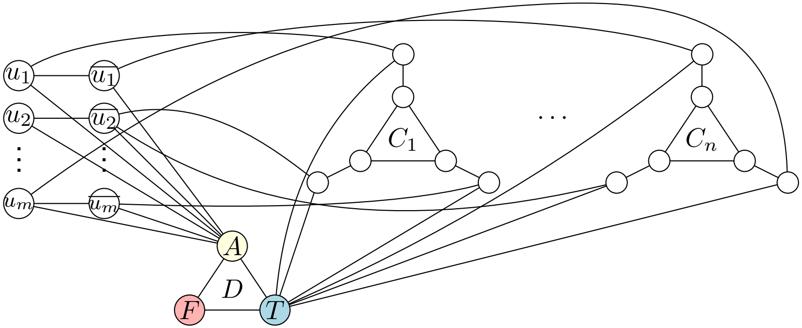
Konstruktion von 3COLOR-Instanz $G = f(I)$

- 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".



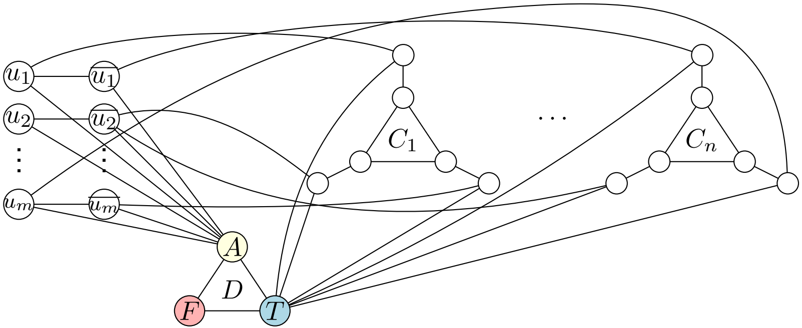
Konstruktion von 3COLOR-Instanz $G = f(I)$

- 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

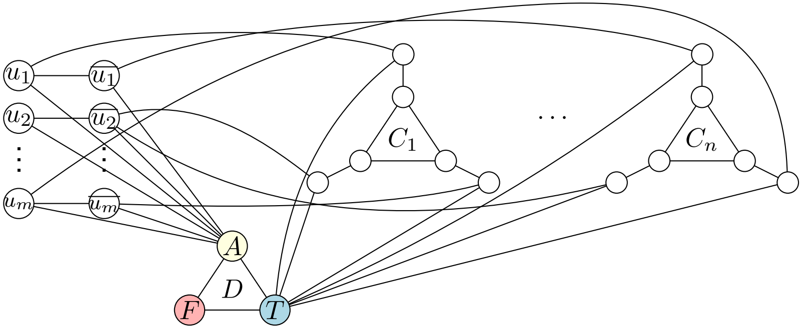
- Die Knotenanzahl von $G = f(I)$ liegt in $\mathcal{O}(n + m)$.
- Deswegen ist die Transformation polynomial.



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

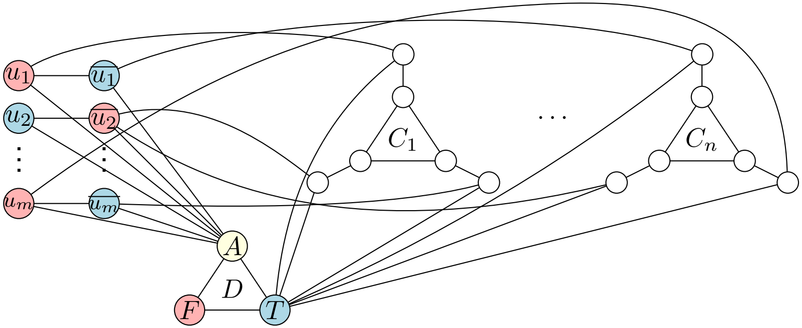
Instanz I erfüllbar $\Rightarrow$ Instanz $G = f(I)$ 3-färbbar

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



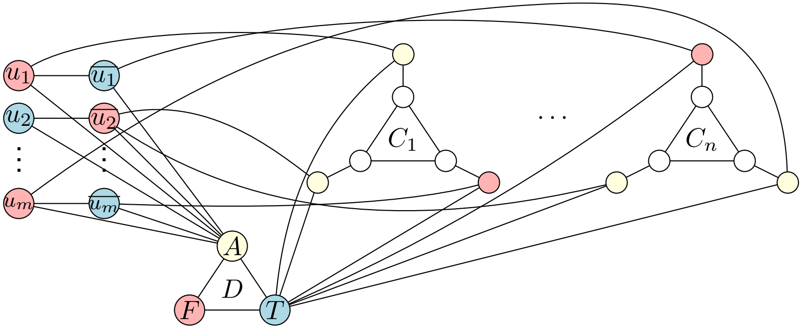
Instanz I erfüllbar $\Rightarrow$ Instanz $G = f(I)$ 3-färbbar

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



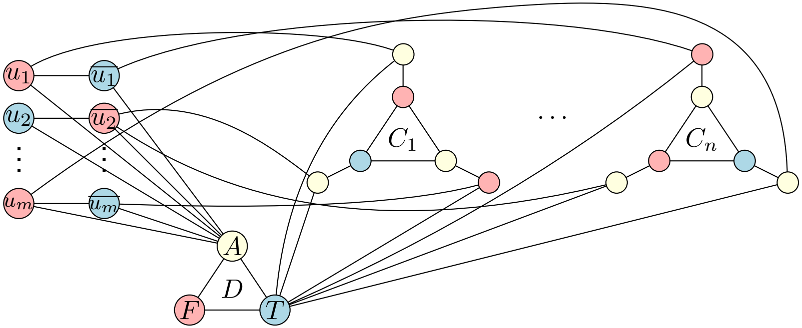
Instanz I erfüllbar $\Rightarrow$ Instanz $G = f(I)$ 3-färbbar

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



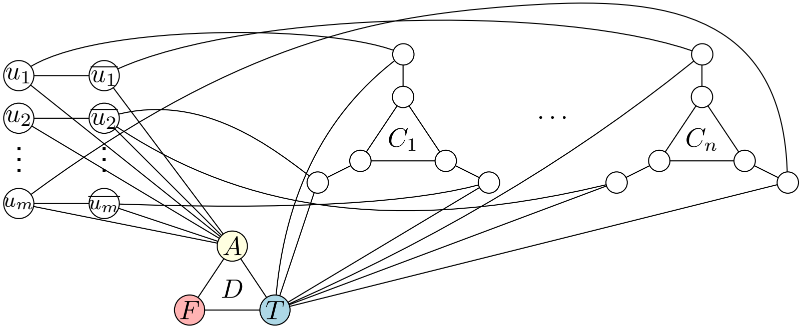
Instanz I erfüllbar $\Rightarrow$ Instanz $G = f(I)$ 3-färbbar

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



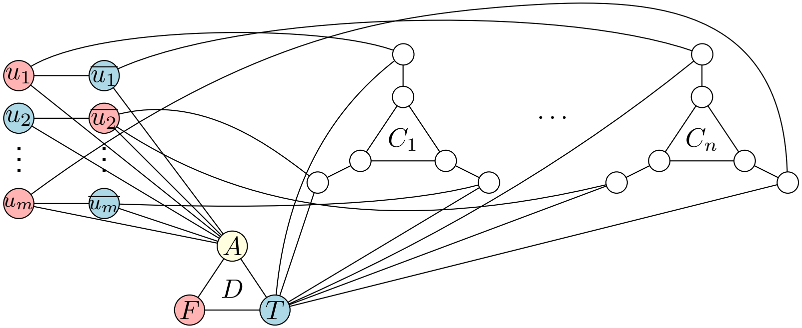
Instanz I erfüllbar $\Leftarrow$ Instanz $G = f(I)$ 3-färbbar

- Betrachte 3-Färbung von $G = f(I)$. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.



