Vorlesung 16
NP-Vollständigkeit ausgewählter Zahlprobleme

Wdh.: NP-Vollständigkeit

Definition (NP-vollständig)

Ein Problem L heißt NP-vollständig (engl. NP-complete), falls gilt

1. $L \in$ NP, und
2. L ist NP-schwer.

Die Klasse der NP-vollständigen Probleme wird mit NPC bezeichnet.

Satz (Cook und Levin)

SAT ist NP-vollständig.

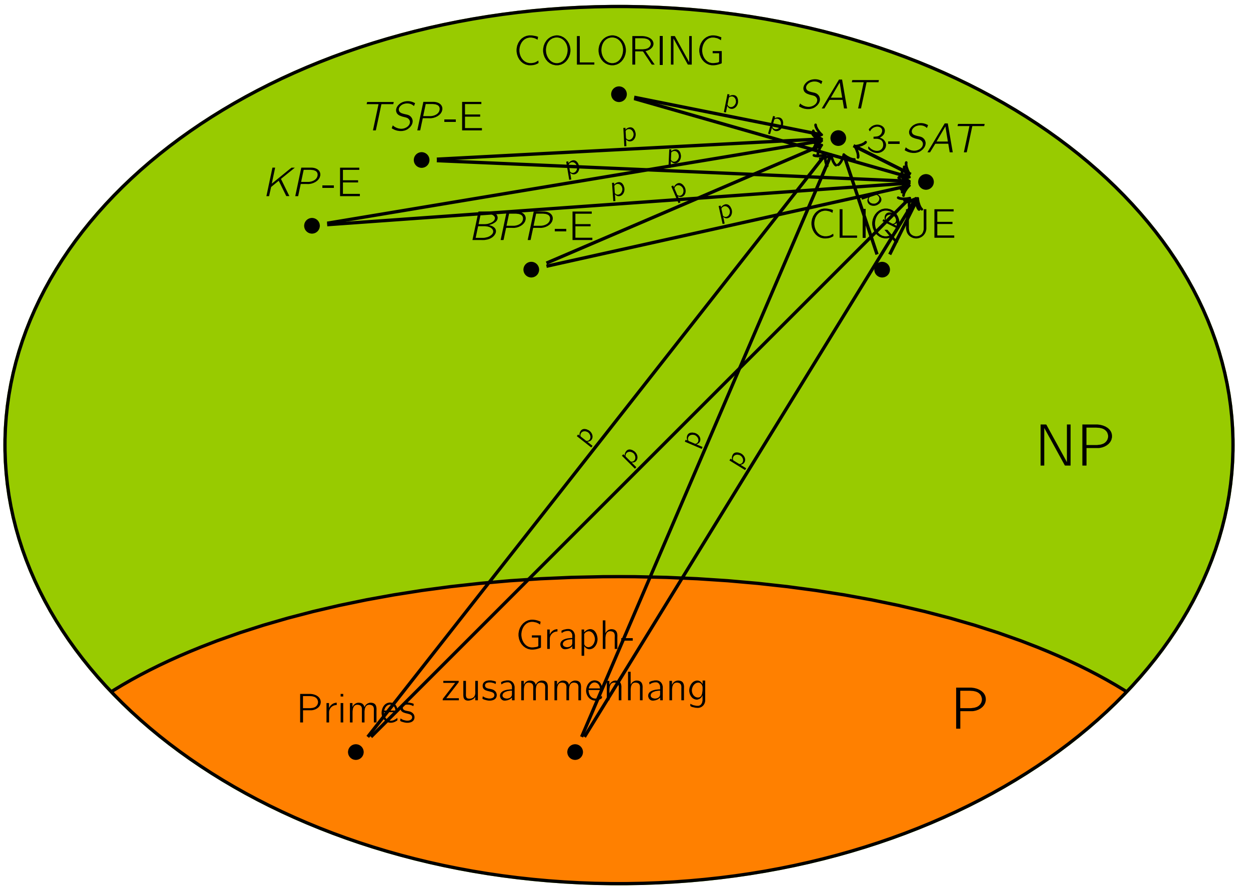
Lemma

3-SAT $\in$ NP und SAT $\leq_p$ 3-SAT.

