Definition: INDEPENDENT SET (Max ...)

Gegeben: (ungerichteter) Graph $G = (V, E)$, $k \in \mathbb{N}$

Gefragt: Besitzt G ein Independent Set der Größe k?

(Ein Independent Set der Größe k ist eine Teilmenge V' der Knotenmenge V mit $|V'| = k$, sodaß $\forall u, v \in V'$ gilt: $\{u, v\} \notin E$).

Theorem: INDEPENDENT SET ist **NP**-vollständig.

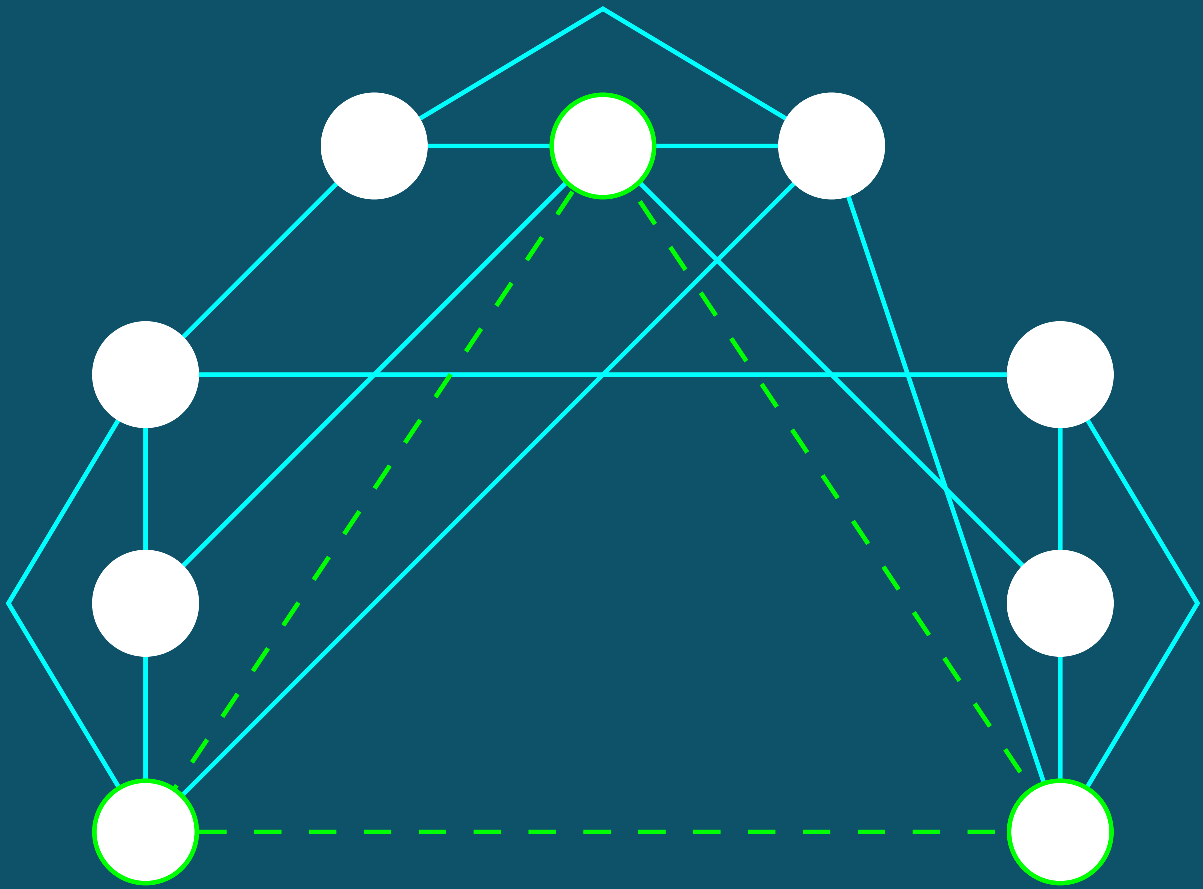
Beweis:

Membership: Guess und check Argument $\surd$

Hardness: CLIQUE $\leq_p$ INDEPENDENT SET.