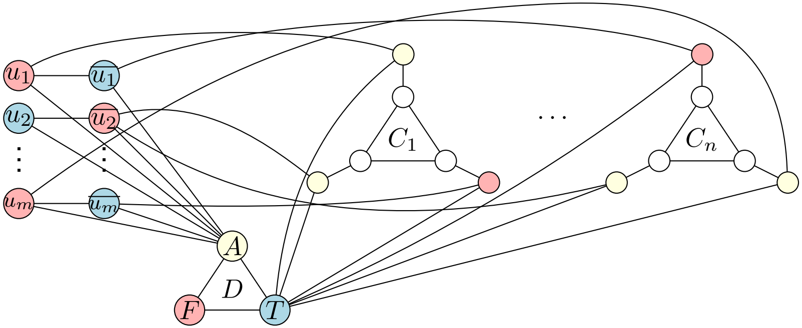
Instanz I erfüllbar $\Leftarrow$ Instanz $G = f(I)$ 3-färbbar

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



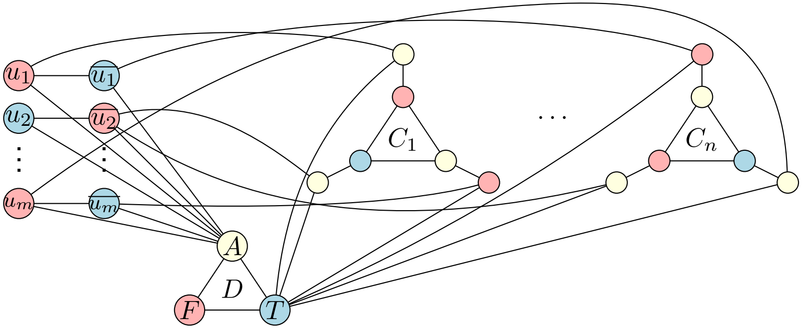
Instanz I erfüllbar $\Leftarrow$ Instanz $G = f(I)$ 3-färbbar

- Betrachte 3-Färbung von $G = f(I)$. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.
- Färbung von Literal-Knoten induziert Wahrheitsbelegung t von I. Kein Satellit hat Farbe von T.



Instanz I erfüllbar $\Leftarrow$ Instanz $G = f(I)$ 3-färbbar

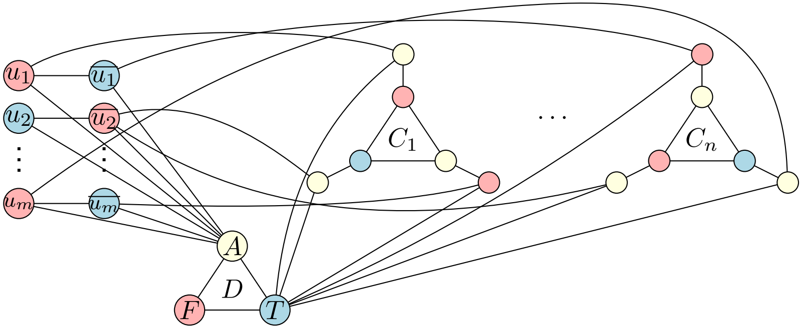
- Betrachte 3-Färbung von $G = f(I)$. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.
- Färbung von Literal-Knoten induziert Wahrheitsbelegung t von I. Kein Satellit hat Farbe von T.
- Nicht alle Satelliten sind in Farbe von A wegen des inneren Dreiecks.



Instanz I erfüllbar $\Leftarrow$ Instanz $G = f(I)$ 3-färbbar

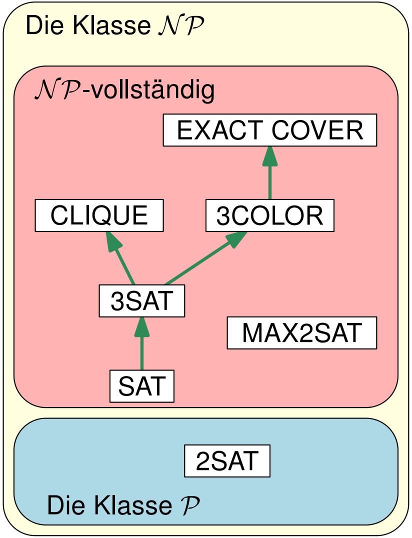
- Betrachte 3-Färbung von $G = f(I)$. Interpretiere $T \longleftrightarrow$ `wahr`, $F \longleftrightarrow$ `falsch`.
- Färbung von Literal-Knoten induziert Wahrheitsbelegung t von I. Kein Satellit hat Farbe von T.
- Nicht alle Satelliten sind in Farbe von A wegen des inneren Dreiecks.

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



Der Plan

- 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
  $\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$
  $\rightsquigarrow$ 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 $\mathcal{S}' \subseteq \mathcal{S}$, so dass 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 $\mathcal{S}' \subseteq \mathcal{S}$, so dass 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 $\mathcal{S}' \subseteq \mathcal{S}$, so dass jedes Element aus X in genau einer Menge aus $\mathcal{S}'$ liegt?

Beispiel:

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

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

Karlsruher Institut für Technologie

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

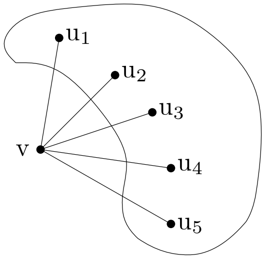
3COLOR $\propto$ EXACT COVER

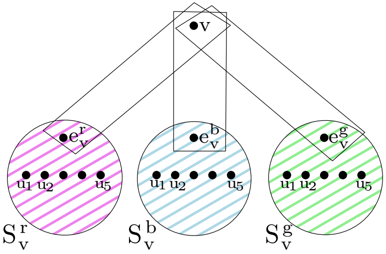
- Sei $G = (V, E)$ eine 3COLOR-Instanz.
- Wir konstruieren in Polynomialzeit eine EXACT COVER-Instanz $(X, \mathcal{S})$.
- Es soll gelten: G ist 3-färbbar $\Leftrightarrow$ $(X, \mathcal{S})$ hat exakte Überdeckung

Konstruktion von (X, S)

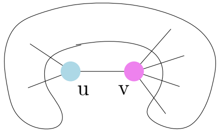
- Sei $C = \{\text{rot}, \text{blau}, \text{grün}\}$. Sei $N(v) := \{u \in V : \{u, v\} \in E\}$ die Nachbarschaft von v.
- Für jedes $v \in V$:
  - ein "Element" v in X und $3 \cdot |N(v)| + 3$ zusätzliche Elemente,
  - drei disjunkte Mengen S_v^r, S_v^b, S_v^g in $\mathcal{S}$ mit jeweils $|N(v)| + 1$ Elementen,
  - drei 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 $e_v^g \in S_v^g$

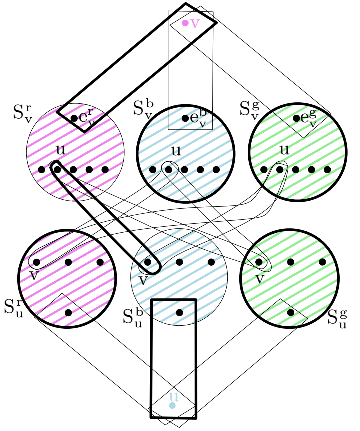
Interpretation: S_v^r entspricht der "Farbe" rot, enthält für jeden Knoten aus $N(v)$ eine Kopie und einen zusätzlichen Knoten e_v^r.