Korollar

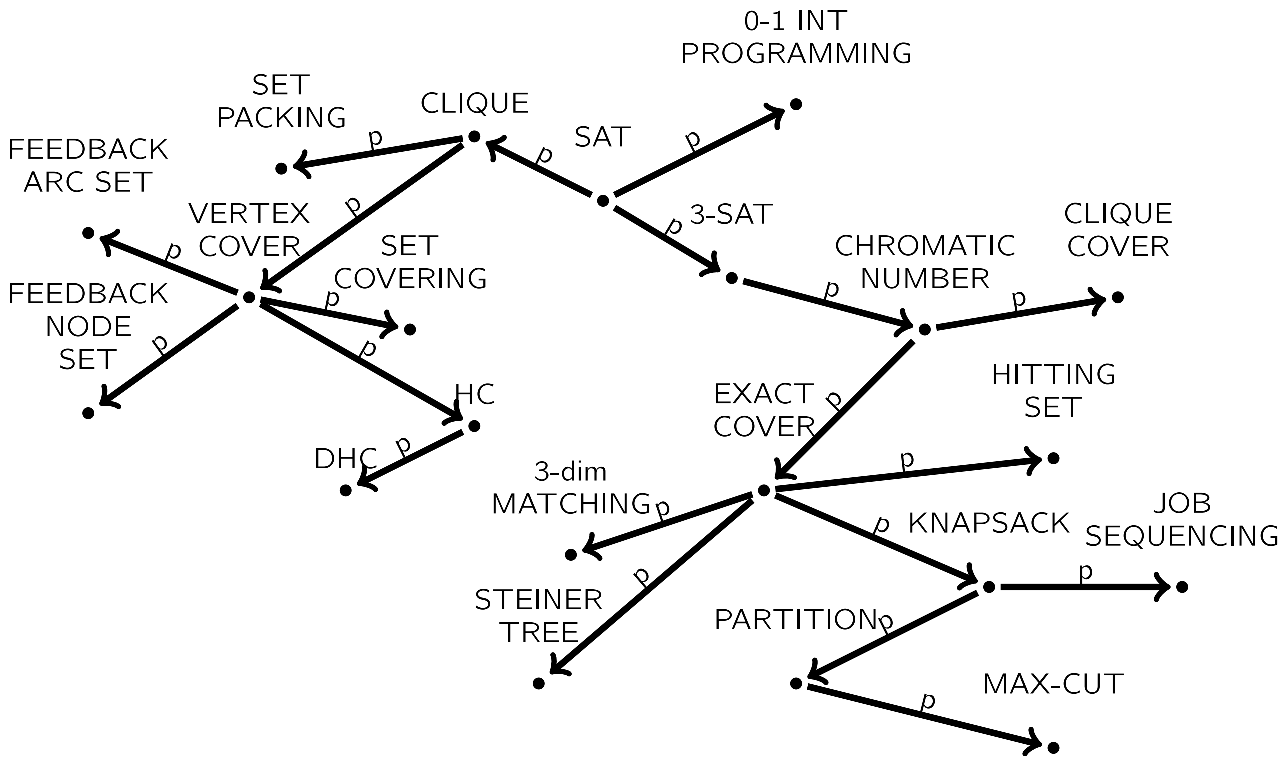
3-SAT ist NP*-vollständig.*

Die Komplexitätslandschaft



Warnung: Dieser Abbildung liegt die Annahme $\mathrm{P} \neq \mathrm{NP}$ zu Grunde.

Wdh.: Karps Liste mit 21 NP-vollständigen Problemen



Es gibt mittlerweile mehrere tausende Berechnungsprobleme verschiedenster Natur, deren NP-Vollständigkeit bekannt ist.

Wdh.: Kochrezept für NP-Vollständigkeitsbeweise

Wie beweist man, dass eine Sprache L NP-vollständig ist?