Bemerkung: Es genügt, den Graphen $G = (V, E)$ des CLIQUE-Problems $< G, k >$ zum Graphen $G' = (V, \overline{E})$ zu invertieren ($\{u, v\} \in E \Leftrightarrow \{u, v\} \notin \overline{E}$), um das INDEPENDENT SET-Problems $< G', k >$ zu bekommen. $\surd$ $\triangleright$

Invertierter Beispielgraph aus dem Beweis für Clique mit Independent Set der Größe 3:



Definition: VERTEX COVER (Node Cover, Min ...)

Gegeben: (ungerichteter) Graph $G = (V, E)$, $k \in \mathbb{N}$

Gefragt: Besitzt G einen Vertex Cover der Größe k? (Ein Vertex Cover der Größe k ist eine Teilmenge $V' \subseteq V$ mit $|V'| = k$, sodaß $\forall \{u, v\} \in E$ gilt: $u \in V'$ oder $v \in V'$, $\rightarrow$ jede Kante aus E wird von mindestens einem Knoten aus V' abgedeckt, d.h. berührt).

Theorem: VERTEX COVER ist **NP**-vollständig.

Beweis:

Membership: Guess und check Argument $\surd$

Hardness: INDEPENDENT SET $\leq_p$ VERTEX COVER.

Wir zeigen: I ist ein Independent Set der Größe i des Graphen $G = (V, E)$ $\Leftrightarrow$ $V' \stackrel{\text{def}}{=} V - I$ ist ein Vertex Cover der Größe $k \stackrel{\text{def}}{=} |V| - i$ von G.

$\triangleright$

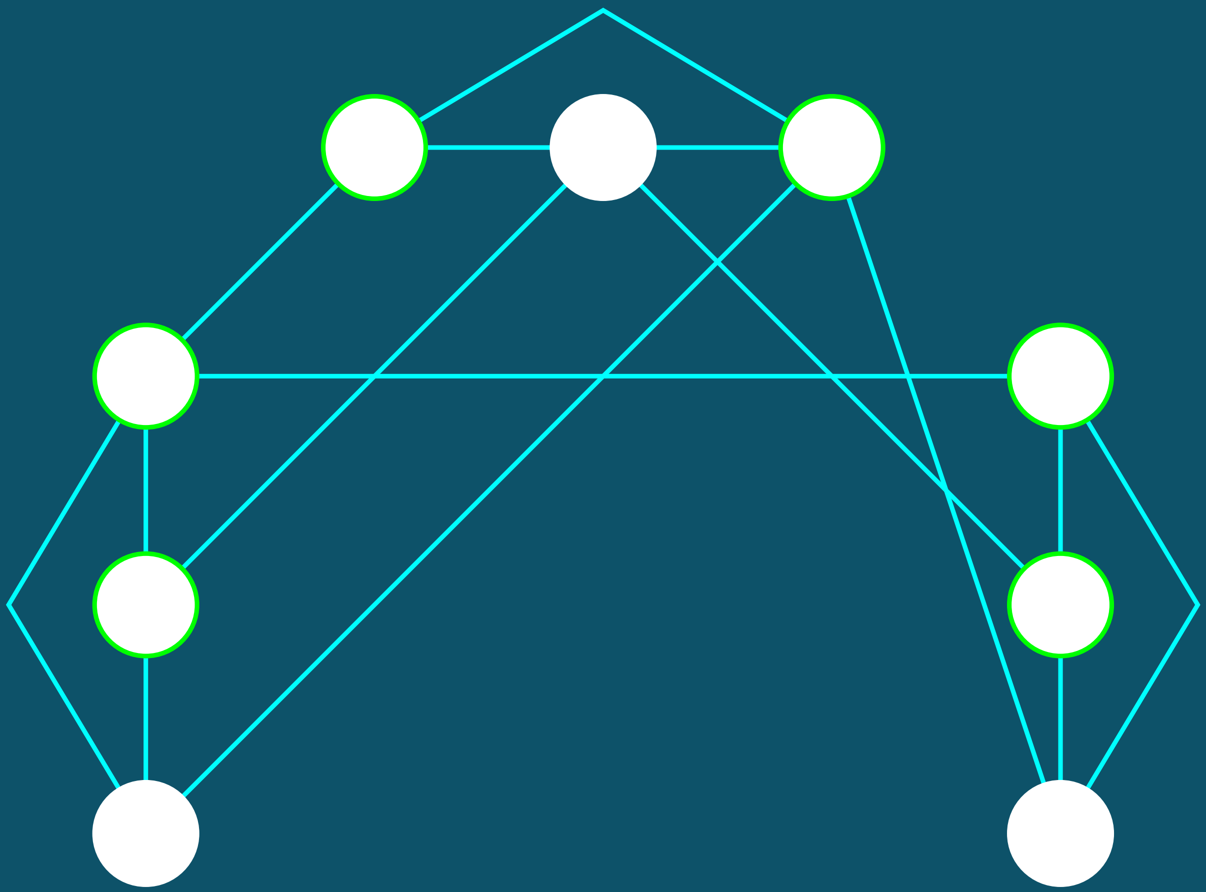
VERTEX COVER (cont)