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.

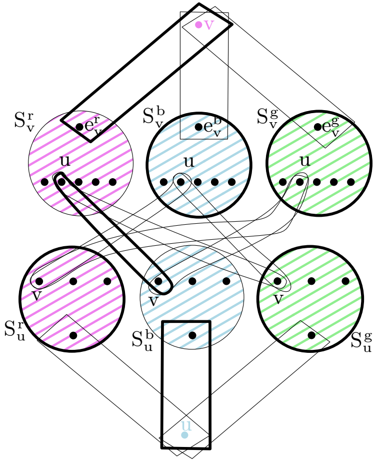
Konstruktion von (X, S)

- Die Konstruktion ist polynomial.

Noch zu zeigen:

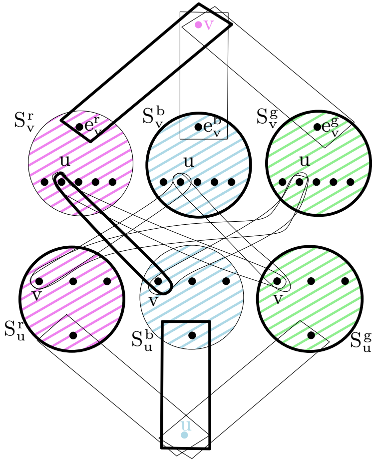
- G ist 3-färbbar $\Leftrightarrow (X, \mathcal{S})$ hat exakte Überdeckung

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



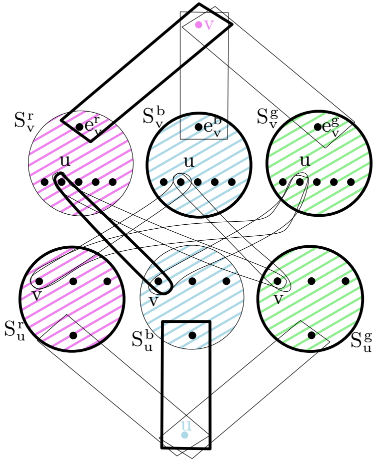
- Sei $\chi : V \to C$ eine zulässige Dreifärbung.
- Nehme für jeden Knoten $v \in V$ die Mengen $\{v, e_v^{\chi(v)}\}$ und S_v^c mit $c \neq \chi(v)$.
- Nehme für jede Kante $\{u, v\} \in E$ die Menge$\{u_v^{\chi(v)}, v_u^{\chi(u)}\}$. Diese Menge existiert, da $\chi(u) \neq \chi(v)$.
- Diese Mengen $\mathcal{S}'$ überdecken 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 rot, blau, grün.
- 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



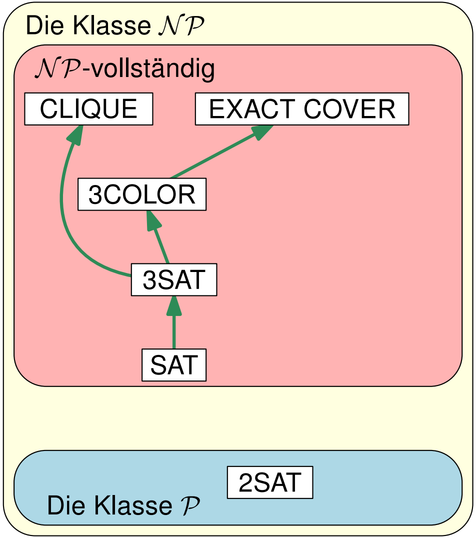
- Da für jedes v bereits $\{v, e_v^{\chi(v)}\} \in S'$, kann e_v^c mit $c \neq \chi(v)$ nur durch die Menge S_v^c überdeckt werden.
- Da die Mengen der Form $\{v, e_v^{\chi(v)}\}$ und S_v^c, $c \neq \chi(v)$, alle Elemente außer den $u_v^{\chi(v)}$ mit $\{u, v\} \in E$ überdecken, müssen auch die Mengen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

Der Plan

- EXACT COVER ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$

$\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER



Der Plan

- EXACT COVER ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$

$\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER

- SUBSET SUM ist $\mathcal{NP}$-vollständig

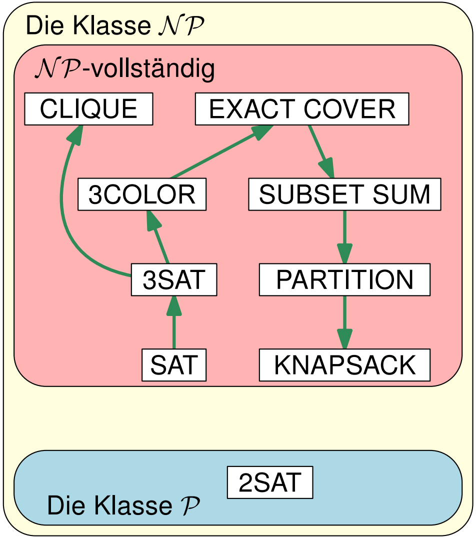
$\rightsquigarrow$ SUBSET SUM $\in \mathcal{NP}$

$\rightsquigarrow$ EXACT COVER $\propto$ SUBSET SUM

- PARTITION ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ PARTITION $\in \mathcal{NP}$

$\rightsquigarrow$ SUBSET SUM $\propto$ PARTITION



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

$\rightsquigarrow$ KNAPSACK $\in \mathcal{NP}$

$\rightsquigarrow$ PARTITION $\propto$ KNAPSACK

Das Problem SUBSET SUM

Problem SUBSET SUM

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w \colon M \longrightarrow \mathbb{N}_0$,
eine Zahl $K \in \mathbb{N}_0$

Frage: Existiert eine Teilmenge $M' \subseteq M$ mit $\sum_{a \in M'} w(a) = K$?

Satz.

Problem SUBSET SUM ist $\mathcal{NP}$-vollständig.

Das Problem SUBSET SUM

Problem SUBSET SUM

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w\colon M \longrightarrow \mathbb{N}_0$,
eine Zahl $K \in \mathbb{N}_0$

Frage: Existiert eine Teilmenge $M' \subseteq M$ mit $\sum_{a \in M'} w(a) = K$?

Satz.

Problem SUBSET SUM ist $\mathcal{NP}$-vollständig.

Notation:

$$w(X) := \sum_{a \in X} w(a)$$

Das Problem SUBSET SUM

Problem SUBSET SUM

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w\colon M \longrightarrow \mathbb{N}_0$,
eine Zahl $K \in \mathbb{N}_0$

Frage: Existiert eine Teilmenge $M' \subseteq M$ mit $w(M') = K$?

Satz.

Problem SUBSET SUM ist $\mathcal{NP}$-vollständig.

Notation:

$$w(X) := \sum_{a \in X} w(a)$$

$\mathcal{NP}$-Vollständigkeit von SUBSET SUM

SUBSET SUM $\in \mathcal{NP}$.

- Es kann für eine gegebene Teilmenge $M' \subseteq M$ in Polynomialzeit der Wert $w(M') = \sum_{a \in M'} w(a)$ ausgerechnet und mit K verglichen werden.