1. Man zeige $L \in \mathrm{NP}$.
2. Man wähle eine NP-vollständige Sprache L'.
3. Man entwerfe eine Funktion f, die Instanzen von L' auf Instanzen von L abbildet. (**Beschreibung der Reduktionsabbildung**)
4. Man zeige, dass f in polynomieller Zeit berechnet werden kann. (**Polynomialzeit**)
5. Man beweise, dass f eine Reduktion ist: Für $x \in \{0,1\}^*$ ist $x \in L'$ genau dann, wenn $f(x) \in L$. (**Korrektheit**)

Das SUBSET-SUM-Problem

Problem (SUBSET-SUM)

Eingabe: $a_1, \ldots, a_N \in \mathbb{N}$, $b \in \mathbb{N}$

Frage: Gibt es $K \subseteq \{1, \ldots, N\}$ *mit* $\sum_{i \in K} a_i = b$?

Das SUBSET-SUM-Problem ist in NP enthalten, denn die Lösung K kann als Zertifikat verwendet werden, das in polynomieller Zeit verifiziert werden kann.

NP-Vollständigkeit des SUBSET-SUM-Problems

Satz

SUBSET-SUM ist NP-vollständig.

Beweis:
1.) SUBSET-SUM $\in$ NP: ✓

2.) Um die NP-Schwere des Problems nachzuweisen, beschreiben wir eine Polynomialzeitreduktion von 3-*SAT* auf SUBSET-SUM.

3.) (Beschreibung der Reduktion) Gegeben sei eine Formel φ in 3-KNF. Diese Formel bestehe aus M Klauseln $c_1, \ldots, c_M$ über N Variablen $x_1, \ldots, x_N$.

Für $i \in \{1, \ldots, N\}$ sei

$$\begin{aligned} S(i) &= \{j \in \{1, \ldots, M\} \mid \text{Klausel } c_j \text{ enthält Literal } x_i\} \ , \\ S'(i) &= \{j \in \{1, \ldots, M\} \mid \text{Klausel } c_j \text{ enthält Literal } \bar{x}_i\} \ . \end{aligned}$$

Reduktion 3-SAT $\leq_p$ SUBSET-SUM

Aus der Formel φ in 3-KNF erzeugen wir verschiedene Dezimalzahlen mit jeweils $N + M$ Ziffern.

Die k-te Ziffer einer Zahl a bezeichnen wir dabei mit $a(k)$.

Für jede boolesche Variable x_i, $i \in \{1, \ldots, N\}$ erzeugen wir zwei Zahlen a_i und a'_i, deren Ziffern wie folgt definiert sind:

$$\begin{aligned} a_i(i) = 1 \quad &\text{und} \quad \forall j \in S(i) : a_i(N+j) = 1 \ , \\ a'_i(i) = 1 \quad &\text{und} \quad \forall j \in S'(i) : a'_i(N+j) = 1 \ . \end{aligned}$$

Alle anderen Ziffern setzen wir auf den Wert 0.

Diese Zahlen bezeichnen wir als a-Zahlen.

Reduktion 3-SAT $\leq_p$ SUBSET-SUM

Beispiel:

Gegeben sei die Formel

$$(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \bar{x_3} \vee \bar{x_4}) \ .$$

Aus dieser Formel werden folgende a-Zahlen erzeugt:

$$\begin{aligned}
a_1 &= 100010 \\
a_1' &= 100000 \\
a_2 &= 010011 \\
a_2' &= 010000 \\
a_3 &= 001010 \\
a_3' &= 001001 \\
a_4 &= 000100 \\
a_4' &= 000101
\end{aligned}$$

Reduktion 3-SAT $\leq_p$ SUBSET-SUM

Zusätzlich erzeugen wir zwei sogenannte h-Zahlen h_j und h_j' für jede Klausel j. Diese Zahlen haben nur an der Ziffernposition $N + j$ eine 1, und alle anderen Ziffern sind 0.

Den Summenwert b definieren wir folgendermaßen: Wir setzen $b(k) = 1$ für $1 \leq k \leq N$ und $b(k) = 3$ für $N + 1 \leq k \leq N + M$.

