Vorlesung 17
NP-Vollständigkeit ausgewählter Graphprobleme

Wdh.: NP-Vollständigkeit von Zahlproblemen

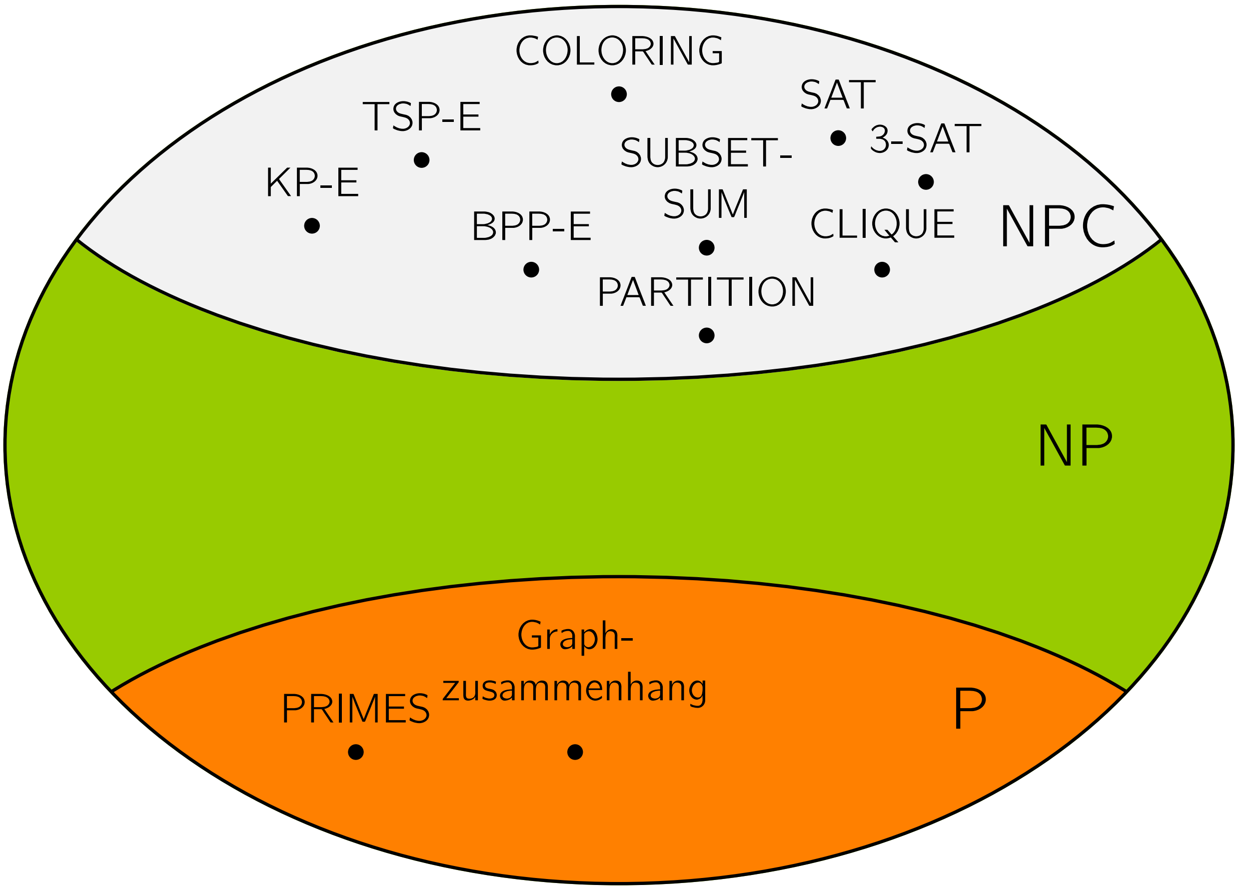
Satz
SUBSET-SUM ist NP-vollständig.

Satz
PARTITION ist NP-vollständig.

Satz
BPP-E ist NP-vollständig.

Satz
KP-E ist NP-vollständig.

Wdh.: Die Komplexitätslandschaft



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

Wdh.: Kochrezept für NP-Vollständigkeitsbeweise

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

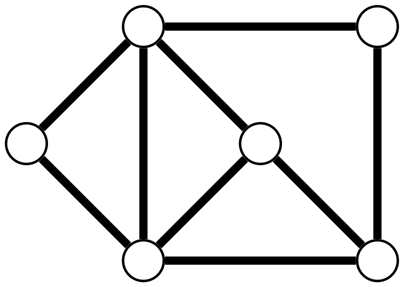
1. Man zeige $L \in \mathrm{NP}$.
2. Man wähle eine NP-vollständige Sprache L'.
3. Man entwerfe eine Funktion f, die Instanzen von L' auf Instanzen von L abbildet. (**Beschreibung der Reduktionsabbildung**)
4. Man zeige, dass f in polynomieller Zeit berechnet werden kann. (**Polynomialzeit**)
5. Man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

NP-Vollständigkeit von CLIQUE

Wie erinnern uns an das Cliquenproblem.

Problem (CLIQUE)

Eingabe: *Graph* $G = (V, E)$, $k \in \{1, \ldots, |V|\}$
Frage: *Hat* G *eine* k*-Clique?*



$k = 4$

Der Graph hat keine Clique der Größe 4.

Satz

CLIQUE ist NP*-vollständig.*

Beweis der NP-Vollständigkeit von CLIQUE

1.) Da wir schon wissen, dass das Cliquenproblem in NP ist, müssen wir zum Nachweis der NP-Vollständigkeit nur noch die NP-Schwere nachweisen.

2.) Dazu zeigen wir SAT $\leq_p$ CLIQUE.

3.) Wir beschreiben eine polynomiell berechenbare Funktion f, die eine KNF-Formel φ in einen Graphen $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$ transformiert, so dass gilt:

$$\varphi \text{ ist erfüllbar } \Leftrightarrow \; G \text{ hat eine } k\text{-Clique} \; .$$

Beweis der NP-Vollständigkeit von CLIQUE

Beschreibung der Funktion f:

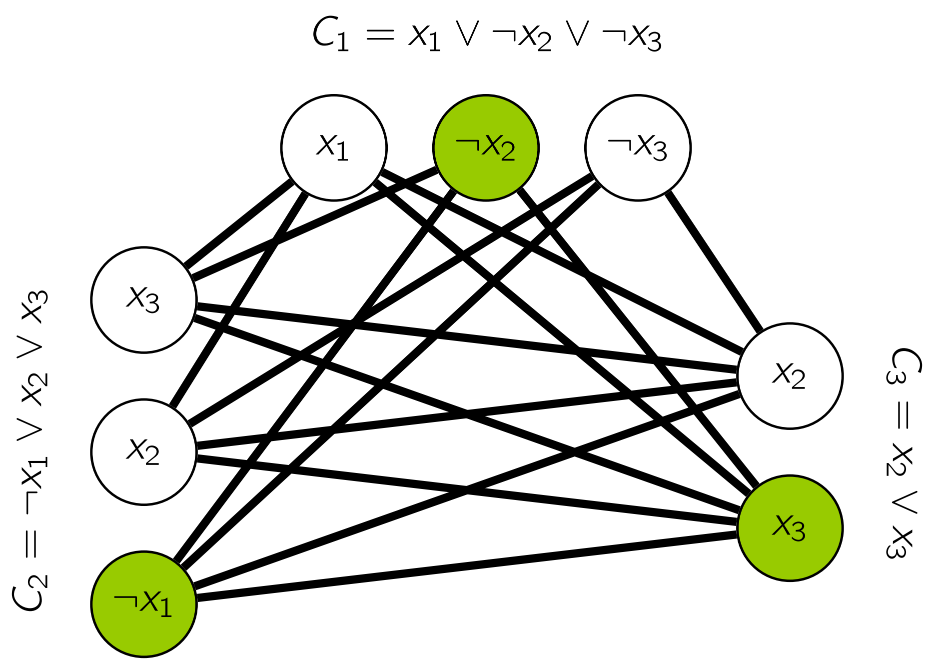
- Seien $C_1, \ldots, C_m$ die Klauseln von φ.
- Sei k_i die Anzahl an Literalen in Klausel C_i.
- Seien $\ell_{i,1}, \ldots, \ell_{i,k_i}$ die Literale in Klausel C_i.
- Identifiziere Literale und Knoten, d.h., setze

$$V = \{\ell_{i,j} \mid 1 \leq i \leq m,\ 1 \leq j \leq k_i\} .$$

- Zwei Knoten werden mit einer Kante verbunden, wenn
  - sie aus verschiedenen Klauseln stammen und
  - ihre Literale nicht Negationen voneinander sind.
- Setze $k = m$.

Beweis der NP-Vollständigkeit von CLIQUE – Illustration

Beispiel: $\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3)$



Erfüllende Belegung: $x_1 = 0$, $x_2 = 0$, $x_3 = 1$.

Beweis der NP-Vollständigkeit von CLIQUE

5.) Korrektheit der Transformation:

Zu zeigen: φ erfüllbar $\Rightarrow$ G hat m-Clique

Betrachte eine beliebige erfüllende Belegung. Pro Klausel wähle ein erfülltes Literal beliebig aus. Sei U die Menge dieser Literale. Wir behaupten, U bildet eine m-Clique.

Begründung: Gemäß der Definition ist $|U| = m$. Seien ℓ und ℓ' zwei unterschiedliche Literale aus U. Wir müssen zeigen, dass ℓ und ℓ' durch eine Kante verbunden sind:

- Nach Konstruktion stammen ℓ und ℓ' aus verschiedenen Klauseln.
- Da ℓ und ℓ' gleichzeitig erfüllt sind, sind sie nicht Negationen voneinander.

Also gibt es eine Kante zwischen ℓ und ℓ'.

Beweis der NP-Vollständigkeit von CLIQUE

Zu zeigen: G hat m-Clique $\Rightarrow$ φ erfüllbar

- Sei U eine m-Clique in G.
- Dann gehören die Literale in U zu verschiedenen Klauseln.
- U enthält somit genau ein Literal pro Klausel in φ.
- Diese Literale können alle gleichzeitig erfüllt werden, da kein Literal positiv und negiert vorkommt.
- Also ist φ erfüllbar.

4.) Die Funktion f ist in Polynomialzeit berechenbar. $\square$

Hamiltonkreisprobleme

Problem (Hamiltonkreis – Hamiltonian Circuit – HC)

Eingabe: Graph $G = (V, E)$

Frage: Gibt es einen Hamiltonkreis in G*?*

Problem (Gerichteter Hamiltonkreis – Directed HC – DHC)

Eingabe: gerichteter Graph $G = (V, E)$

Frage: Gibt es einen Hamiltonkreis in G*?*

HC $\leq_p$ DHC

Lemma

HC $\leq_p$ *DHC.*

Beweis:

Reduktion: Für HC liege ein ungerichteter Graph G vor. Wir transformieren G in einen gerichteten Graphen G', indem wir jede ungerichtete Kante durch zwei entgegengesetzte gerichtete Kanten ersetzen.

Polynomialzeit: Diese lokale Ersetzung ist offensichtlich in polynomieller Zeit möglich.

Korrektheit: Nach Konstruktion hat G genau dann einen Hamiltonkreis, wenn auch G' einen Hamiltonkreis hat. □

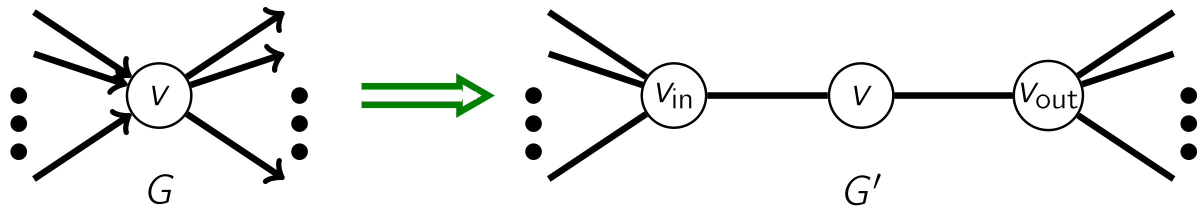
DHC $\leq_p$ HC

Lemma

DHC $\leq_p$ HC.

Beweis:

Reduktion: Gegeben sei nun ein gerichteter Graph $G = (V, E)$. Aus G konstruieren wir wieder mittels lokaler Ersetzung einen ungerichteten Graphen G':



Interpretation: v_{in} ist der Eingangsknoten für v, und v_{out} ist der Ausgangsknoten für v.

DHC $\leq_p$ HC – Fortsetzung Beweis

Korrektheit:
Wir müssen zeigen: G hat genau dann einen Hamiltonkreis, wenn auch G' einen Hamiltonkreis hat.

Jede Rundreise in G kann offensichtlich in eine Rundreise in G' transformiert werden.

Aber wie sieht es mit der Umkehrrichtung aus?

- Eine Rundreise in G', die einen Knoten v direkt nach v_{in} besucht, besucht jeden Knoten v' direkt nach v'_{in}.
- Eine Rundreise in G', die einen Knoten v direkt nach v_{out} besucht, besucht jeden Knoten v' direkt nach v'_{out}. Aber diese Rundreise können wir rückwärts ablaufen.

Also kann auch jeder Hamiltonkreis in G' in einen Hamiltonkreis in G transformiert werden. $\square$

NP-Vollständigkeit von HC und DHC

Satz

HC und DHC sind NP-*vollständig.*

Beweis:

Beide Probleme sind offensichtlich in NP, da man die Kodierung eines Hamiltonkreises in polynomieller Zeit auf ihre Korrektheit überprüfen kann.

Da HC und DHC gegenseitig aufeinander polynomiell reduzierbar sind, genügt es die NP-Schwere eines der beiden Probleme nachzuweisen.

Wir zeigen die NP-Schwere von DHC durch eine polynomielle Reduktion von SAT auf DHC.

NP-Vollständigkeit von HC und DHC – Fortsetzung Beweis

(Reduktion:)

Wir präsentieren eine polynomiell berechenbare Funktion f, die eine KNF-Formel φ mit Variablen

$$x_1, \ldots, x_N$$

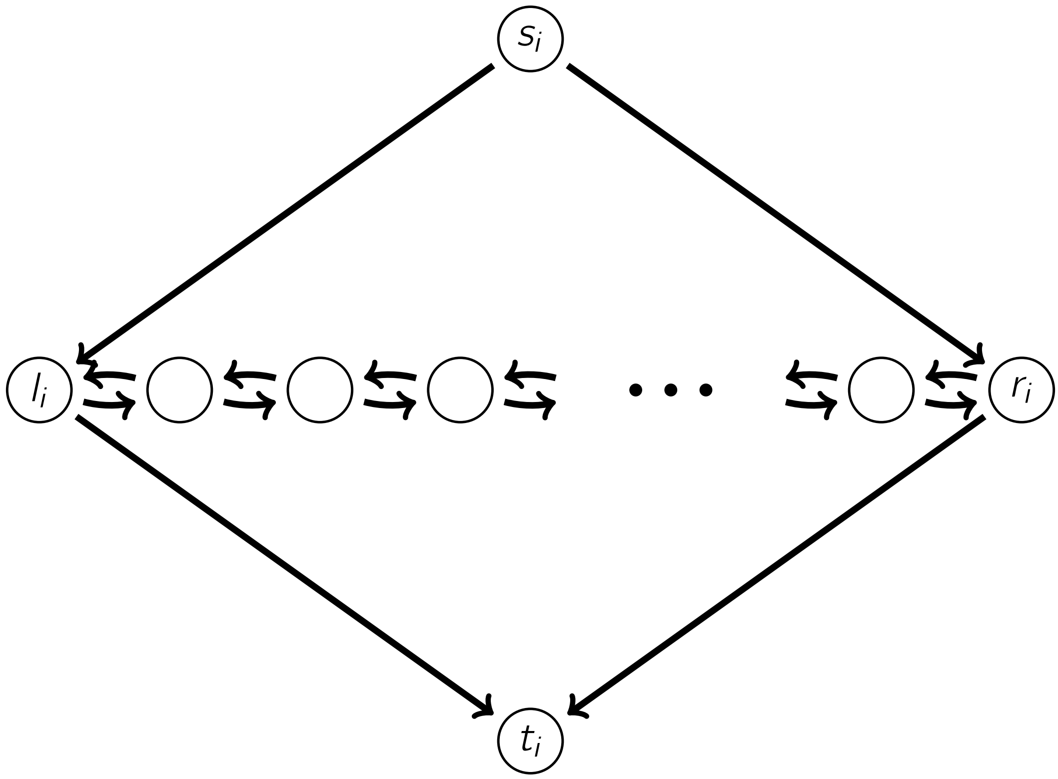
und Klauseln

$$c_1, \ldots, c_M$$

auf einen gerichteten Graphen $G = (V, E)$ abbildet, der genau dann einen Hamiltonkreis hat, wenn φ erfüllbar ist.

NP-Vollständigkeit von HC und DHC – Beweis

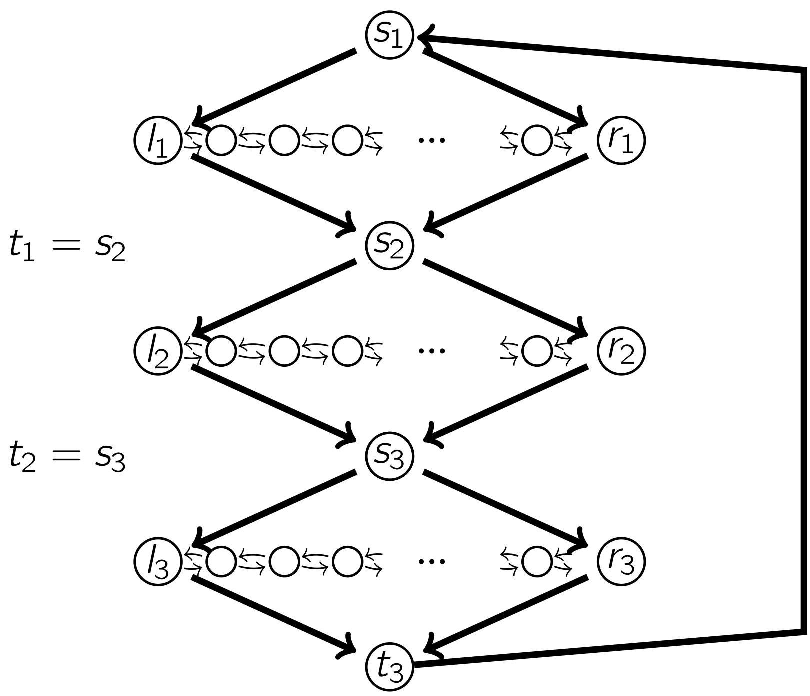
Für jede Variable x_i enthalte der Graph G die folgende Struktur G_i.



Diese Struktur heißt Diamantengadget.

NP-Vollständigkeit von HC und DHC – Fortsetzung Beweis

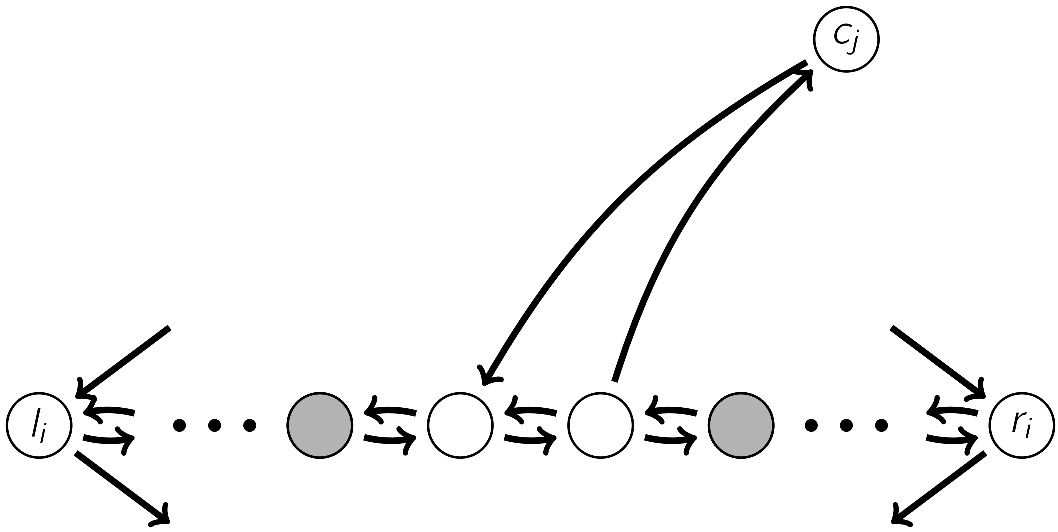
Diese N Gadgets werden miteinander verbunden, indem wir die Knoten t_i und s_{i+1} (für $1 \leq i \leq N-1$) sowie t_N und s_1 miteinander identifizieren.



NP-Vollständigkeit von HC und DHC – Fortsetzung
Beweis

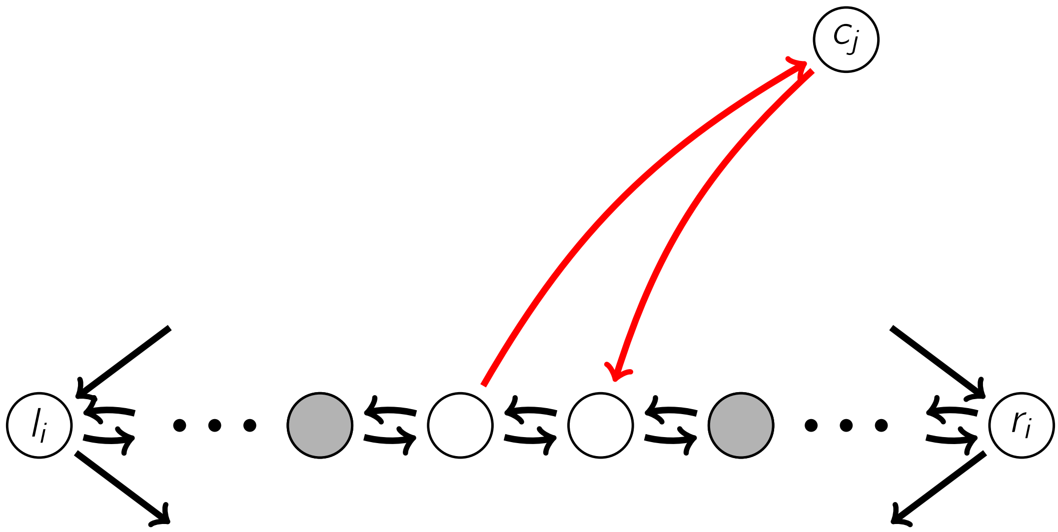
Jetzt fügen wir einen weiteren Knoten für jede Klausel c_j ein.

Falls das Literal x_i in Klausel c_j enthalten ist, so verbinden wir das Gadget G_i wie folgt mit dem Klauselknoten c_j:



NP-Vollständigkeit von HC und DHC – Fortsetzung
Beweis

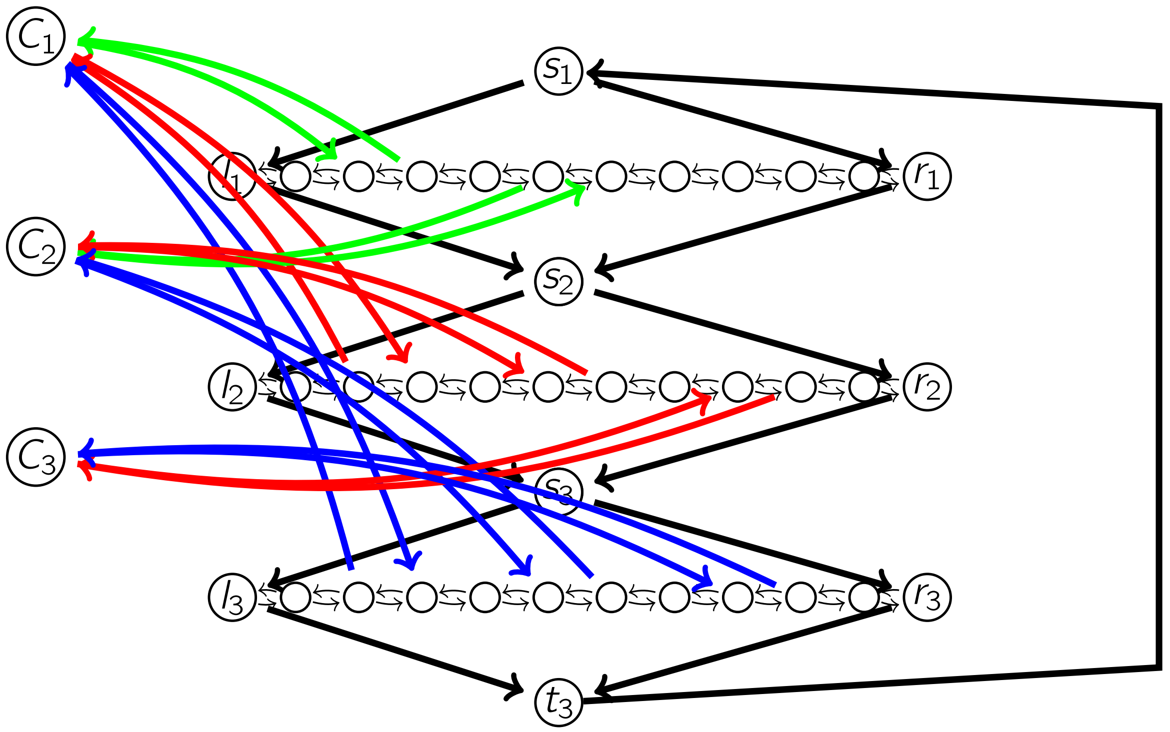
Falls das Literal $\bar{x}_i$ in Klausel c_j enthalten ist, so verbinden wir das Gadget G_i wie folgt mit dem Klauselknoten c_j:



Ist es nach Hinzunahme der Klauselknoten möglich, dass eine Rundreise zwischen den Gadgets hin- und herspringt, statt sie in der vorgesehenen Reihenfolge zu besuchen? – Nein. (Warum?)

NP-Vollständigkeit von HC und DHC – Illustration

$$\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3)$$



NP-Vollständigkeit von HC und DHC – Fortsetzung Beweis

Korrektheit:

Zu zeigen: G hat einen Hamiltonkreis $\Rightarrow$ φ ist erfüllbar

- Wird ein Klauselknoten c_j aus einem Gadget G_i heraus von links nach rechts durchlaufen, so muss gemäß unserer Konstruktion die Klausel c_j das Literal x_i enthalten.
- Also wird diese Klausel durch die mit der Laufrichtung von links nach rechts assoziierten Belegung $x_i = 1$ erfüllt.
- Bei einer Laufrichtung von rechts nach links, die mit der Belegung $x_i = 0$ assoziiert ist, wird die Klausel ebenso erfüllt, weil sie in diesem Fall das Literal $\bar{x}_i$ enthält.

Also erfüllt die mit der Rundreise assoziierte Belegung alle Klauseln.

NP-Vollständigkeit von HC und DHC – Fortsetzung Beweis

Zu zeigen: φ ist erfüllbar $\Rightarrow$ G hat einen Hamiltonkreis

- Eine Belegung beschreibt, in welcher Richtung die Gadgets $G_1, \ldots, G_N$ jeweils durchlaufen werden.
- Klauselknoten c_j können wir in die Rundreise einbauen, indem wir eine der Variablen x_i auswählen, die c_j erfüllt, und c_j vom Gadget G_i aus besuchen.
- Sollte c_j für $x_i = 1$ erfüllt sein, so ist x_i positiv in c_j enthalten und somit ist ein Besuch von c_j beim Durchlaufen des Gadgets G_i von links nach rechts möglich.
- Sollte c_j hingegen für $x_i = 0$ erfüllt sein, so ist die Variable negiert in der Klausel enthalten und der Besuch von c_j kann beim Durchlaufen des Gadgets G_i von rechts nach links erfolgen.

Also können alle Klauselknoten in die Rundreise eingebunden werden. $\square$

NP-Vollständigkeit von TSP

Satz

TSP-E ist NP*-schwer.*

Beweis: Wir zeigen, dass TSP sogar dann NP-schwer ist, wenn nur die Kantengewichte 1 und 2 verwendet werden.

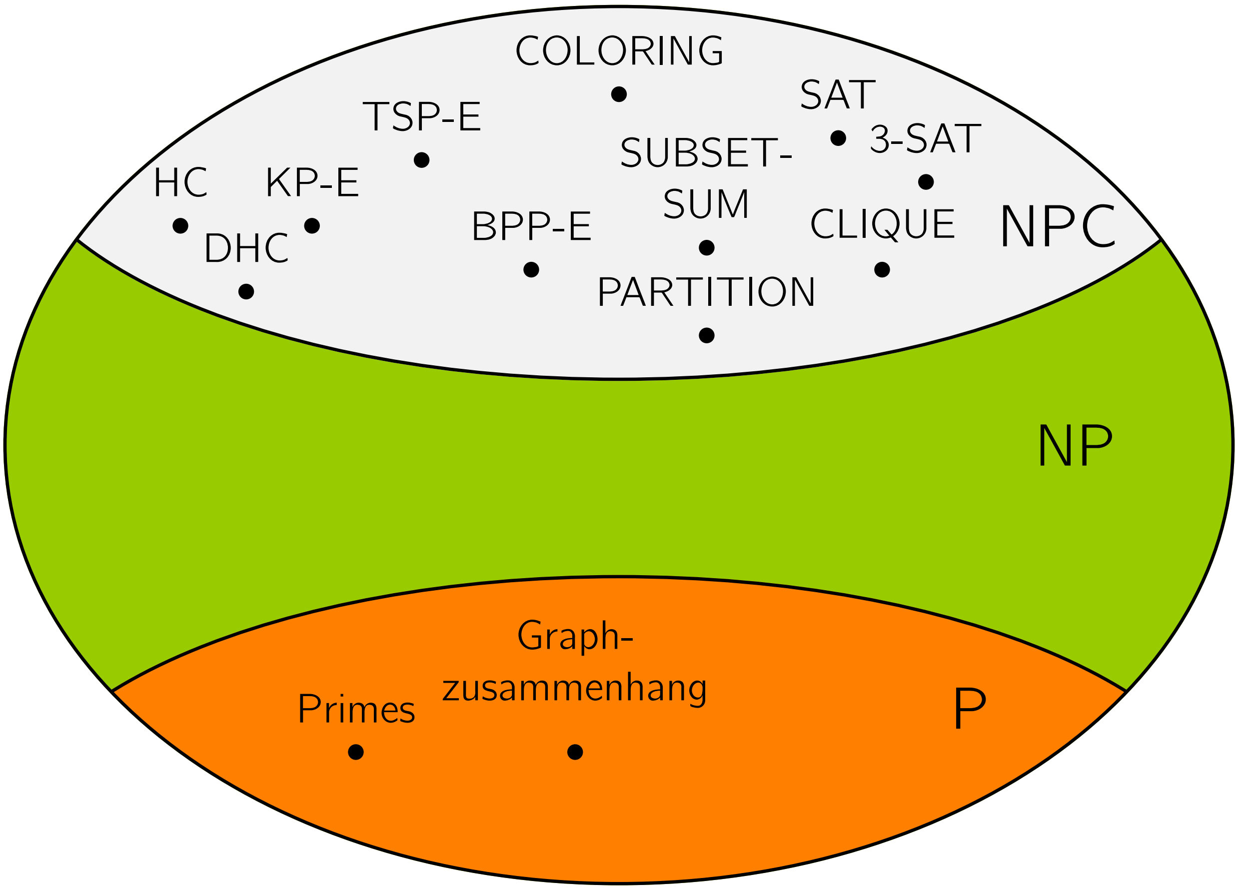
Diese eingeschränkte Variante von TSP nennen wir $\{1,2\}$-TSP. Wir zeigen HC $\leq_p$ $\{1,2\}$-TSP-E.

Aus dem Graphen $G = (V, E)$ für HC erzeugen wir einen vollständigen Graphen $G' = (V, E')$ mit Kosten

$$c(u, v) = \begin{cases} 1 & \text{falls } \{u, v\} \in E \\ 2 & \text{falls } \{u, v\} \notin E \end{cases}$$

G hat genau dann einen Hamiltonkreis, wenn G' eine TSP-Tour der Länge höchstens n hat. $\square$

Die Komplexitätslandschaft



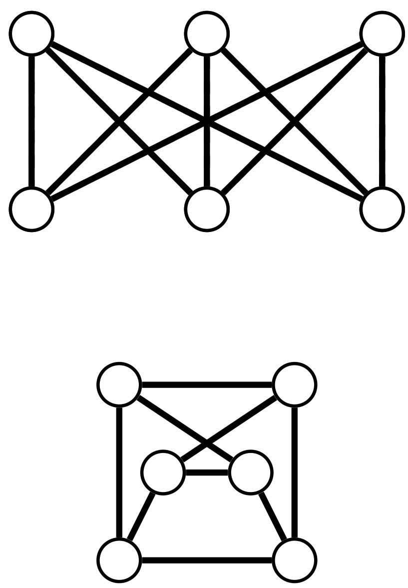
Warnung: Dieser Abbildung liegt die Annahme $P \neq NP$ zu Grunde.

Das Graphisomorphieproblem GI

Zwei Graphen G_1, G_2 sind isomorph, wenn es eine Bijektion von den Knoten von G_1 auf die Knoten von G_2 gibt, die Adjazenz und Nicht-Adjazenz erhält.

Eine solche Bijektion heißt Isomorphismus.

isomorphe Graphen



Problem (Graphisomorphieproblem GI)

Eingabe: *Zwei Graphen* G_1, G_2
Frage: *Gibt es einen Isomorphismus von* G_1 *nach* G_2*?*

Komplexität des Graphisomorphieproblems

Was ist die Komplexität des Graphisomorphieproblems?

- Es ist $GI \in \text{NP}$, da man einen Isomorphismus als Zertifikat verwenden kann.
- Damit gilt $GI \in \text{EXPTIME}$, also ist GI insbesondere berechenbar.

Gibt es einen effizienten Algorithmus für das Graphisomorphieproblem?

Folgende Fragen sind derzeit noch unbekannt:

- $GI \in P$?
- Ist GI NP-vollständig?
- $GI \in \text{co-NP}$?

NP-intermediate

Definition (NP-intermediate)

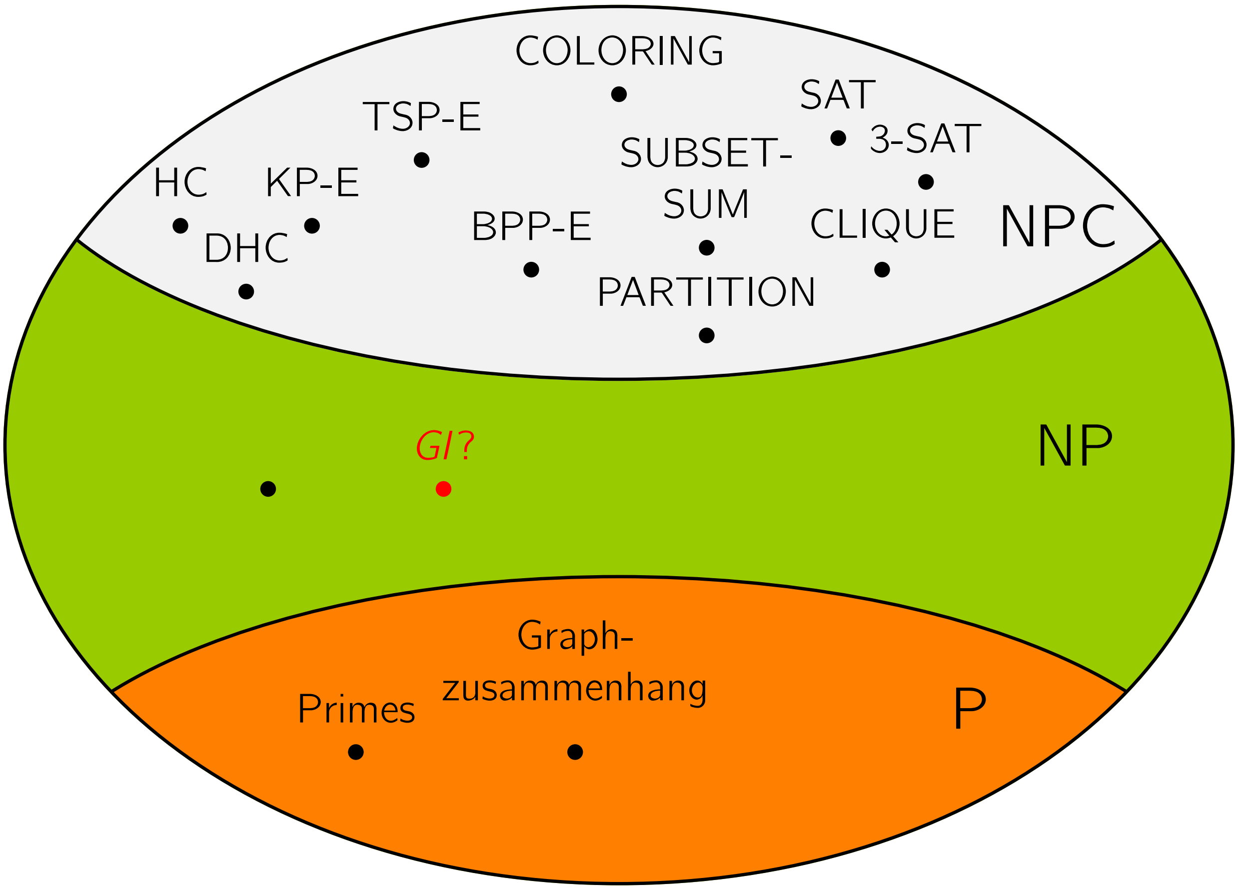
Ein Entscheidungsproblem L heißt NP-intermediate, wenn $L \in NP$ aber sowohl $L \notin \text{P}$ als auch $L \notin \text{NPC}$ gilt.

- Möglicherweise könnte GI NP-intermediate sein.

Satz von Ladner [1975]

Wenn $\text{P} \neq \text{NP}$, dann gibt es Probleme, die NP-intermediate ist.

Die Komplexitätslandschaft



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

Graphisomorphieproblem – Algorithmen

- Trivialer Weise kann man GI in Zeit $O(n! \cdot n^2)$ lösen.
- Lange Zeit hatte der beste (bekannte) Algorithmus für GI eine Laufzeit von $2^{O(\sqrt{n \log n})}$. [Babai, Luks] (1983)
- Jetzt ist ein Algorithmus bekannt, der das Graphisomorphieproblem in Zeit $2^{p(\log n)}$ löst, wobei p ein Polynom ist. [Babai] (2015)
- Dieser Algorithmus verwendet die algorithmische und strukturelle Theorie der Permutationsgruppen.

Eine Laufzeit der Form $2^{p(\log n)}$ für ein Polynom p nennt man quasi-polynomiell.

Die Exponential time hypothesis – ETH

- Wenn $\mathrm{P} \neq \mathrm{NP}$, dann kann man 3-SAT nicht in polynomieller Zeit lösen.
- Aber dann ist immer noch unklar, wie schnell man 3-SAT lösen kann.

Hypothese (Exponential time hypothesis – ETH)

Es gibt ein $\delta > 0$, so dass kein Algorithmus 3-SAT in Zeit $O(2^{\delta n})$ löst.

Diese Hypothese impliziert $P \neq NP$.

$$\begin{array}{ccl} 2^{p(n)} & \sim & \text{exponentiell} \\ 2^{p(\log n)} & \sim & \text{quasi-polynomiell} \end{array}$$

Beobachtung: Wenn $L \leq_p L'$ und L' in quasi-polynomieller Zeit gelöst werden kann, dann kann auch L in quasi-polynomieller Zeit gelöst werden.

Es folgt: Wenn GI NP-vollständig ist, dann ist 3-SAT in quasi-polynomieller Zeit lösbar, und damit wäre die ETH falsch.