Beweis: $\mathcal{NP}$-Vollständigkeit von SUBSET SUM

EXACT COVER $\propto$ SUBSET SUM

- Sei $(X = \{0, 1, \ldots, m-1\}, \mathcal{S})$ eine beliebige EXACT COVER-Instanz.
- Konstruiere SUBSET SUM Instanz (M, w, K)

$$
\begin{aligned}
M &:= \mathcal{S} \\
\#x &:= \#\{Y \in \mathcal{S} \mid x \in Y\} \\
p &:= \max\{\#x + 1 \mid x \in X\} \\
w(x) &:= p^x \\
K &:= w(X) = \sum_{x=0}^{m-1} p^x
\end{aligned}
$$

- Die Konstruktion benötigt nur Polynomialzeit.

Beweis: $\mathcal{NP}$-Vollständigkeit von SUBSET SUM

$$
\begin{aligned}
M &:= \mathcal{S} \\
\#x &:= \#\{Y \in \mathcal{S} \mid x \in Y\} \\
p &:= \max\{\#x + 1 \mid x \in X\} \\
w(x) &:= p^x \\
K &:= w(X) = \sum_{x=0}^{m-1} p^x
\end{aligned}
$$

Beispiel:

$$
\begin{aligned}
X &= \{0, \dots, 6\} \\
Y &= \{0, 2, 5\} \in \mathcal{S} \\
\hookrightarrow w(Y) &= p^0 + p^2 + p^5 \\
\hookrightarrow w(Y) &= 0100101_p \\
K &= 1111111_p
\end{aligned}
$$

Veranschaulichung:

- Wir stellen die Mengenzugehörigkeiten als Zahlen zur Basis p dar.
- Kodiere $w(Y)$ für $Y \in \mathcal{S}$ als 01-String der Länge m, wobei an $(m - i)$-ter Stelle eine 1 steht genau dann, wenn $i \in Y$;
- entsprechend ist K ein String der Länge m aus Einsen.

Beweis: $\mathcal{NP}$-Vollständigkeit von SUBSET SUM

$$
\begin{aligned}
M &:= \mathcal{S} \\
\#x &:= \#\{Y \in \mathcal{S} \mid x \in Y\} \\
p &:= \max\{\#x + 1 \mid x \in X\} \\
w(x) &:= p^x \\
K &:= w(X) = \sum_{x=0}^{m-1} p^x
\end{aligned}
$$

Beispiel:

$$
\begin{aligned}
X &= \{0, \ldots, 6\} \\
Y &= \{0, 2, 5\} \in \mathcal{S} \\
\hookrightarrow w(Y) &= p^0 + p^2 + p^5 \\
\hookrightarrow w(Y) &= 0100101_p \\
K &= 1111111_p
\end{aligned}
$$

Veranschaulichung:

- Komponentenweise Addition der Strings $w(Y_1), \ldots, w(Y_n)$ ergibt einen String der Länge m, an dessen $(m - i)$-ter Stelle steht in wievielen der Y_j, $j = 1, \ldots, n$, das Element i vorkommt.
- $\sum_{Y \in \mathcal{S}'} w(Y) = K$ bedeutet also, dass jedes $x \in X$ in genau einem $Y \in \mathcal{S}'$ vorkommt.

Beweis: $\mathcal{NP}$-Vollständigkeit von SUBSET SUM

$$\begin{aligned}
M &:= \mathcal{S} \\
\#x &:= \#\{Y \in \mathcal{S} \mid x \in Y\} \\
p &:= \max\{\#x + 1 \mid x \in X\} \\
w(x) &:= p^x \\
K &:= w(X) = \sum_{x=0}^{m-1} p^x
\end{aligned}$$

Beispiel:

$$\begin{aligned}
X &= \{0, \ldots, 6\} \\
Y &= \{0, 2, 5\} \in \mathcal{S} \\
\hookrightarrow w(Y) &= p^0 + p^2 + p^5 \\
\hookrightarrow w(Y) &= 0100101_p \\
K &= 1111111_p
\end{aligned}$$

Beispiel:

$\mathcal{S} = \{Y_1 = \{0, 2, 5\}, Y_2 = \{0, 1, 2\}, Y_3 = \{2, 4, 5\}, Y_4 = \{3, 4, 6\}\}$

$\#0 = 2$, $\#1 = 1$, $\#2 = 3$, $\#3 = 1$, $\#4 = 2$, $\#5 = 1$, $\#6 = 1$, $p = 4$

$w(Y_1) = 0100101_4$, $w(Y_2) = 0000111_4$

$w(Y_3) = 0110100_4$, $w(Y_4) = 1011000_4$.

Beweis: $\mathcal{NP}$-Vollständigkeit von SUBSET SUM

$$
\begin{aligned}
M &:= \mathcal{S} \\
\#x &:= \#\{Y \in \mathcal{S} \mid x \in Y\} \\
p &:= \max\{\#x + 1 \mid x \in X\} \\
w(x) &:= p^x \\
K &:= w(X) = \sum_{x=0}^{m-1} p^x
\end{aligned}
$$

Beispiel:

$$
\begin{aligned}
X &= \{0, \ldots, 6\} \\
Y &= \{0, 2, 5\} \in \mathcal{S} \\
\hookrightarrow w(Y) &= p^0 + p^2 + p^5 \\
\hookrightarrow w(Y) &= 0100101_p \\
K &= 1111111_p
\end{aligned}
$$

- $(X, \mathcal{S})$ lösbar $\Rightarrow (M, w, K)$ lösbar.

Sei $\mathcal{S}' \subseteq \mathcal{S}$ exakte Überdeckung von $(X, \mathcal{S})$. Dann gilt

$$
\sum_{Y \in \mathcal{S}'} w(Y) = \sum_{Y \in \mathcal{S}'} \sum_{x \in Y} p^x = \sum_{x=0}^{m-1} p^x = K,
$$

da jedes $x \in X$ genau einmal überdeckt wird.
$\rightsquigarrow \mathcal{S}'$ erfüllt die Bedingung für SUBSET SUM.

Beweis: $\mathcal{NP}$-Vollständigkeit von SUBSET SUM

$$
\begin{aligned}
M &:= \mathcal{S} \\
\#x &:= \#\{Y \in \mathcal{S} \mid x \in Y\} \\
p &:= \max\{\#x + 1 \mid x \in X\} \\
w(x) &:= p^x \\
K &:= w(X) = \sum_{x=0}^{m-1} p^x
\end{aligned}
$$

Beispiel:

$$
\begin{aligned}
X &= \{0, \dots, 6\} \\
Y &= \{0, 2, 5\} \in \mathcal{S} \\
\hookrightarrow w(Y) &= p^0 + p^2 + p^5 \\
\hookrightarrow w(Y) &= 0100101_p \\
K &= 1111111_p
\end{aligned}
$$

- $(X, \mathcal{S})$ lösbar $\Leftarrow (M, w, K)$ lösbar.

Ist $\mathcal{S}' \subseteq M = \mathcal{S}$ eine geeignete Menge für SUBSET SUM, so gilt

$$\sum_{Y \in \mathcal{S}'} w(Y) = K = \sum_{x=0}^{m-1} p^x.$$

Nach Wahl von p kommt jedes $x \in X$ in genau einem $Y \in \mathcal{S}'$ vor.
$\rightsquigarrow \mathcal{S}'$ ist damit eine exakte Überdeckung von $(X, \mathcal{S})$.