Fortsetzung des Beispiels $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \bar{x_3} \vee \bar{x_4})$:

Die h-Zahlen und der Summenwert lauten

$$\begin{aligned}
h_1 &= 000010 \\
h_1' &= 000010 \\
h_2 &= 000001 \\
h_2' &= 000001 \\
b &= 111133
\end{aligned}$$

Reduktion 3-SAT $\leq_p$ SUBSET-SUM

Für eine Formel aus N Variablen und M Klauseln könnten sich beispielsweise die folgenden Zahlen ergeben:

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	1	0	0	$\cdots$	0	1	0	$\cdots$	$\cdots$
a'_1	1	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	1	0	$\cdots$	0	0	1	$\cdots$	$\cdots$
a'_2	0	1	0	$\cdots$	0	1	0	$\cdots$	$\cdots$
a_3	0	0	1	$\cdots$	0	1	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_N	0	0	0	$\cdots$	1	0	0	$\cdots$	$\cdots$
a'_N	0	0	0	$\cdots$	1	0	1	$\cdots$	$\cdots$
h_1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
h'_1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
h_M	0	0	0	$\cdots$	0	0	0	$\cdots$	1
h'_M	0	0	0	$\cdots$	0	0	0	$\cdots$	1
b	1	1	1	$\cdots$	1	3	3	$\cdots$	3

Reduktion 3-SAT $\leq_p$ SUBSET-SUM

4.) (Polynomialzeit) Die Eingabezahlen zu SUBSET-SUM können in polynomieller Zeit erzeugt werden. (Die Zahlenwerte können aber natürlich exponentiell groß sein).

5.) (Korrektheit)

Beobachtung

Bei der Addition einer beliebigen Teilmenge der a-Zahlen und der h-Zahlen gibt es keinen Additionsübertrag von Stelle zu Stelle, weil höchstens fünf Ziffern pro Spalte den Wert 1 haben.

Anmerkung: Die Beobachtung beruht darauf, dass wir mit Dezimalziffern, d.h. zur Basis 10, rechnen. De facto wäre es auch ausreichend, wenn wir zur Basis 6 rechnen würden.

Reduktion 3-SAT $\leq_p$ SUBSET-SUM: *Korrektheit*

Zu zeigen: φ erfüllbar $\Rightarrow$ es gibt eine Teilmenge der a- und h-Zahlen, deren Summe b ist

Angenommen, es gibt eine erfüllende Belegung x^* für φ.

- Falls $x_i^* = 1$, so *wähle* a_i aus, ansonsten wähle a_i'.
- Sei A die Summe der ausgewählten a-Zahlen.
- Da für jedes $i \in \{1, \ldots, N\}$ entweder a_i oder a_i' ausgewählt wurde, gilt $A(i) = 1$.
- Zudem gilt $A(N+j) \in \{1, 2, 3\}$ für $1 \leq j \leq M$, weil in jeder Klausel mindestens ein und höchstens drei Literale erfüllt werden.
- Falls $A(N+j) < 3$, so können wir zusätzlich h_j oder h_j und h_j' auswählen, um exakt den geforderten Wert 3 an Ziffernposition $N+j$ der Summe zu erhalten.

Also gibt es eine Teilmenge mit Summenwert b.

Reduktion 3-SAT $\leq_p$ SUBSET-SUM: *Korrektheit*

Zu zeigen: es gibt eine Teilmenge der a- und h-Zahlen, deren Summe b ist $\Rightarrow$ φ erfüllbar

Sei A die Summe einer Teilmenge K_A der a-Zahlen und H die Summe einer Teilmenge der h-Zahlen, so dass gilt $A + H = b$.

In K_A ist für jedes $i \in \{1, \ldots, N\}$ genau eine der beiden a-Zahlen a_i oder a_i' enthalten, denn ansonsten wäre $A(i) \neq 1$.

Setze $x_i = 1$, falls $a_i \in K_A$, und $x_i = 0$, sonst.