I ist ein Independent Set der Größe i von $G = (V, E)$:

- $\Leftrightarrow |I| = i$ und $\forall u, v \in I \subseteq V \quad \{u, v\} \notin E$
- $\Leftrightarrow |I| = i$ und $\forall \{u, v\} \in E \quad u \notin I$ oder $v \notin I$
- $\Leftrightarrow V' \stackrel{\text{def}}{=} V - I, \ |V'| = |V| - i$ und $\forall \{u, v\} \in E$ $u \in V'$ oder $v \in V'$
- $\Leftrightarrow V' \stackrel{\text{def}}{=} V - I$ ist ein Vertex Cover der Größe $k \stackrel{\text{def}}{=} |V| - i$ von G.

Die Konstruktion (Berechnung von $k = |V| - i$; G bleibt ja gleich) ist trivialerweise polynomiell machbar. $\surd$ $\triangleright$

Beispielgraph aus Beweis für Independent Set mit Vertex Cover der Größe $6 = 9 - 3$:



Definition: SUBSET SUM

Gegeben: Zahlen $a_1, a_2, \ldots, a_k \in \mathbb{N}$ und $b \in \mathbb{N}$

Gefragt: Gibt es $I \subseteq \{1, 2, \ldots, k\}$ mit $\sum_{i \in I} a_i = b$?

Theorem: SUBSET SUM ist **NP**-vollständig.

Beweis:

Membership: Guess und check Argument $\surd$

Hardness: 3CNF $\leq_p$ SUBSET SUM.

Sei $F \stackrel{\text{def}}{=} \bigwedge_{i=1}^{m} (z_{i,1} \vee z_{i,2} \vee z_{i,3})$ eine beliebige 3CNF Formel wobei $z_{i,j} \in \{x_1, x_2, \ldots, x_n\} \cup \{\neg x_1, \neg x_2, \ldots, \neg x_n\}$.

Dann ist die Zahl b gegeben durch (zB im Dezimalsystem)

$$b \stackrel{\text{def}}{=} \underbrace{444\ldots444}_{m}\underbrace{11\ldots11}_{n}$$

Falls zB F aus 3 Klauseln besteht und darin 5 Variablen vorkommen, so ist $b = 444\,11111$.

▷

zB $F = (x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_5 \vee x_4) \wedge (\neg x_2 \vee \neg x_2 \vee \neg x_5)$

Wir müssen noch die Zahlen a_i definieren (4 Klassen):

- (1) Positive Variablenvorkommen (inkl. Anzahl):

$$\begin{aligned}
v_1 &= 100\ \ 10000 \\
v_2 &= 000\ \ 01000 \\
v_3 &= 000\ \ 00100 \\
v_4 &= 010\ \ 00010 \\
v_5 &= 110\ \ 00001
\end{aligned}$$

- (2) Negative Variablenvorkommen (inkl. Anzahl):

$$\begin{aligned}
v'_1 &= 010\ \ 10000 \\
v'_2 &= 002\ \ 01000 \\
v'_3 &= 100\ \ 00100 \\
v'_4 &= 000\ \ 00010 \\
v'_5 &= 001\ \ 00001
\end{aligned}$$

▷

- (3) Ausgleichszahlen (falls ein Literal **`falsch`**):

$$\begin{aligned} c_1 &= 100\ 00000 \\ c_2 &= 010\ 00000 \\ c_3 &= 001\ 00000 \end{aligned}$$

- (4) Ausgleichszahlen (falls zwei Literale **`falsch`**):

$$\begin{aligned} d_1 &= 200\ 00000 \\ d_2 &= 020\ 00000 \\ d_3 &= 002\ 00000 \end{aligned}$$