Der Plan

- EXACT COVER ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$

$\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER

- SUBSET SUM ist $\mathcal{NP}$-vollständig

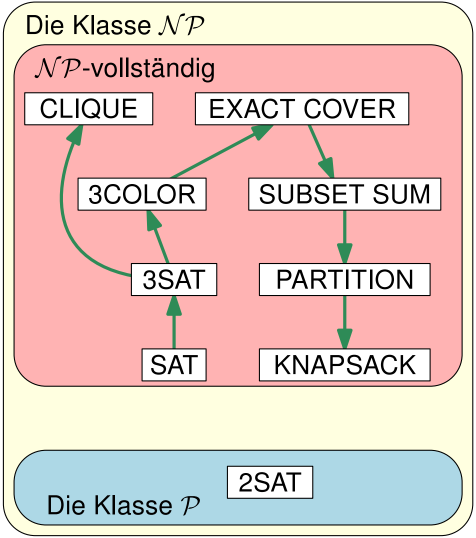
$\rightsquigarrow$ SUBSET SUM $\in \mathcal{NP}$

$\rightsquigarrow$ EXACT COVER $\propto$ SUBSET SUM

- PARTITION ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ PARTITION $\in \mathcal{NP}$

$\rightsquigarrow$ SUBSET SUM $\propto$ PARTITION



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

$\rightsquigarrow$ KNAPSACK $\in \mathcal{NP}$

$\rightsquigarrow$ PARTITION $\propto$ KNAPSACK

Das Problem PARTITION

Problem PARTITION

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w\colon M \to \mathbb{N}_0$

Frage: Existiert eine Teilmenge $M' \subseteq M$ mit
$\sum_{a \in M'} w(a) = \sum_{a \in M \setminus M'} w(a)$, d.h. $w(M') = w(M \setminus M')$?

Satz.

Problem PARTITION ist $\mathcal{NP}$-vollständig.

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

PARTITION $\in \mathcal{NP}$.

- Für eine Menge M' können in Polynomialzeit die Werte $w(M') = \sum_{a \in M'} w(a)$ und $w(M \setminus M') = \sum_{a \in M \setminus M'} w(a)$ ausgerechnet und verglichen werden.

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

SUBSET SUM $\propto$ PARTITION.

- Sei (M, w, K) eine beliebige SUBSET SUM-Instanz.
- Konstruiere PARTITION-Instanz (M^*, w^*)

$$\begin{aligned}
N &:= w(M) + 1 \\
M^* &:= M \cup \{b, c\} \\
w^*(a) &:= w(a) \qquad \text{für } a \in M \\
w^*(b) &:= N - K \\
w^*(c) &:= K + 1
\end{aligned}$$

- Die Konstruktion benötigt nur Polynomialzeit.

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

$$\begin{aligned} N &:= w(M) + 1 \\ M^* &:= M \cup \{b, c\} \\ w^*(a) &:= w(a) \qquad \text{für } a \in M \\ w^*(b) &:= N - K \\ w^*(c) &:= K + 1 \end{aligned}$$

Beispiel:

$$\begin{aligned} M &= \{i, j, k, \ell\} \\ w(j) &= 13 \\ &\downarrow \\ M^* &= \{i, j, k, \ell, b, c\} \\ w^*(j) &= 13 \end{aligned}$$

- (M^*, w^*) Ja-Instanz genau dann, wenn (M, w, K) Ja-Instanz:

$$\boxed{\exists M' \subseteq M^* \colon w^*(M') = w^*(M^* \setminus M')} \iff \boxed{\exists M'' \subseteq M \colon w(M'') = K}$$

$$w^*(b) + w^*(c) = (N - K) + (K + 1) = N + 1$$

$$w^*(M^*) = (N - 1) + (N - K) + (K + 1) = 2N$$

- Also: b, c nicht beide in M' bzw. $M^* \setminus M'$ enthalten
- o.B.d.A. sei $b \in M'$ und $c \in M^* \setminus M'$

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

$$\begin{aligned} N &:= w(M) + 1 \\ M^* &:= M \cup \{b, c\} \\ w^*(a) &:= w(a) \quad \text{für } a \in M \\ w^*(b) &:= N - K \\ w^*(c) &:= K + 1 \end{aligned}$$

Beispiel:

$$\begin{aligned} M &= \{i, j, k, \ell\} \\ w(j) &= 13 \\ &\downarrow \\ M^* &= \{i, j, k, \ell, b, c\} \\ w^*(j) &= 13 \end{aligned}$$

- (M^*, w^*) Ja-Instanz genau dann, wenn (M, w, K) Ja-Instanz:

$$\boxed{\exists M' \subseteq M^* \colon w^*(M') = w^*(M^* \setminus M')} \iff \boxed{\exists M'' \subseteq M \colon w(M'') = K}$$

"$\Longrightarrow$"
- Sei $M' \subseteq M^*$, so dass $w^*(M') = w^*(M^* \setminus M')$ mit $b \in M'$.
- Dann gilt $w^*(M') = N$, da $w^*(M^*) = 2N$.
- Damit erfüllt $M'' := M' \setminus \{b\}$ die Bedingung für SUBSET SUM, denn $w(M'') = w^*(M') - w^*(b) = N - (N - K) = K$.

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

$$
\begin{aligned}
N &:= w(M) + 1 \\
M^* &:= M \cup \{b, c\} \\
w^*(a) &:= w(a) \qquad \text{für } a \in M \\
w^*(b) &:= N - K \\
w^*(c) &:= K + 1
\end{aligned}
$$

Beispiel:

$$
\begin{aligned}
M &= \{i, j, k, \ell\} \\
w(j) &= 13 \\
&\downarrow \\
M^* &= \{i, j, k, \ell, b, c\} \\
w^*(j) &= 13
\end{aligned}
$$

- (M^*, w^*) Ja-Instanz genau dann, wenn (M, w, K) Ja-Instanz:

$$\boxed{\exists M' \subseteq M^* \colon\ w^*(M') = w^*(M^* \setminus M')} \Longleftrightarrow \boxed{\exists M'' \subseteq M \colon\ w(M'') = K}$$

"$\Longleftarrow$"
- Sei M'', so dass $w(M'') = K$.
- Dann erfüllt $M' := M'' \cup \{b\}$ die Bedingung für PARTITION, denn $w^*(M') = w(M') + w^*(b) = K + (N - K) = N$.

Der Plan

- EXACT COVER ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$

$\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER

- SUBSET SUM ist $\mathcal{NP}$-vollständig

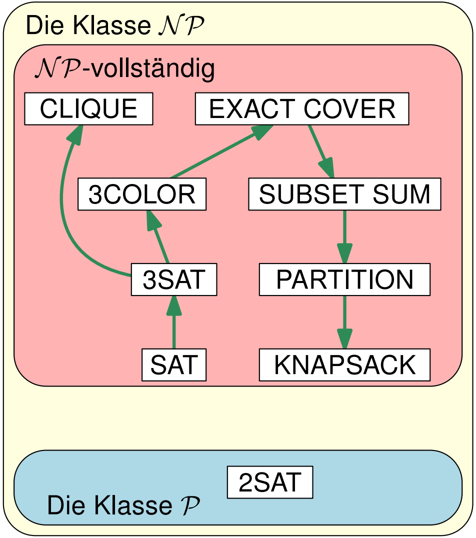
$\rightsquigarrow$ SUBSET SUM $\in \mathcal{NP}$

$\rightsquigarrow$ EXACT COVER $\propto$ SUBSET SUM

- PARTITION ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ PARTITION $\in \mathcal{NP}$

$\rightsquigarrow$ SUBSET SUM $\propto$ PARTITION



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

$\rightsquigarrow$ KNAPSACK $\in \mathcal{NP}$

$\rightsquigarrow$ PARTITION $\propto$ KNAPSACK

Das Problem KNAPSACK

Problem KNAPSACK

Gegeben: Eine endliche Menge M,
eine Gewichtsfunktion $w\colon M \longrightarrow \mathbb{N}_0$,
eine Kostenfunktion $c\colon M \longrightarrow \mathbb{N}_0$
Zahlen $W, C \in \mathbb{N}_0$

Frage: Existiert eine Teilmenge $M' \subseteq M$ mit $w(M') \leq W$ und $c(M') \geq C$?

Satz.

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

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

KNAPSACK $\in \mathcal{NP}$.

- Für eine Menge M' kann in Polynomialzeit überprüft werden, ob
  - $w(M') = \sum_{a \in M'} w(a) \leq W$ und
  - $c(M') = \sum_{a \in M'} c(a) \geq C$

  gilt.

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

PARTITION $\propto$ KNAPSACK.

- Sei (M, w) eine PARTITION-Instanz.
- Konstruiere KNAPSACK-Instanz (M, w', c, W, C)

$$\begin{aligned} w' &:= 2w \\ c &:= 2w \\ W = C &:= w(M) = \sum_{a \in M} w(a) \end{aligned}$$

- Die Konstruktion benötigt nur Polynomialzeit.
- Es ist (M, w) genau dann eine Ja-Instanz, wenn (M, w', c, W, C) eine Ja-Instanz ist (ohne Beweis).

Der Plan

- EXACT COVER ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$

$\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER

- SUBSET SUM ist $\mathcal{NP}$-vollständig

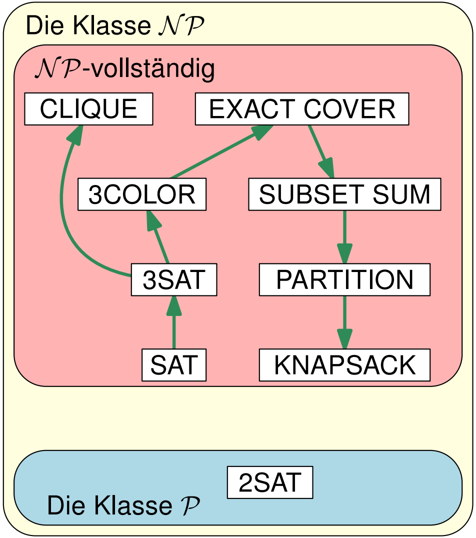
$\rightsquigarrow$ SUBSET SUM $\in \mathcal{NP}$

$\rightsquigarrow$ EXACT COVER $\propto$ SUBSET SUM

- PARTITION ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ PARTITION $\in \mathcal{NP}$

$\rightsquigarrow$ SUBSET SUM $\propto$ PARTITION



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

$\rightsquigarrow$ KNAPSACK $\in \mathcal{NP}$

$\rightsquigarrow$ PARTITION $\propto$ KNAPSACK

Der Plan

- EXACT COVER ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ EXACT COVER $\in \mathcal{NP}$

$\rightsquigarrow$ 3COLOR $\propto$ EXACT COVER

- SUBSET SUM ist $\mathcal{NP}$-vollständig

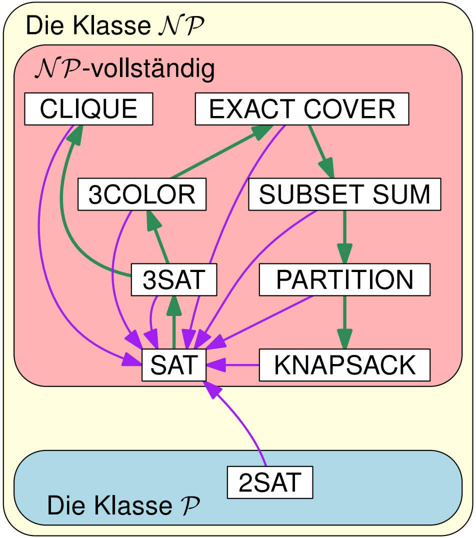
$\rightsquigarrow$ SUBSET SUM $\in \mathcal{NP}$

$\rightsquigarrow$ EXACT COVER $\propto$ SUBSET SUM

- PARTITION ist $\mathcal{NP}$-vollständig

$\rightsquigarrow$ PARTITION $\in \mathcal{NP}$

$\rightsquigarrow$ SUBSET SUM $\propto$ PARTITION



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

$\rightsquigarrow$ KNAPSACK $\in \mathcal{NP}$

$\rightsquigarrow$ PARTITION $\propto$ KNAPSACK

Auswirkung auf die Frage $\mathcal{P} = \mathcal{NP}$

Karlsruher Institut für Technologie

- Wir haben gesehen, dass es für je zwei $\mathcal{NP}$-vollständige Probleme eine polynomiale Transformation von einem zum anderen Problem gibt.
- Deshalb sind alle $\mathcal{NP}$-vollständigen Probleme im wesentlichen gleich schwer.
- Dies hat Auswirkungen auf die Frage, ob $\mathcal{P} = \mathcal{NP}$ ist.

Satz.

Sei L eine $\mathcal{NP}$-vollständige Sprache. Dann gilt:

- $L \in \mathcal{P} \Longrightarrow \mathcal{P} = \mathcal{NP}$
- $L \notin \mathcal{P} \Longrightarrow$ für jede $\mathcal{NP}$-vollständige Sprache L' gilt $L' \notin \mathcal{P}$

Auswirkung auf die Frage $\mathcal{P} = \mathcal{NP}$

Satz.

Sei L eine $\mathcal{NP}$-vollständige Sprache. Dann gilt:

- $L \in \mathcal{P} \Longrightarrow \mathcal{P} = \mathcal{NP}$
- $L \notin \mathcal{P} \Longrightarrow$ für jede $\mathcal{NP}$-vollständige Sprache L' gilt $L' \notin \mathcal{P}$

Beweis Teil 1:

- Da $L \in \mathcal{P}$, existiert eine polynomiale DTM $\mathcal{M}$ für L.
- Da L $\mathcal{NP}$-vollständig, gibt es für jede Sprache $L' \in \mathcal{NP}$ eine polynomiale Transformation $L' \propto L$.
- Hintereinanderausführung von $L' \propto L$ und $\mathcal{M}$ liefert eine polynomiale DTM für L'.
- Damit ist jede Sprache $L' \in \mathcal{NP}$ auch in $\mathcal{P}$.

Auswirkung auf die Frage $\mathcal{P} = \mathcal{NP}$

Satz.

Sei L eine $\mathcal{NP}$-vollständige Sprache. Dann gilt:

- $L \in \mathcal{P} \Longrightarrow \mathcal{P} = \mathcal{NP}$
- $L \notin \mathcal{P} \Longrightarrow$ für jede $\mathcal{NP}$-vollständige Sprache L' gilt $L' \notin \mathcal{P}$

Beweis Teil 2:

- Sei $L \notin \mathcal{P}$ und L $\mathcal{NP}$-vollständig.
- Angenommen für eine $\mathcal{NP}$-vollständige Sprache L' gilt: $L' \in \mathcal{P}$
- Dann folgt aus Teil 1 des Satzes $\mathcal{P} = \mathcal{NP}$.
- Dies ist aber ein Widerspruch zur Voraussetzung $L \notin \mathcal{P}$.