Zu zeigen: x ist eine erfüllende Belegung für φ

- Es gilt $A(N+j) \geq 1$ für $1 \leq j \leq M$, denn ansonsten wäre $A(N+j) + H(N+j) \leq A(N+j) + 2 < 3$.
- Dadurch ist sichergestellt, dass in jeder Klausel mindestens eines der Literale den Wert 1 hat, so dass φ erfüllt ist.

Damit ist die Korrektheit der Reduktion nachgewiesen. $\square$

NP-Vollständigkeit von PARTITION

Problem (PARTITION)

Eingabe: $a_1, \ldots, a_N \in \mathbb{N}$

Frage: Gibt es $K \subseteq \{1, \ldots, N\}$ *mit* $\sum_{i \in K} a_i = \sum_{i \in \{1,\ldots,N\} \setminus K} a_i$*?*

PARTITION ist ein Spezialfall von SUBSET-SUM, da die gestellte Frage äquivalent zur der Frage ist, ob es eine Teilmenge K mit Summenwert $b = \frac{1}{2} \sum_{i=1}^{N} a_i$ gibt.

NP-Vollständigkeit von PARTITION

Satz

PARTITION ist NP*-vollständig.*

Beweis:

1.) PARTITION ist offensichtlich in NP, weil es als Spezialfall von SUBSET-SUM aufgefasst werden kann.

2.) Um zu zeigen, dass PARTITION NP-schwer ist, zeigen wir SUBSET-SUM $\leq_p$ PARTITION.

Reduktion von SUBSET-SUM auf PARTITION

3.) Die Eingabe von SUBSET-SUM sei $a_1, \ldots, a_N \in \mathbb{N}$ und $b \in \mathbb{N}$.

Es sei $A = \sum_{i=1}^{N} a_i$.

Wir bilden diese Eingabe für SUBSET-SUM auf eine Eingabe für PARTITION ab, die aus $N + 2$ Zahlen $a'_1, \ldots, a'_{N+2}$ bestehe.

Dazu setzen wir

- $a'_i = a_i$ für $1 \leq i \leq N$,
- $a'_{N+1} = 2A - b$, und
- $a'_{N+2} = A + b$.

In der Summe ergeben diese $N + 2$ Zahlen den Wert $4A$.

Diese Zahlen bilden genau dann eine Ja-Instanz von PARTITION, wenn es eine Teilmenge der Zahlen $a'_1, \ldots, a'_{N+2}$ mit Summenwert $2A$ gibt.

Reduktion von SUBSET-SUM auf PARTITION

4.) Die Reduktion ist in polynomieller Zeit berechenbar.

5.) Wir zeigen: Es existiert eine Lösung für PARTITION $\Rightarrow$ es existiert eine Lösung für SUBSET-SUM

- Wenn es eine geeignete Aufteilung der Eingabezahlen für PARTITION gibt, so können a'_{N+1} und a'_{N+2} dabei nicht in derselben Teilmenge sein, denn $a'_{N+1} + a'_{N+2} = 3A$.
- Deshalb ergibt sich auch eine Lösung für SUBSET-SUM, denn diejenigen Zahlen aus $a'_1, \ldots, a'_N$, die sich in derselben Teilmenge wie a'_{N+1} befinden, summieren sich auf zu $2A - a'_{N+1} = b$.

Reduktion von SUBSET-SUM auf PARTITION

Wir zeigen: Es existiert eine Lösung für SUBSET-SUM $\Rightarrow$ es existiert eine Lösung für PARTITION

- Wenn es eine Teilmenge der Zahlen $a_1, \ldots, a_N$ mit Summenwert b gibt, so gibt es auch eine Teilmenge der Zahlen $a'_1, \ldots, a'_N$ mit diesem Summenwert.
- Wir können die Zahl $a'_{N+1} = 2A - b$ zu dieser Teilmenge hinzufügen und erhalten dadurch eine Teilmenge mit Summenwert $2A$. $\square$

Bin Packing ist NP-vollständig

Problem (Bin Packing Problem – BPP)