- Es gibt keine Ausgleichszahlen für den Fall, daß drei Literale **`falsch`** sind :-)

- $\exists$ erfüllende Belegung B $\Leftrightarrow$
  $\exists$ Auswahl A der Zahlen, die sich zu b aufsummiert:

x_i in B **`wahr`** $\Leftrightarrow$ v_i in A

x_i in B **`falsch`** $\Leftrightarrow$ v_i' in A

▷

- Da hinterer Teil von $b = 444\ 11111$ nur aus Einsen besteht kann nur entweder v_i oder v_i' in A sein; einer davon muß aber drin sein, da die anderen Variablen an der entsprechenden Stelle alle eine Null stehen haben.
- Bsp: $B = \{x_1, x_4\}$ (andere `falsch`) ergibt Auswahl v_1, v_2', v_3', v_4, v_5'. Summe davon ist 213 11111 (wichtig: im vorderen Teil sind alle größer als Null, da die Ausgleichszahlen nur Zahlen größer als Null ausgleichen können, und hinten sind alle Eins, dh alle Variablen haben einen Wert zugewiesen bekommen).
- Durch Hinzunehmen von geeigneten Ausgleichzahlen, in diesem Fall d_1, c_2, d_2, c_3, erhalten wir die gewünschte Summe $b = 444\ 11111$.
- Da es keine Überträge zwischen den Stellen gibt, „funktioniert" der Beweis in beide Richtungen. $\surd$

Definition: RUCKSACK (Engl: Knapsack)

Gegeben: n Objekte, jedes mit Wert $v_i \in \mathbb{N}$ und Gewicht $w_i \in \mathbb{N}$, maximale Kapazität $W \in \mathbb{N}$ und Vorgabe $V \in \mathbb{N}$.

Gefragt: Gibt es $S \subseteq \{1, 2, \ldots, n\}$ mit $\sum_{i \in S} w_i \leq W$ und $\sum_{i \in S} v_i \geq V$?

Theorem: RUCKSACK ist **NP**-vollständig.

Beweis:

Membership: Guess und check Argument $\surd$

Hardness: SUBSET SUM $\leq_p$ RUCKSACK: Für ein beliebiges Subset-Sum-Problem setzt man $n \stackrel{\text{def}}{=} k$, $v_i \stackrel{\text{def}}{=} w_i \stackrel{\text{def}}{=} a_i$ und $W \stackrel{\text{def}}{=} V \stackrel{\text{def}}{=} b$ und erhält dadurch ein Rucksack-Problem. Dh SUBSET SUM ist eigentlich nur ein Spezialfall (Technik!) von RUCKSACK, daher ist das größere Problem RUCKSACK ebenfalls **NP**-vollständig. $\surd$

$\triangleright$

Theorem: RUCKSACK ist in $O(nW)$ lösbar.

Beweis: Sei $V(w,i)$ der größte Wert einer Teilmenge der ersten i Objekte, deren Gewicht genau w ergibt. Es genügt, $n \times W$ Einträge in der Tabelle $V(w,i)$ mit Werten bis zur Größe der Vorgabe V zu berechnen (sobald ein Wert $\geq V$ ist, lautet die Antwort „Ja“). Es gilt:

$$V(w, i+1) = \max\{V(w,i), v_{i+1} + V(w - w_{i+1}, i)\}$$

mit $\forall w \geq 0 \quad V(w,0) = 0$ und $\forall w < 0 \quad V(w,i) = -\infty$, dh $\rightarrow$ dynamische Programmierung kann in $O(nW)$ (auf RAM) das Problem RUCKSACK lösen. $\surd$

Bemerkung: Kein Widerspruch mit **NP**-Vollständigkeit von RUCKSACK, da exponentieller Sprung bezogen auf Länge der Eingabe ($\rightarrow \log(W) + \ldots$).

Algorithmen wie diesen nennt man pseudo-polynomiell, und die entsprechenden Probleme (RUCKSACK, SUBSET SUM, ...) schwach **NP**-vollständig. Eigenschaft: Mittels dynamischer Programmierung (= Wiederverwendung von Zwischenergebnissen) effizient lösbar, solange die vorkommenden Zahlen „klein“ bleiben.

$\rightarrow$ In der Reduktion von 3CNF nach SUBSET SUM haben wir exponentiell große Zahlen verwendet.