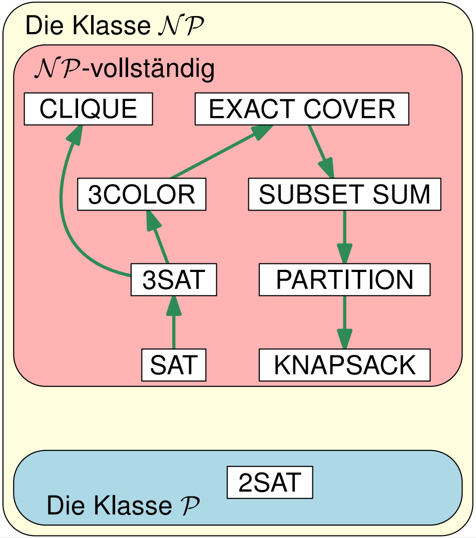
Zusammenfassung

- Die Klasse $\mathcal{P}$ ist die Klasse aller Entscheidungsprobleme/Sprachen die mit einer **deterministischen** Turingmaschine in polynomieller Zeit gelöst werden können.
- Die Klasse $\mathcal{NP}$ ist die Klasse aller Entscheidungsprobleme/Sprachen die mit einer **nicht-deterministischen** Turingmaschine in polynomieller Zeit gelöst werden können.
- Informell ausgedrückt gehört Π zu $\mathcal{NP}$, falls Π folgende Eigenschaft hat:
  Ist die Antwort bei Eingabe eines Instanz I von Π Ja, dann kann die Korrektheit eines Beweises (Zeugen) dafür in polynomialer Zeit überprüft werden.

Zusammenfassung

- Eine polynomiale Transformation einer Sprache $L_1 \subseteq \Sigma_1^*$ in eine Sprache $L_2 \subseteq \Sigma_2^*$ ist eine Funktion $f\colon \Sigma_1^* \longrightarrow \Sigma_2^*$ mit den Eigenschaften:
  - es existiert eine polynomiale deterministische Turing-Maschine, die f berechnet;
  - für alle $x \in \Sigma_1^*$ gilt: $x \in L_1 \Leftrightarrow f(x) \in L_2$.
- Eine Sprache L heißt $\mathcal{NP}$-vollständig, falls gilt:
  - $L \in \mathcal{NP}$ und
  - für alle $L' \in \mathcal{NP}$ gilt $L' \propto L$ ($\mathcal{NP}$-Schwere).
- **Bedeutung**:
  Unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ gibt es kein polynomielles Lösungsverfahren für ein $\mathcal{NP}$-vollständiges Problem.

Zusammenfassung



- Mit dem Satz von Cook haben wir direkt gezeigt, dass das Problem SAT $\mathcal{NP}$-schwer ist.
- Bei allen anderen Problemen haben wir polynomielle Transformationen (Reduktionen) benutzt um die $\mathcal{NP}$-Schwere nachzuweisen:

  SAT $\propto$ 3SAT $\propto$ 3COLOR $\propto$ EXACT COVER

  $\propto$ SUBSET SUM $\propto$ PARTITION $\propto$ KNAPSACK

Ein Blick über den Tellerrand

- Entscheidungsprobleme außerhalb von $\mathcal{P}$ und $\mathcal{NP}$
- nächste Vorlesung: Probleme die nicht Entscheidungsprobleme sind

Die Klassen $\mathcal{NPC}$ und $\mathcal{NPI}$

- Die Klasse $\mathcal{NPC}$ ($\mathcal{NP}$-complete) sei die Klasse der $\mathcal{NP}$-vollständigen Sprachen/Probleme.
- Die Klasse $\mathcal{NPI}$ ($\mathcal{NP}$-intermediate) ist definiert durch $\mathcal{NPI} := \mathcal{NP} \setminus (\mathcal{P} \cup \mathcal{NPC})$.

Die Klassen $\mathcal{NPC}$ und $\mathcal{NPI}$

Karlsruher Institut für Technologie

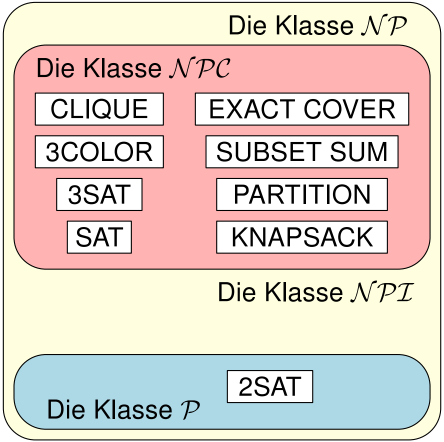
- Die Klasse $\mathcal{NPC}$ ($\mathcal{NP}$-complete) sei die Klasse der $\mathcal{NP}$-vollständigen Sprachen/Probleme.
- Die Klasse $\mathcal{NPI}$ ($\mathcal{NP}$-intermediate) ist definiert durch $\mathcal{NPI} := \mathcal{NP} \setminus (\mathcal{P} \cup \mathcal{NPC})$.

Satz von Ladner (1975).

Falls $\mathcal{P} \neq \mathcal{NP}$, so folgt $\mathcal{NPI} \neq \emptyset$.

- **Entweder:** $\mathcal{P} = \mathcal{NP} = \mathcal{NPC}$ (wovon wir nicht ausgehen)
- **Oder:** $\mathcal{P} \neq \mathcal{NP}$ und es gibt Entscheidungsprobleme Π in $\mathcal{NP}$, die weder in $\mathcal{P}$ sind, noch $\mathcal{NP}$-vollständig sind.

Vermutete Situation



Die Klassen co-$\mathcal{P}$ und co-$\mathcal{NP}$

Klasse der Komplementsprachen

- Die Klasse **co-$\mathcal{P}$** ist die Klasse aller Sprachen
$$L' = \Sigma^* \setminus L \text{ mit } L \subseteq \Sigma^* \text{ und } L \in \mathcal{P}.$$
- Die Klasse **co-$\mathcal{NP}$** ist die Klasse aller Sprachen
$$L' = \Sigma^* \setminus L \text{ mit } L \subseteq \Sigma^* \text{ und } L \in \mathcal{NP}.$$

Die Klassen co-$\mathcal{P}$ und co-$\mathcal{NP}$

Karlsruher Institut für Technologie

Klasse der Komplementsprachen

- Die Klasse **co-$\mathcal{P}$** ist die Klasse aller Sprachen
$$L' = \Sigma^* \setminus L \text{ mit } L \subseteq \Sigma^* \text{ und } L \in \mathcal{P}.$$
- Die Klasse **co-$\mathcal{NP}$** ist die Klasse aller Sprachen
$$L' = \Sigma^* \setminus L \text{ mit } L \subseteq \Sigma^* \text{ und } L \in \mathcal{NP}.$$

Wir wissen: $\mathcal{P} = \text{co-}\mathcal{P} \subseteq \text{co-}\mathcal{NP}$.

Die Klassen co-$\mathcal{P}$ und co-$\mathcal{NP}$

Klasse der Komplementsprachen

- Die Klasse **co-$\mathcal{P}$** ist die Klasse aller Sprachen

$$L' = \Sigma^* \setminus L \text{ mit } L \subseteq \Sigma^* \text{ und } L \in \mathcal{P}.$$

- Die Klasse **co-$\mathcal{NP}$** ist die Klasse aller Sprachen

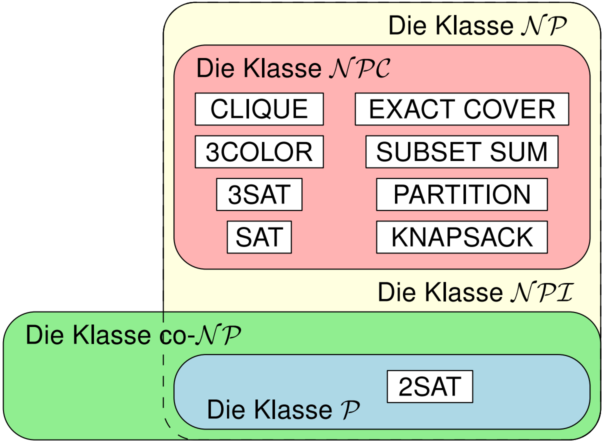
$$L' = \Sigma^* \setminus L \text{ mit } L \subseteq \Sigma^* \text{ und } L \in \mathcal{NP}.$$

Wir wissen: $\mathcal{P}$ = co-$\mathcal{P}$ $\subseteq$ co-$\mathcal{NP}$.

Frage: Gilt auch $\mathcal{NP}$ = co-$\mathcal{NP}$?

- Sollte $\mathcal{NP} \neq$ co-$\mathcal{NP}$ sein, dann würde auch $\mathcal{P} \neq \mathcal{NP}$ gelten.
- Vermutlich ist $\mathcal{NP} \neq$ co-$\mathcal{NP}$
  $\rightsquigarrow$ Verschärfung der $\mathcal{P} \neq \mathcal{NP}$-Vermutung.
- Aber auch $\mathcal{NP}$ = co-$\mathcal{NP}$ und $\mathcal{P} \neq \mathcal{NP}$ ist theoretisch möglich.

Vermutete Situation



Das TSP-Komplement-Problem

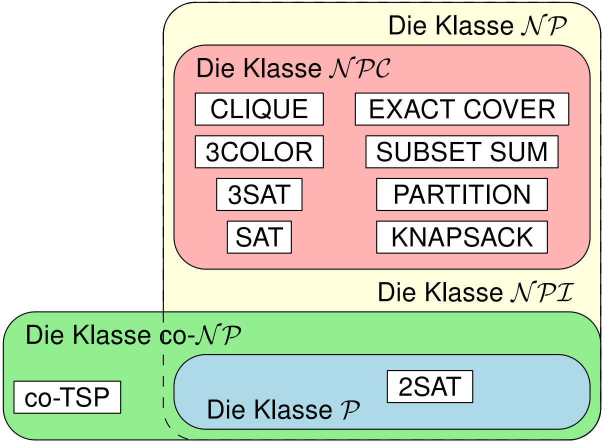
Problem co-TSP

Gegeben: Graph $G = (V, E)$,
Kantengewichtung $c\colon E \to \mathbb{Z}^+$,
Parameter $K \in \mathbb{Z}$.

Frage: Gibt es keine Tour der Länge $\leq K$?

- co-TSP ist in co-$\mathcal{NP}$ (denn TSP ist in $\mathcal{NP}$)
  $\rightsquigarrow$ Eine Nein-Instanz kann durch einen geeigneten Zeugen einfach verifiziert werden.
- Frage: Ist co-TSP in $\mathcal{NP}$?
  $\rightsquigarrow$ Kann eine Ja-Instanz durch einen geeigneten Zeugen einfach verifiziert werden? Vermutung: Nein.

Vermutete Situation



$\mathcal{NP}$-vollständig vs. co-$\mathcal{NP}$

Lemma.

Falls L $\mathcal{NP}$-vollständig ist und $L \in$ co-$\mathcal{NP}$, so ist $\mathcal{NP}$ = co-$\mathcal{NP}$.

$\mathcal{NP}$-vollständig vs. co-$\mathcal{NP}$

Lemma.

Falls L $\mathcal{NP}$-vollständig ist und $L \in$ co-$\mathcal{NP}$, so ist $\mathcal{NP} =$ co-$\mathcal{NP}$.

Beweis:

- Da $L \in$ co-$\mathcal{NP}$, existiert eine polynomiale NTM $\mathcal{M}$ für L^c.
- Da L $\mathcal{NP}$-vollständig ist, existiert für alle $L' \in \mathcal{NP}$ eine deterministische polynomiale Transformation $f_{L'} : L' \longrightarrow L$.
- Eine leichte Anpassung liefert eine deterministische polynomiale Transformation $f_{L'^c} : L'^c \longrightarrow L^c$.
- Hintereinanderausführung von $f_{L'^c}$ und $\mathcal{M}$ ergibt eine polynomiale NTM für L'^c.
- Also $L' \in$ co-$\mathcal{NP}$.

$\mathcal{NP}$-vollständig vs. co-$\mathcal{NP}$

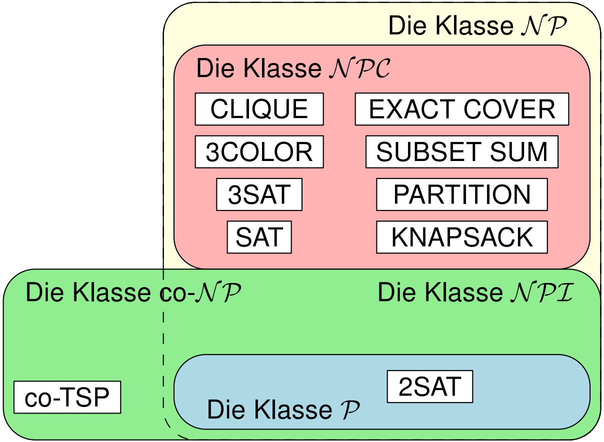
Lemma.

Falls L $\mathcal{NP}$-vollständig ist und $L \in$ co-$\mathcal{NP}$, so ist $\mathcal{NP} =$ co-$\mathcal{NP}$.

Bemerkung:

- Mit der Vermutung $\mathcal{NP} \neq$ co-$\mathcal{NP}$ folgt also $\mathcal{NPC} \cap$ co-$\mathcal{NP} = \emptyset$.
- Unter dieser Annahme:
  Wenn ein Problem in $\mathcal{NP}$ und co-$\mathcal{NP}$ ist, aber nicht in $\mathcal{P}$, so ist es in $\mathcal{NPI}$.

Vermutete Situation



Das Problem PRIMTEILER

Problem PRIMTEILER

Gegeben: Zahl $N \in \mathbb{N}$, Zahl $K \in \mathbb{N}$

Frage: Hat N einen Primteiler $P \leq K$?

- Ein Primteiler ist eine Primzahl P mit $N/P \in \mathbb{N}$.

Das Problem PRIMTEILER

Problem PRIMTEILER

Gegeben: Zahl $N \in \mathbb{N}$, Zahl $K \in \mathbb{N}$

Frage: Hat N einen Primteiler $P \leq K$?

- Ein Primteiler ist eine Primzahl P mit $N/P \in \mathbb{N}$.

Satz.

PRIMTEILER ist in $\mathcal{NP}$ und in co-$\mathcal{NP}$.

Beweis:

- Für eine gegebene Zahl P kann in polynomialer Zeit getestet werden, ob P Primzahl, $N/P \in \mathbb{N}$ und $P \leq K$.
- Also ist PRIMTEILER $\in \mathcal{NP}$.

Das Problem PRIMTEILER

Problem PRIMTEILER

Gegeben: Zahl $N \in \mathbb{N}$, Zahl $K \in \mathbb{N}$

Frage: Hat N einen Primteiler $P \leq K$?

- Ein Primteiler ist eine Primzahl P mit $N/P \in \mathbb{N}$.

Satz.

PRIMTEILER ist in $\mathcal{NP}$ und in co-$\mathcal{NP}$.

Beweis:

- Für gegebene Zahlen $P_1, \ldots, P_m$ (mit Vielfachheit) kann in polynomialer Zeit getestet werden, ob $P_1, \ldots, P_m$ Primzahlen, $N = P_1 \times \cdots \times P_m$ und $K < P_1, \ldots, P_m$.
- Also ist PRIMTEILER $\in$ co-$\mathcal{NP}$.

Vermutete Situation