Eingabe: $b \in \mathbb{N}$, $w_1, \ldots, w_N \in \{1, \ldots, b\}$

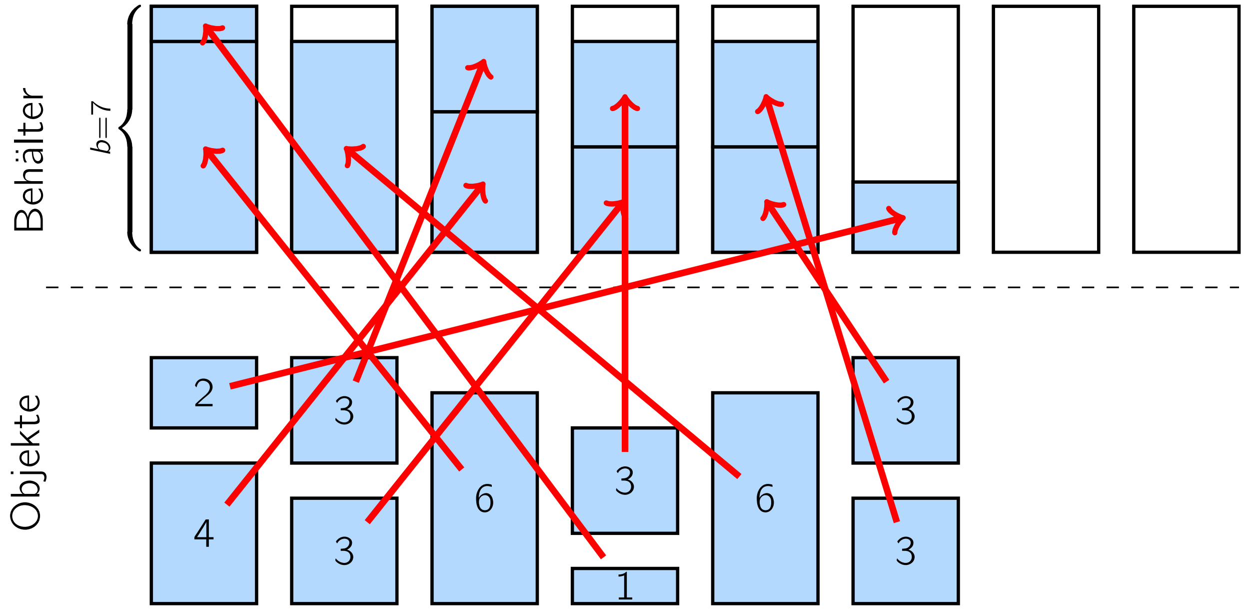
zulässige Lösungen: $k \in \mathbb{N}$ *und Funktion* $f \colon \{1, \ldots, N\} \to \{1, \ldots, k\}$,

$$\textit{so dass } \forall i \in \{1, \ldots, k\} \colon \sum_{j \in f^{-1}(i)} w_j \le b$$

Zielfunktion: *Minimiere* k *(= Anzahl Behälter)*

Entscheidungsvariante (BPP-E): Zusätzlich ist $k \in \mathbb{N}$ gegeben. Passen die Objekte in k Behälter?

Bin Packing Problem – Beispiel



Eine Lösung die $k = 6$ Behälter verwendet

Bin Packing ist NP-vollständig

Satz

BPP-E ist NP*-vollständig.*

Beweis:

1.) BPP-E $\in$ NP haben wir bereits gezeigt.

(2.-5.) Die NP-Schwere ergibt sich durch eine triviale Reduktion von PARTITION:

$$\text{Setze } k = 2,\ w_i = a_i \text{ für } 1 \le i \le N \text{ und } b = \left\lfloor \tfrac{1}{2} \textstyle\sum_{i=1}^{N} w_i \right\rfloor.$$

□

Das Rucksackproblem ist NP-vollständig

Problem (Entscheidungsvariante des Rucksackproblems – KP-E)

Eingabe: $B, P \in \mathbb{N}$, $w_1, \ldots, w_N \in \{1, \ldots, B\}$, $p_1, \ldots, p_N \in \mathbb{N}$