Umgekehrt ist ein Problem stark **NP**-vollständig, wenn es **NP**-vollständig bleibt, auch wenn alle vorkommenden Zahlen polynomiell begrenzt sind (Bsp: SAT, 3CNF, CLIQUE, VERTEX COVER, ...). $\longrightarrow$ unäre statt binäre Codierung von Zahlen wäre OK.

Bemerkung: Ein pseudo-polynomieller Algorithmus für ein stark **NP**-vollständiges Problem würde **P**=**NP** implizieren.

Es gibt auch Probleme mit natürlich vorkommenden Zahlenmengen, ähnlich wie in SUBSET SUM und RUCKSACK, die aber stark **NP**-vollständig sind, zB das Problem des Handelsreisenden (Traveling salesman problem, kurz TSP). Wir zeigen es im folgenden in mehreren Schritten.

Definition: Gerichteter Hamiltonscher Kreis (GHK)

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

Gefragt: Besitzt G einen Hamiltonschen Kreis? (Dies ist eine Permutation π der Knotenindizes $(v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)})$, so daß für $i \in \{1, \ldots, n-1\}$ gilt: $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ und außerdem $(v_{\pi(n)}, v_{\pi(1)}) \in E$).

Theorem: GHK ist **NP**-vollständig.

Beweis:

Membership: Guess und check Argument $\surd$

Hardness: 3CNF $\leq_p$ GHK.

Sei $F \stackrel{\text{def}}{=} \bigwedge_{i=1}^{m} (z_{i,1} \vee z_{i,2} \vee z_{i,3})$ eine beliebige 3CNF Formel wobei $z_{i,j} \in \{x_1, x_2, \ldots, x_n\} \cup \{\neg x_1, \neg x_2, \ldots, \neg x_n\}$.

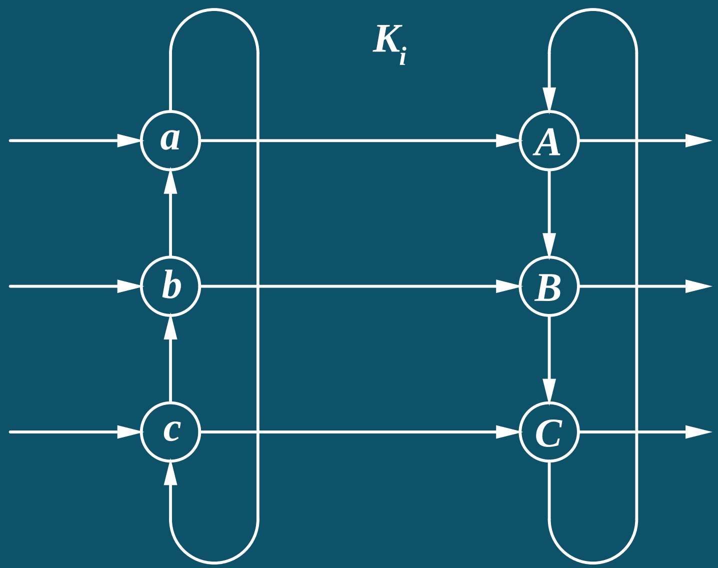
Dann konstruieren wir ein GHK Problem mit Graphen G. ▷

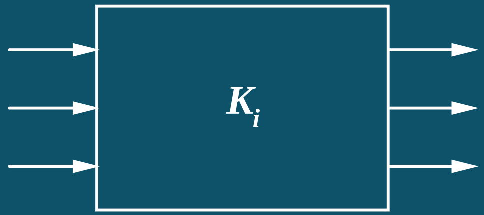
Für jede Variable x_j gibt es einen Knoten j, wie folgt:

Von jedem Knoten j gehen jeweils zwei Kanten aus. Die entsprechenden Pfade führen dann zu weiteren Teilgraphen, die wir als nächstes beschreiben, und enden dann im Knoten $j+1$ (vom Knoten n aus führen die Pfade zurück zum Knoten 1).