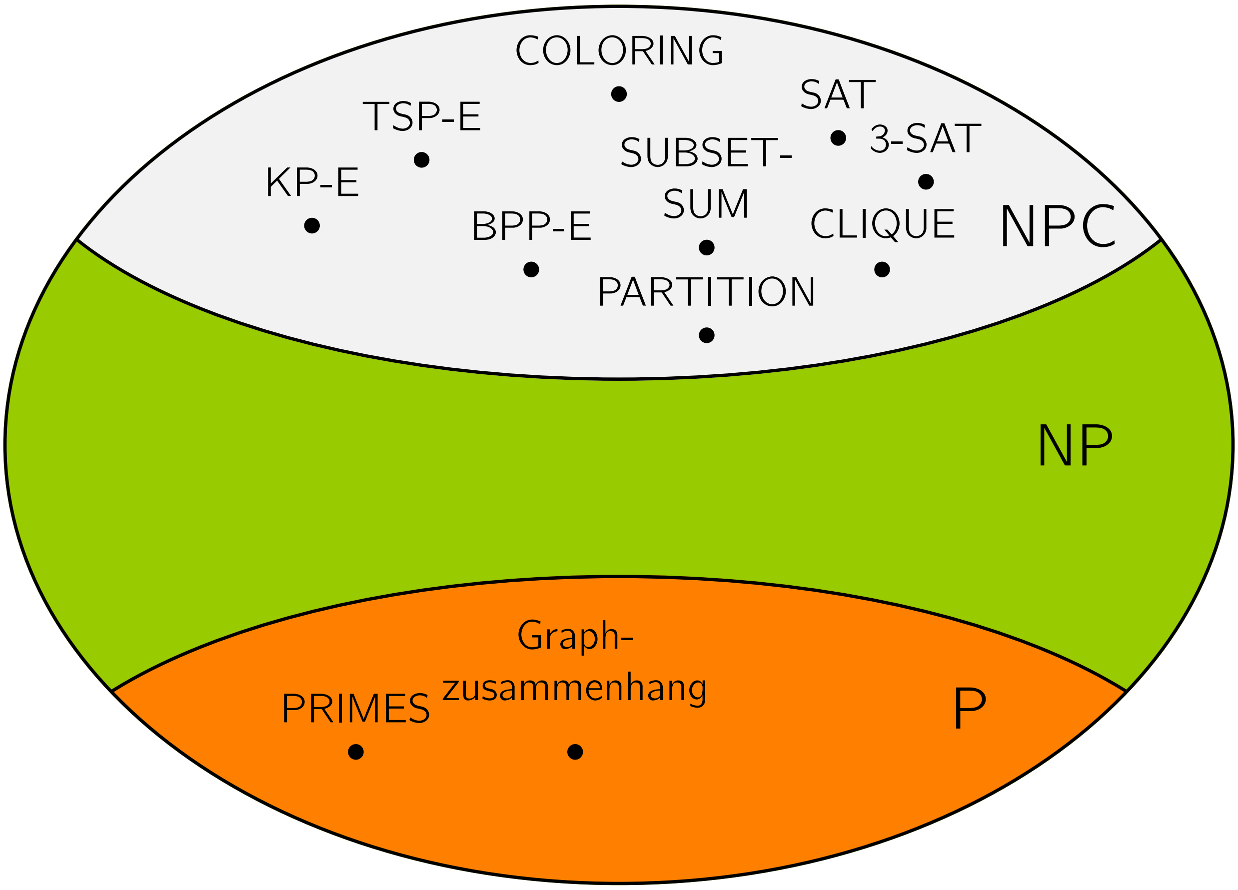
Frage: *Gibt es* $K \subseteq \{1, \ldots, N\}$ *mit* $\sum_{i \in K} w_i \leq B$ *und* $\sum_{i \in K} p_i \geq P$?

Korollar

KP-E ist NP*-vollständig.*

Beweis durch einfache Reduktion von SUBSET-SUM (Wie?)

Die Komplexitätslandschaft



Warnung: Dieser Abbildung liegt die Annahme P $\neq$ NP zu Grunde.