▷

Für jede Klausel $(z_{i,1} \vee z_{i,2} \vee z_{i,3})$ gibt es einen Teilgraph K_i, im Detail wie links, im folgenden abgekürzt wie rechts:

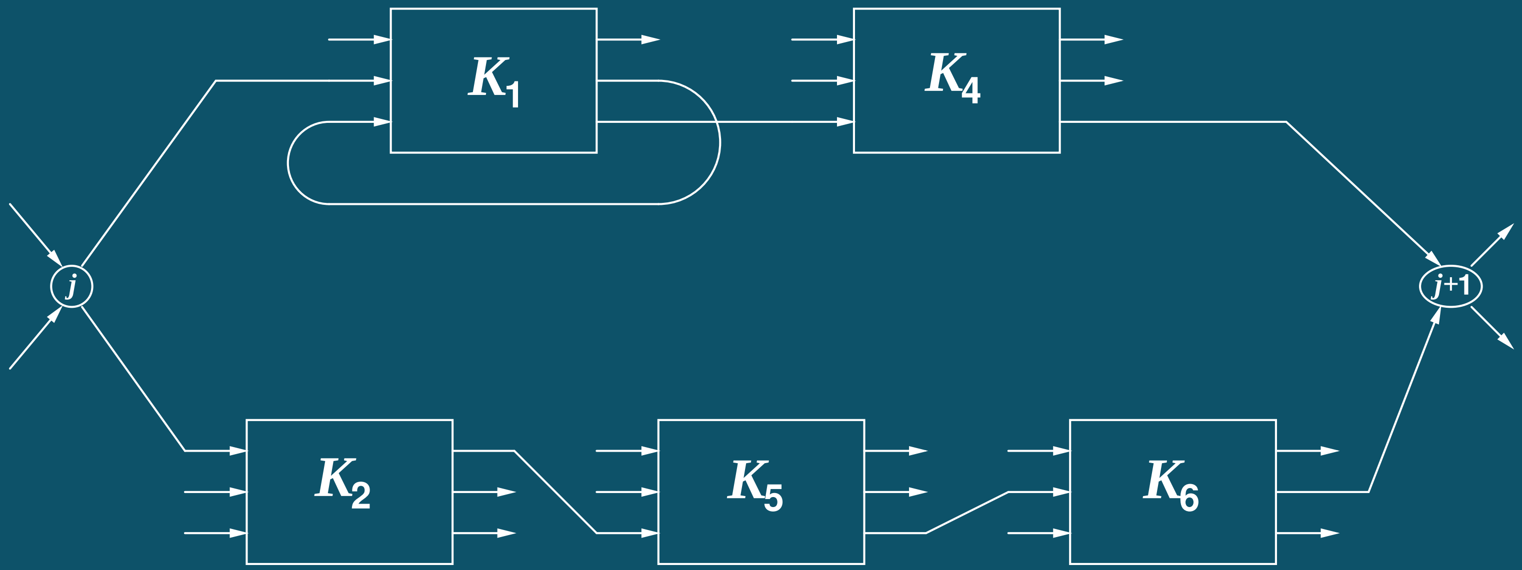




Lemma: Wenn der K_i umgebende Graph G einen Hamiltonschen Kreis besitzt, so verläuft dieser, wenn er K_i passiert, folgendermaßen: Wenn er bei a (bzw. b bzw. c) hineinläuft, so verläßt er K_i bei A (bzw. B bzw. C).

▷

Der obere vom Knoten j ausgehende Weg orientiert sich an den Vorkommen von x_j in den Klauseln, der untere an denen von $\neg x_j$. Bsp: x_j kommt in Klausel 1 an der 2. und 3. und in Klausel 4 an der 3. Position vor, während $\neg x_j$ in Klausel 2 an der 1., in Klausel 5 an der 3. und in Klausel 6 an der 2. Position vorkommt. Dann sehen die Verbindungen zwischen Knoten j und Knoten $j+1$ wie folgt aus:



Bemerkung: Es gibt keine „frei hängenden" Kanten in G.

▷

Zu zeigen: F ist erfüllbar $\Leftrightarrow$ G hat Hamiltonschen Kreis

$\Rightarrow$) Wenn die Variable x_j in der F erfüllenden Belegung den Wert **`wahr`** hat, dann folgt man dem oberen Pfad, sonst dem unteren. Danach durchläuft der Pfad die entsprechenden K_i's. Jedes K_i muß nun je nachdem, wie viele Literale es **`wahr`** machen, in einer bestimmten Sequenz durchlaufen werden.

Machen zB $z_{i,2} = x_5$ und $z_{i,3} = \neg x_1$ die i. Klausel **`wahr`**, dann muß der Hamiltonsche Kreis durch K_i einmal (beim oberen Pfad zwischen Knoten 5 und 6) den Weg $b - a - A - B$ und einmal (beim unteren Pfad zwischen Knoten 1 und 2) den Weg $c - C$ nehmen.

Dies kann immer so arrangiert werden, daß alle Knoten genau einmal durchlaufen werden, es gibt also sicher einen Hamiltonschen Kreis.

▷

$\Leftarrow$) Angenommen, G besitze einen Hamiltonschen Kreis. Dieser durchläuft Knoten 1, dann gewisse K_i's, Knoten 2, dann gewisse K_i's, usw., bis er zum Knoten 1 zurückkehrt. An dieser Stelle brauchen wir das Lemma über die K_i's: Es ist dem Hamiltonschen Kreis nämlich nicht möglich, die Graphen K_i anders als vorgesehen zu passieren. Wir definieren nun eine Variablen-Belegung für x_j anhand dessen, ob der Hamiltonsche Kreis den Knoten j nach oben (=**`wahr`**) oder nach unten (=**`falsch`**) verläßt. Diese Belegung erfüllt F, denn jede Klausel wird mindestens einmal durchlaufen—die entsprechende Klausel erhält dabei ja den Wert **`wahr`**. $\surd$

Definition: Hamiltonian Cycle

Gegeben: (ungerichteter) Graph $G = (V, E)$

Gefragt: Besitzt G einen Hamiltonschen Kreis? (Dies ist eine Permutation π der Knotenindizes $(v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)})$, so daß für $i \in \{1, \ldots, n-1\}$ gilt: $\{v_{\pi(i)}, v_{\pi(i+1)}\} \in E$ und außerdem $\{v_{\pi(n)}, v_{\pi(1)}\} \in E$).

Theorem: Hamiltonian Cycle ist **NP**-vollständig.

Beweis:

Membership: Guess und check Argument $\surd$

Hardness: GHK $\leq_p$ Hamiltonian Cycle.

Wir zeigen, daß man den gerichteten Fall auf den ungerichteten reduzieren kann. Wir geben dazu an, wie ein gerichteter Graph in einen ungerichteten umgewandelt werden kann, so daß die Hamiltonsche-Kreis Eigenschaft erhalten bleibt.

$\triangleright$

Jeder Knoten mit gewissen herein- und herausgehenden Kanten wird lokal ersetzt durch drei Knoten:

Es gilt: Wenn der gerichtete Graph einen Hamiltonschen Kreis besitzt, dann besitzt auch der ungerichtete einen. Umgekehrt muß jeder Hamiltonsche Kreis im ungerichteten Graphen jede der Dreiergruppen entweder von links nach rechts, oder von rechts nach links durchlaufen (denn sonst kommt es beim mittleren Knoten zu einer „Sackgasse“). $\rightarrow$ Es wird genau die Pfeilrichtung (oder Gegenpfeilrichtung) eingehalten, und daher läßt sich aus dem ungerichteten Hamiltonschen Kreis auch wieder ein entsprechender im gerichteten Graphen gewinnen. $\surd$

Definition: TSP (Problem des Handelsreisenden)

Gegeben: Eine $n \times n$ Matrix $(M_{i,j})$ von „Reisekosten" zwischen n „Städten" und ein „Reisebudget" k.

Gefragt: Gibt es eine Permutation π (eine „Rundreise"), so daß $\sum_{i=1}^{n-1} M_{\pi(i),\pi(i+1)} + M_{\pi(n),\pi(1)} \leq k$?

Theorem: TSP ist **NP**-vollständig.

Beweis:

Membership: Guess und check Argument $\surd$

Hardness: Hamiltonian Cycle $\leq_p$ TSP:

$$G = (\{1, \ldots, n\}, E) \mapsto \begin{cases} \text{Matrix:} \quad M_{i,j} = \begin{cases} 0, & \{i,j\} \in E \\ 1, & \{i,j\} \notin E \end{cases} \\ \text{Budget:} \quad 0 \qquad\qquad \surd \end{cases}$$

Daß TSP auch stark **NP**-vollständig ist sieht man daran, daß in alle Reduktionen, ausgehend von der beliebigen Sprache $L \in$ **NP** in Cooks Beweis der **NP**-Vollständigkeit von SAT, nur polynomiell begrenzte Zahlen vorkamen.

Zusammenfassung der NP-Vollständigkeits Teils: