VL-15: NP-vollständige Graphprobleme

(Berechenbarkeit und Komplexität, WS 2018)

Gerhard Woeginger

WS 2018, RWTH

Organisatorisches

- Nächste Vorlesung:
  Freitag, Januar 11, 16:30–18:00 Uhr, Audimax

- Webseite:
  http://algo.rwth-aachen.de/Lehre/WS1819/BuK.php
  ($\longrightarrow$ Arbeitsheft zur Berechenbarkeit)
  ($\longrightarrow$ Arbeitsheft zur NP-Vollständigkeit)

Wiederholung

Wdh.: NP-schwer & NP-Vollständig

Definition

- Ein Problem L heisst NP-schwer, falls $\forall L' \in NP : \; L' \leq_p L$
- Ein Problem L heisst NP-vollständig, falls $L \in$ NP und L NP-schwer.

Satz

Wenn L NP-vollständig ist, dann gilt: $\quad L \in P \;\Rightarrow\; P = NP$

Unter der Annahme $P \neq NP$ (Standardannahme) besitzt also kein NP-vollständiges Problem einen polynomiellen Algorithmus.

Wdh.: Der Satz von Cook & Levin

Problem: Satisfiability (SAT)

Eingabe: Boole'sche Formel φ in CNF über der Variablenmenge X

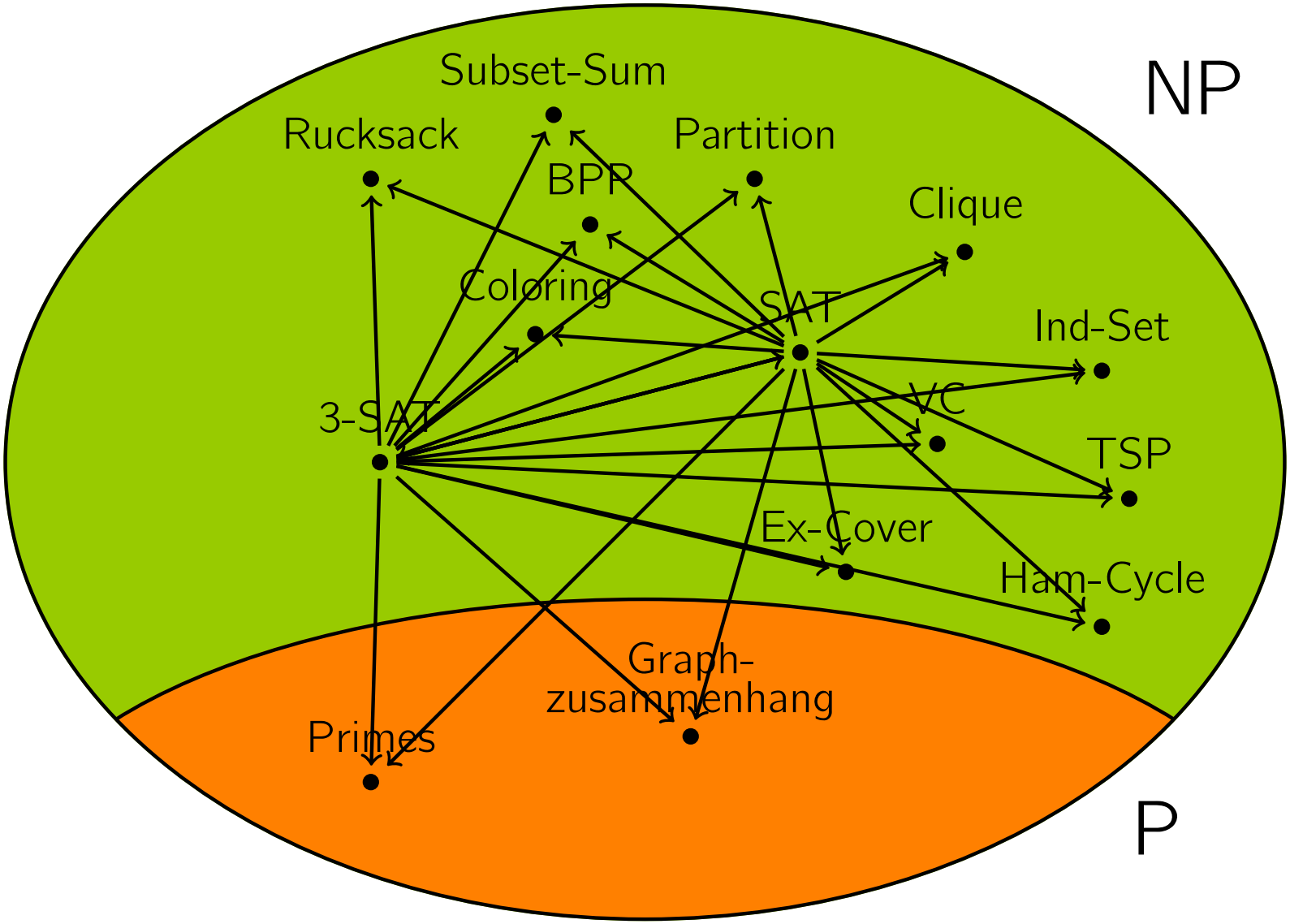
Frage: Existiert eine Wahrheitsbelegung von X, die φ erfüllt?

Satz (Cook & Levin)

SAT ist NP-vollständig.

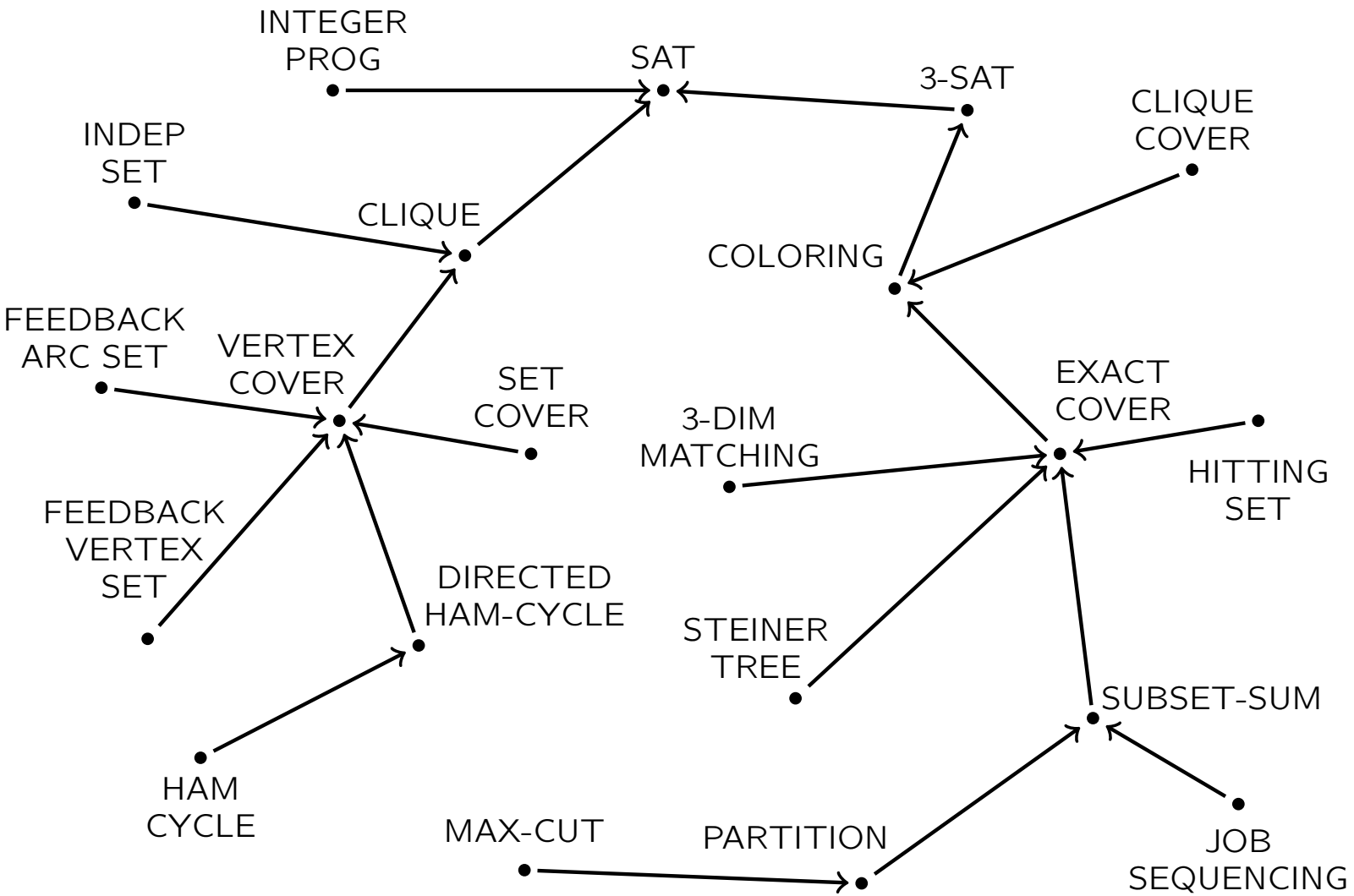
- Arbeitsphase A: Für jeden Zeitpunkt t beschreiben die Variablen $Q(t, q)$, $H(t, j)$ und $B(t, j, a)$ eine legale Konfiguration.
- Arbeitsphase B: Die Konfiguration zum Zeitpunkt $t + 1$ entsteht legal aus der Konfiguration zum Zeitpunkt t.
- Arbeitsphase C: Startkonfiguration und Endkonfiguration sind legal.

Wdh.: Die Komplexitätslandschaft



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

Wdh.: Landkarte mit Karp's 20 Reduktionen



Wdh.: Kochrezept für NP-Vollständigkeitsbeweise

Kochrezept:

1. Man zeige $L \in NP$.

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

Vorlesung VL-15
Einige NP-vollständige Graphprobleme

- NP-Vollständigkeit von CLIQUE
- NP-Vollständigkeit von INDEP-SET
- NP-Vollständigkeit von Vertex Cover

- NP-Vollständigkeit von Ham-Cycle (gerichtet)
- NP-Vollständigkeit von Ham-Cycle (ungerichtet)
- NP-Vollständigkeit des TSP

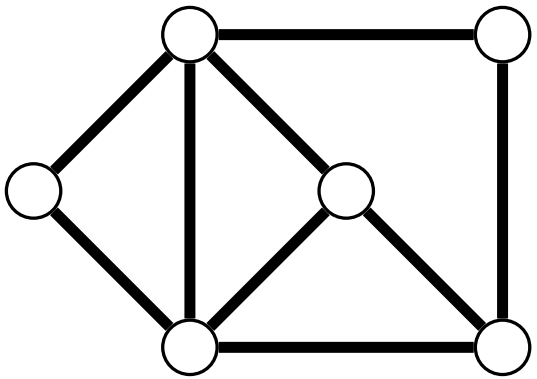
NP-Vollständigkeit von CLIQUE

CLIQUE (1): Definition

Problem: CLIQUE

Eingabe: Ein ungerichteter Graph $G = (V, E)$; eine Zahl k

Frage: Enthält G eine Clique mit $\geq k$ Knoten?



$k = 4$

Dieser Graph hat keine Clique der Grösse 4.

Satz

CLIQUE ist NP-vollständig.

CLIQUE (2): Nach unserem Kochrezept

1. Wir wissen bereits (aus VL-12), dass CLIQUE in NP liegt.

2. Wir wählen die NP-vollständige Sprache $L^* = SAT$
   und wir werden SAT $\leq_p$ CLIQUE zeigen.

3. **(Reduktionsabbildung):**
   Wir konstruieren eine Funktion f, die eine CNF-Formel φ in einen Graphen $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$ transformiert, sodass gilt:

   $$\varphi \text{ ist erfüllbar} \quad \Leftrightarrow \quad G \text{ besitzt } k\text{-Clique}$$

(Die Punkte **4** und **5** des Kochrezeptes werden später erledigt.)

CLIQUE (3): Beschreibung der Funktion f

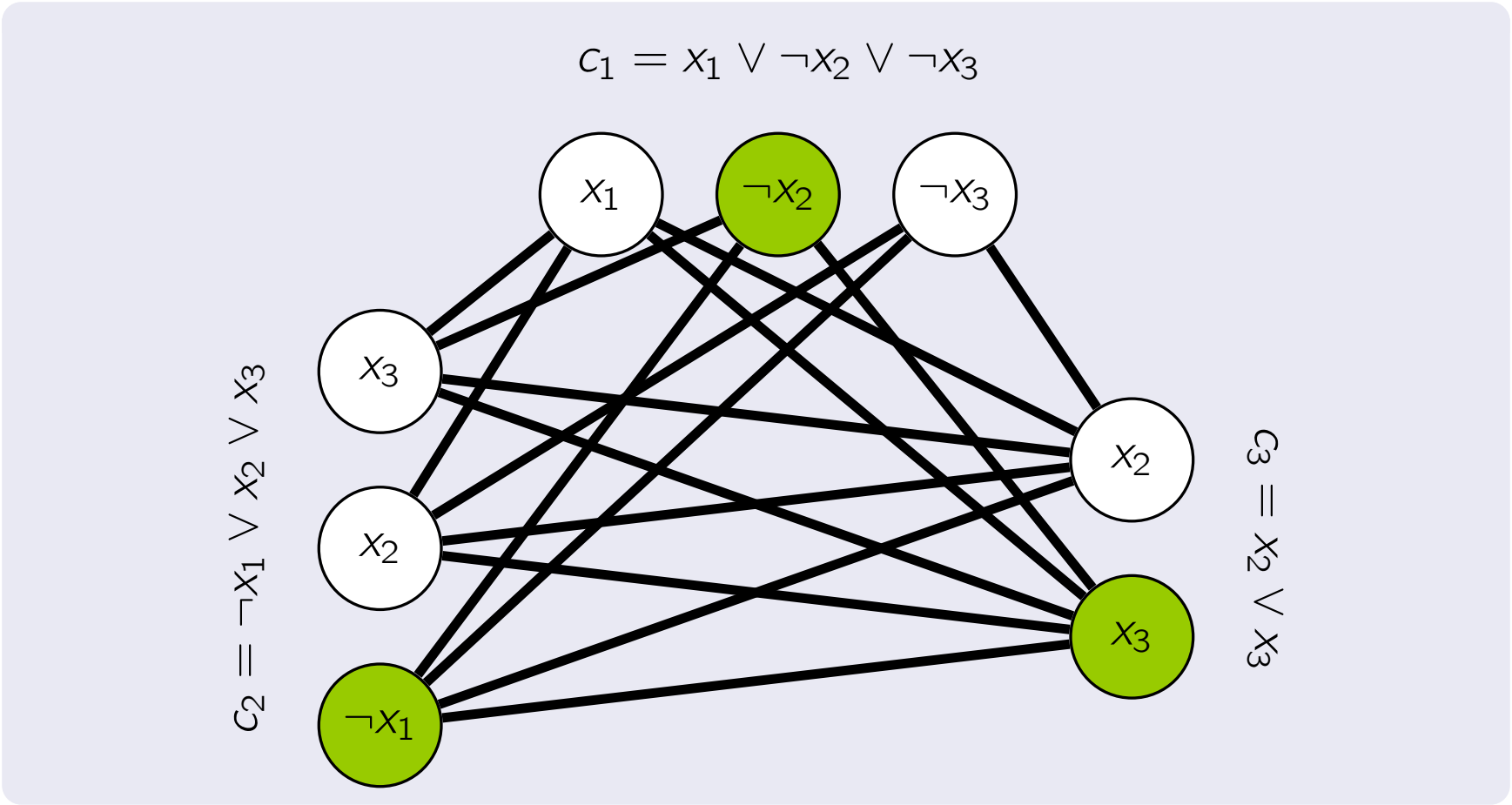
- Es seien $c_1, \ldots, c_m$ die Klauseln der Formel φ.
  Es sei k_i die Anzahl an Literalen in Klausel c_i.
  Es seien $\ell_{i,1}, \ldots, \ell_{i,k_i}$ die Literale in Klausel c_i.

- Für jedes Literal in jeder Klausel erzeugen wir einen entsprechenden Knoten: $V = \{\ell_{i,j} \mid 1 \leq i \leq m,\ 1 \leq j \leq k_i\}$
- Zwei Knoten werden mit einer Kante verbunden,
  wenn sie aus verschiedenen Klauseln stammen und
  wenn ihre Literale nicht Negationen voneinander sind.
- Wir setzen $k = m$.

4. (Polynomielle Zeit):
Die Funktion f ist in Polynomialzeit berechenbar.

CLIQUE (4): 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: $x_1 = 0$, $x_2 = 0$, $x_3 = 1$

CLIQUE (5a): Korrektheit

Lemma A: Formel φ erfüllbar $\Rightarrow$ G hat m-Clique

- Betrachte beliebige erfüllende Belegung von φ
- Bilde Menge U mit einem erfüllten Literal von jeder Klausel
- Behauptung: U bildet m-Clique

Begründung:

- Laut Definition ist $|U| = m$
- Es seien ℓ und ℓ' zwei verschiedene Literale aus U
- Nach Konstruktion kommen ℓ und ℓ' aus verschiedenen Klauseln
- Da ℓ und ℓ' erfüllt sind, sind sie nicht Negationen voneinander.
- Also gibt es eine Kante zwischen ℓ und ℓ'

CLIQUE (5b): Korrektheit

Lemma B: G hat m-Clique $\Rightarrow$ Formel φ erfüllbar

- Betrachte m-Clique U in G
- Dann gehören die Literale in U zu lauter verschiedenen Klauseln

- U enthält somit genau ein Literal pro Klausel
- Kein Literal tritt sowohl positiv als auch negiert auf
- Ergo: Alle diese Literale können gleichzeitig erfüllt werden
- Also ist φ erfüllbar

5. (Korrektheit):
f ist Reduktion: $x \in L^* \Leftrightarrow f(x) \in L$

$\varphi \in \text{SAT} \Leftrightarrow f(\varphi) = \langle G; m \rangle \in \text{CLIQUE}$

NP-Vollständigkeit von INDEP-SET und Vertex Cover

Independent Set

Unabhängige Menge (independent set):
Teilmenge der Knoten, die keine Kanten induziert

Problem: INDEP-SET

Eingabe: Ein ungerichteter Graph $G' = (V', E')$; eine Zahl k'

Frage: Enthält G' eine unabhängige Menge mit $\geq k'$ Knoten?

Satz

INDEP-SET ist NP-vollständig.

Beweisskizze: im Tutorium

- Wir zeigen CLIQUE $\leq_p$ INDEP-SET
- Setze $V' = V$ und $E' = V \times V - E$ und $k' = k$

Vertex Cover (1)

Vertex Cover: Teilmenge der Knoten, die alle Kanten berührt

Problem: Vertex Cover (VC)

Eingabe: Ein ungerichteter Graph $G'' = (V'', E'')$; eine Zahl k''

Frage: Enthält G'' ein Vertex Cover mit $\leq k''$ Knoten?

Satz

Vertex Cover ist NP-vollständig.

Beweisskizze:

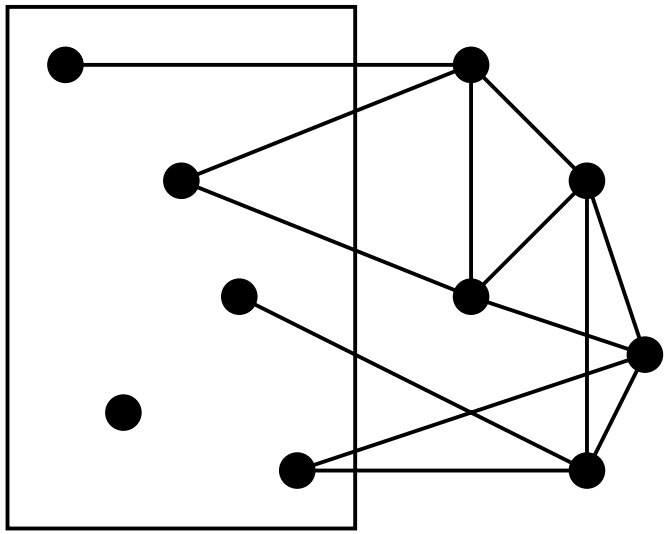
- Wir zeigen INDEP-SET $\leq_p$ Vertex Cover
- Setze $V'' = V'$ und $E'' = E'$ und $k'' = |V'| - k'$

Vertex Cover (2)

Beobachtung

In einem ungerichteten Graphen $G = (V, E)$ gilt für alle $S \subseteq V$:

- S ist unabhängige Menge $\Leftrightarrow$ $V - S$ ist Vertex Cover
- S ist Vertex Cover $\Leftrightarrow$ $V - S$ ist unabhängige Menge



NP-Vollständigkeit von Ham-Cycle (gerichtet)

D-Ham-Cycle (1): Definition

Problem: Gerichteter Hamiltonkreis (D-Ham-Cycle)

Eingabe: Ein gerichteter Graph $G = (V, A)$

Frage: Besitzt G einen gerichteten Hamiltonkreis?

Satz

D-Ham-Cycle ist NP-vollständig.

D-Ham-Cycle (2): Nach unserem Kochrezept

1. D-Ham-Cycle liegt in NP

2. Wir wählen die NP-vollständige Sprache $L^* = SAT$
   und wir werden SAT $\leq_p$ D-Ham-Cycle zeigen.

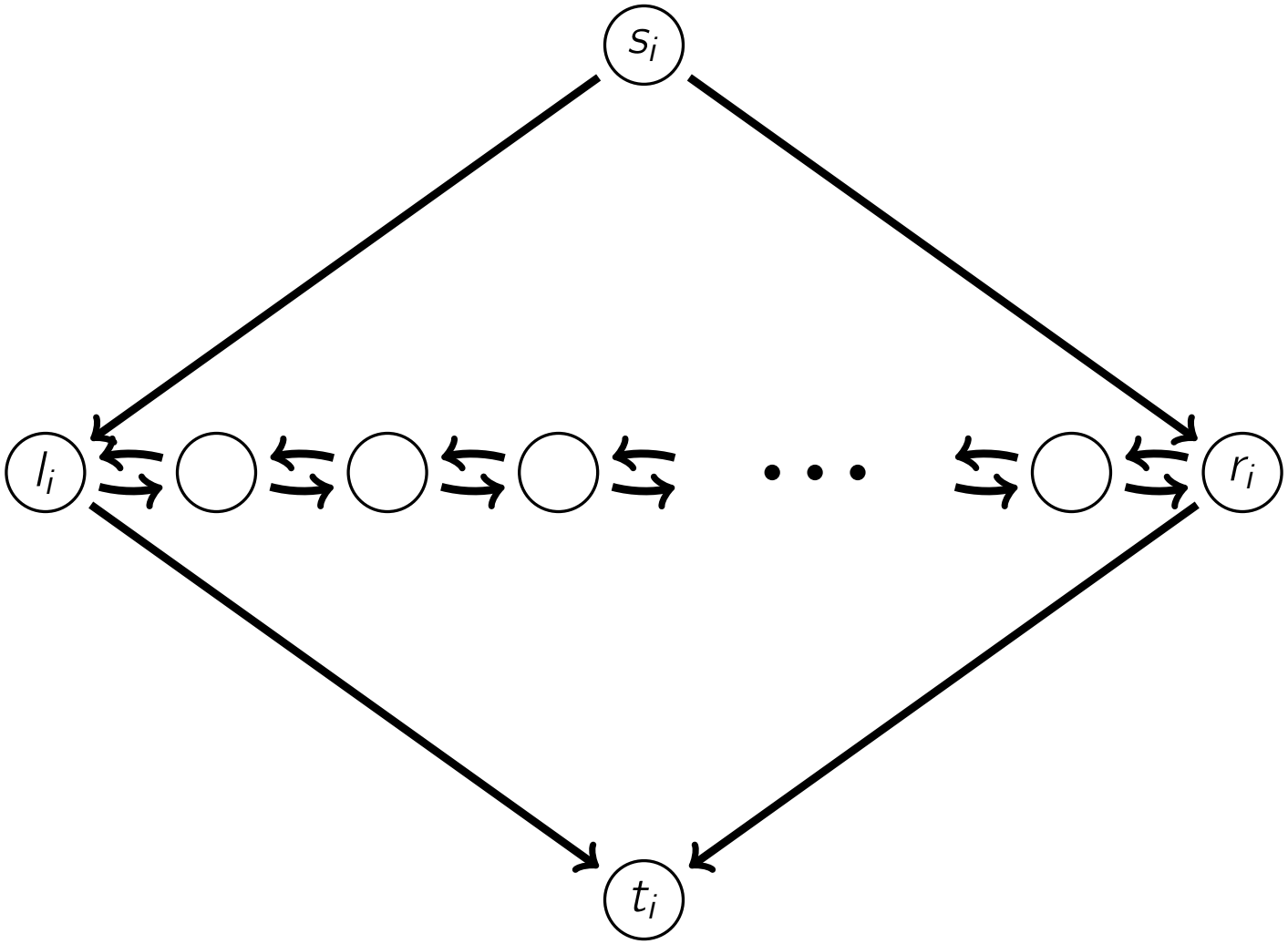
3. **(Reduktionsabbildung):**
   Wir konstruieren eine Funktion f, die eine CNF-Formel φ in einen gerichteten Graphen $G = (V, A)$ transformiert, sodass gilt:

$$\varphi \text{ ist erfüllbar} \quad \Leftrightarrow \quad G \text{ hat gerichteten Hamiltonkreis}$$

Die CNF-Formel φ besteht aus Klauseln $c_1, \ldots, c_m$ mit Boole'schen Variablen $x_1, \ldots, x_n$.

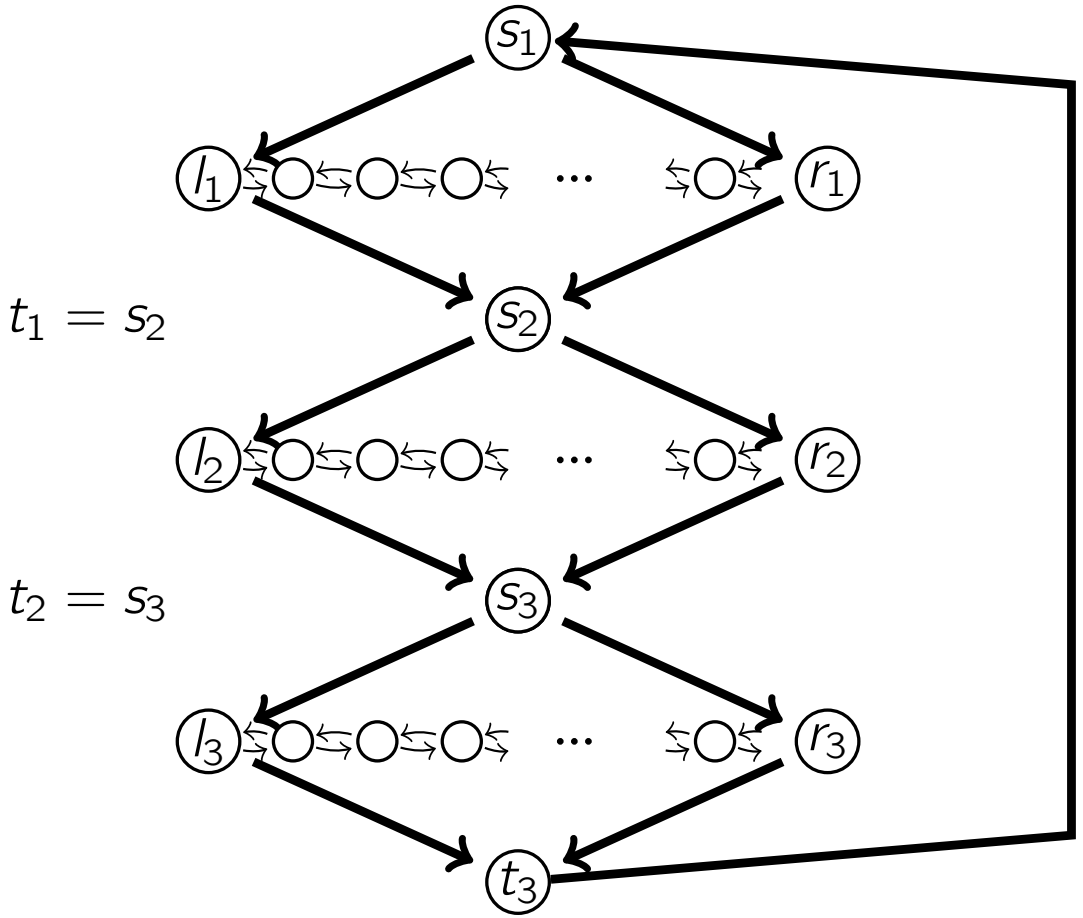
D-Ham-Cycle (3a): Reduktion / Diamantengadgets

Für jede Variable x_i enthält der Graph G das Diamantengadget G_i:



D-Ham-Cycle (3b): Reduktion / Diamantengadgets

Diese n Diamantengadgets 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:



D-Ham-Cycle (3c): Reduktion / Diamantengadgets

In dem resultierenden Graphen besucht jede Rundreise, die im Knoten s_1 startet, die Diamantengadgets in der Reihenfolge $G_1, G_2, \ldots, G_n$.

Die Rundreise hat dabei für jedes Gadget G_i die Freiheit, das Gadget

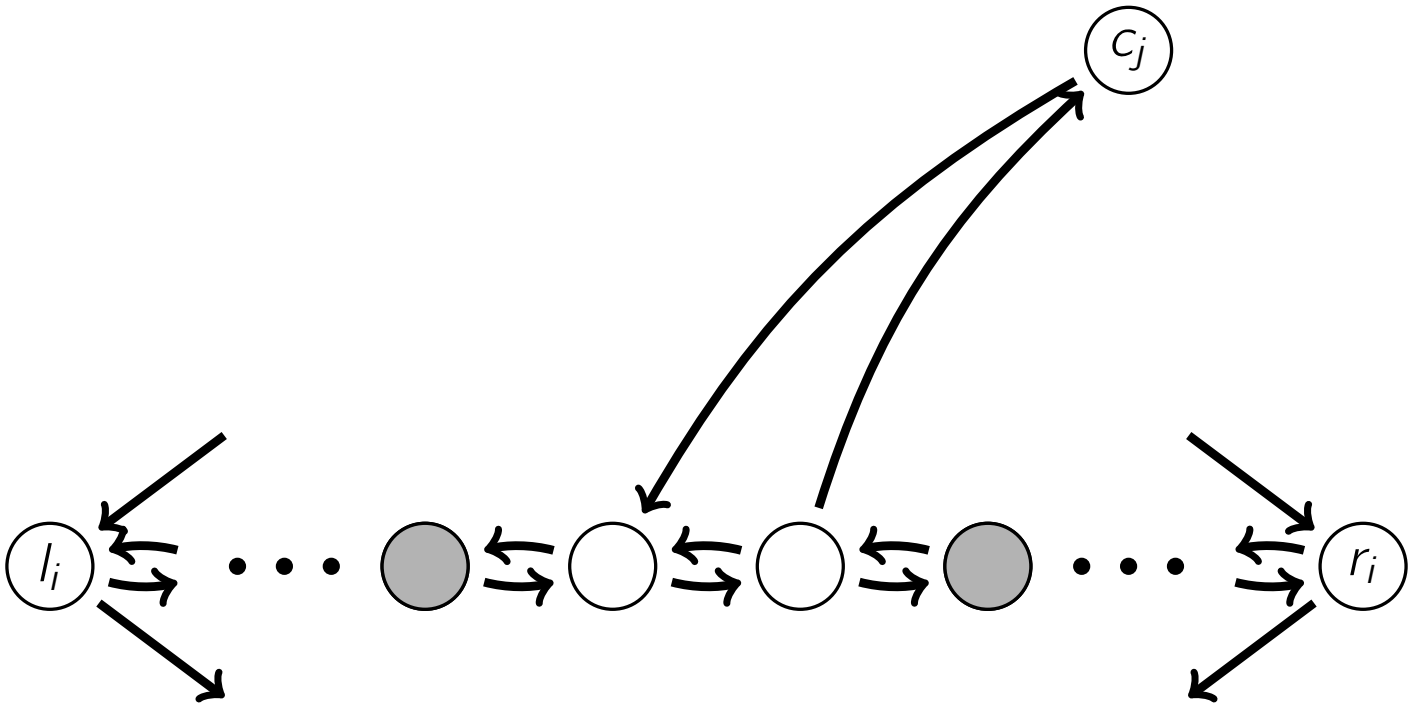
- entweder *von links nach rechts* (also: von l_i bis r_i)
- oder *von rechts nach links* (also: von r_i bis l_i) zu durchlaufen.

Die LR Variante interpretieren wir als Variablenbelegung $x_i = 0$,
und die RL Variante als Variablenbelegung $x_i = 1$.

D-Ham-Cycle (4a): Reduktion / Klauselknoten

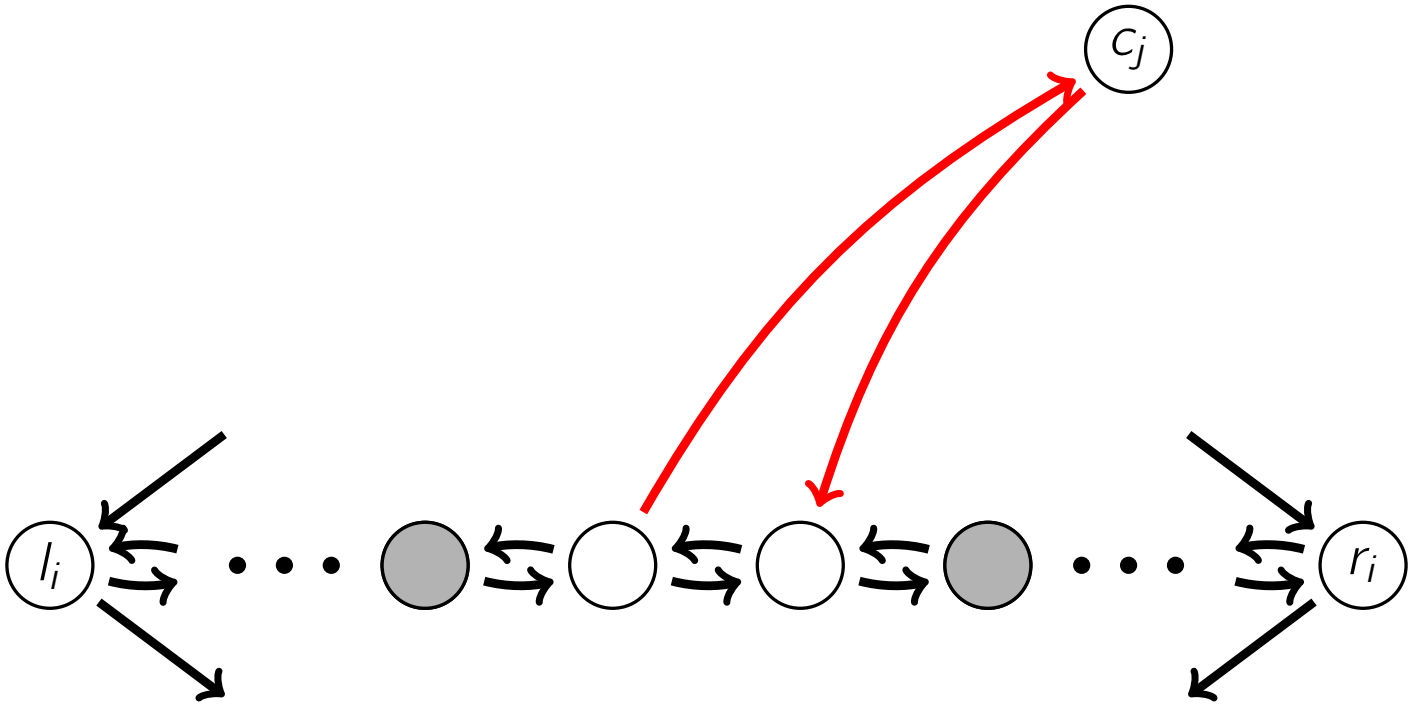
Jetzt fügen wir für jede Klausel c_j einen weiteren Knoten ein.

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



D-Ham-Cycle (4b): Reduktion / Klauselknoten

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



D-Ham-Cycle (4c): Reduktion / Klauselknoten

Frage

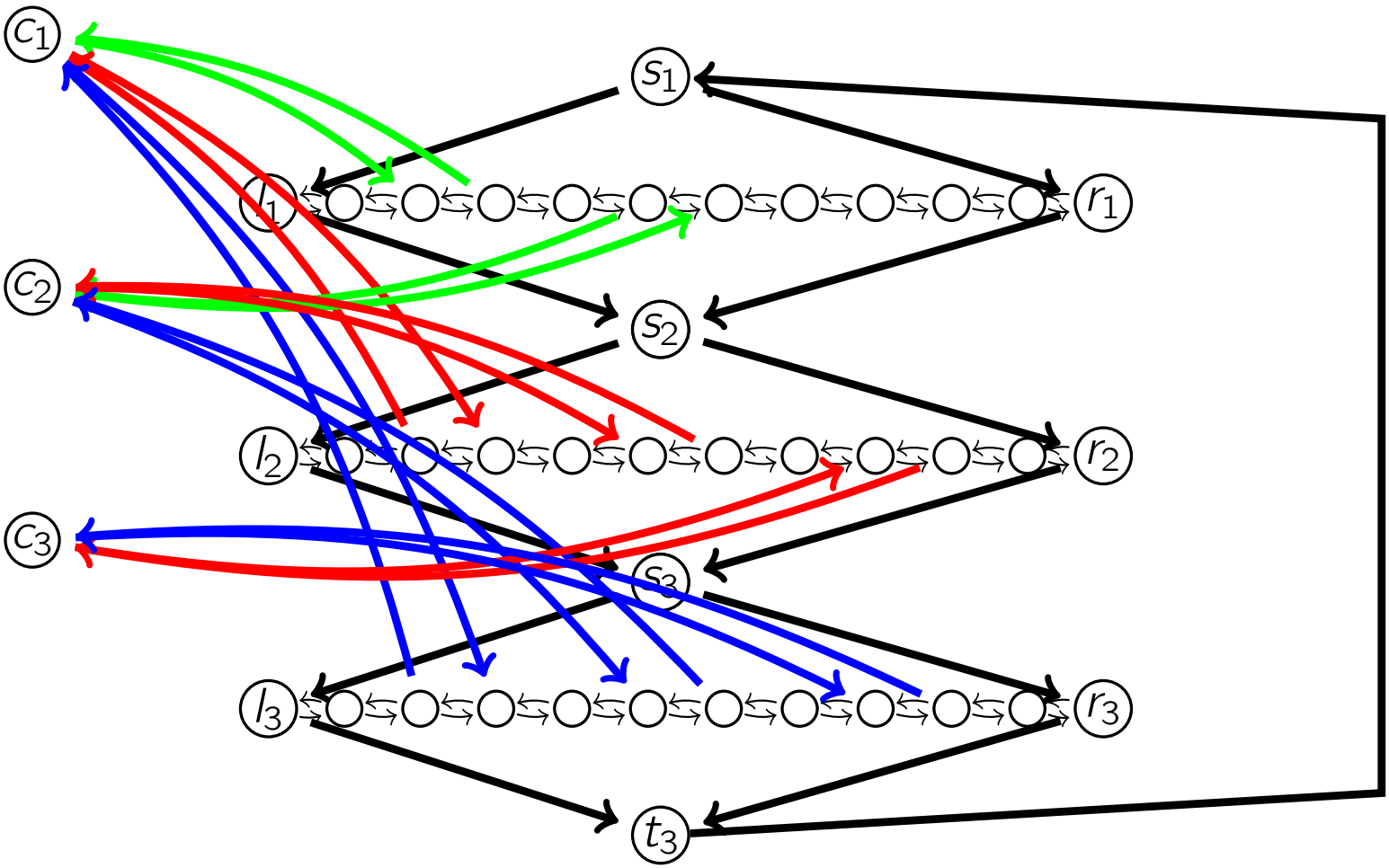
Ist es nach Hinzufügen der Klauselknoten möglich, dass
eine Rundreise zwischen den Diamantengadgets hin- und herspringt,
anstatt sie in der vorgesehenen Reihenfolge zu besuchen?

Antwort

Nein. (Warum??)

D-Ham-Cycle (5): Illustration

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



D-Ham-Cycle (6a): Korrektheit

Lemma A: G hat gerichteten Hamiltonkreis $\Rightarrow$ φ erfüllbar

- Wenn ein Klauselknoten c_j aus einem Gadget G_i heraus von links nach rechts durchlaufen wird, so muss nach unserer Konstruktion die Klausel c_j das Literal $\bar{x}_i$ enthalten.
- Also wird diese Klausel durch die mit der Laufrichtung von links nach rechts assoziierten Belegung $x_i = 0$ erfüllt.

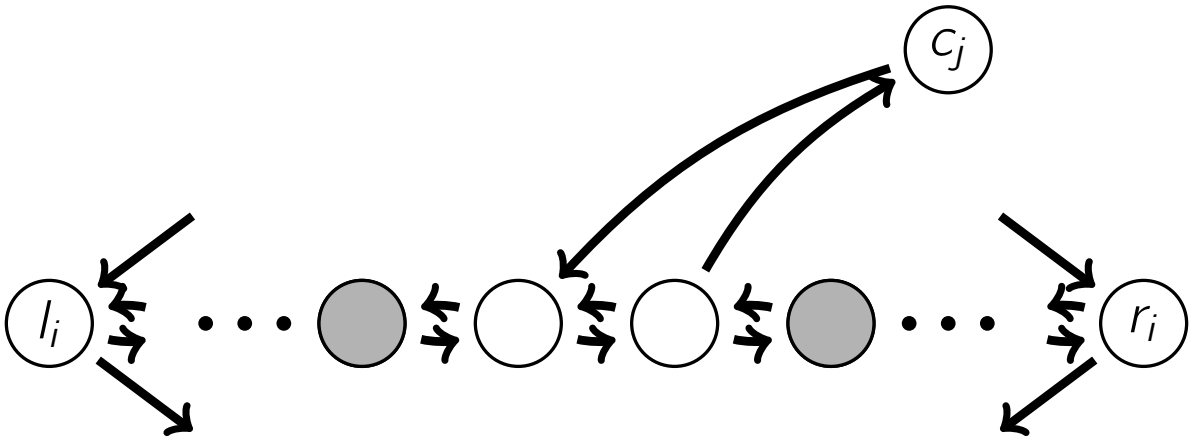
- Wenn ein Klauselknoten c_j aus einem Gadget G_i heraus von rechts nach links durchlaufen wird, so muss nach unserer Konstruktion die Klausel c_j das Literal x_i enthalten.
- Also wird diese Klausel c_j durch die mit der Laufrichtung von rechts nach links assoziierten Belegung $x_i = 1$ erfüllt.

- Also erfüllt die mit der Rundreise assoziierte Wahrheitsbelegung der Variablen die Formel φ.

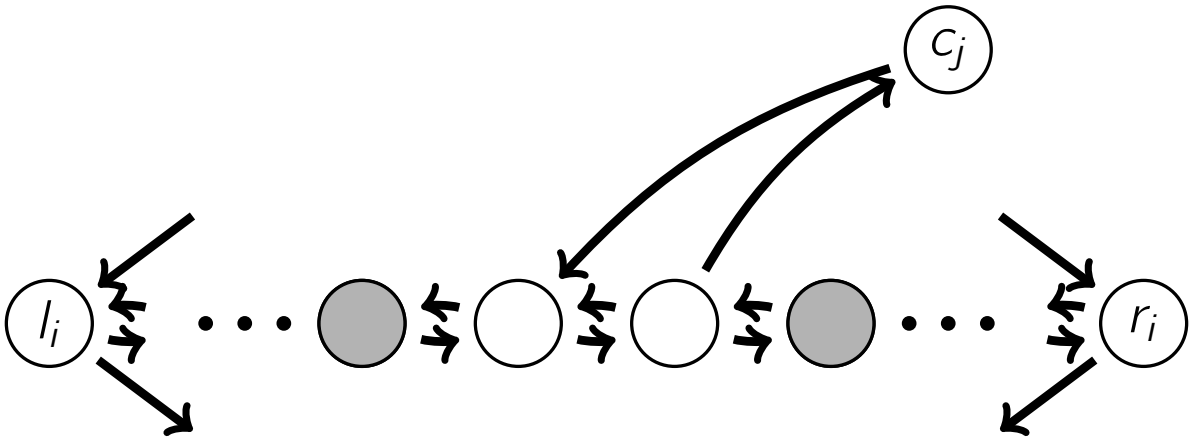
D-Ham-Cycle (6b): Korrektheit

Lemma B: φ erfüllbar $\Rightarrow$ G hat gerichteten Hamiltonkreis

- Eine erfüllende Wahrheitsbelegung der Variablen legt für jedes Diamantengadget $G_1, \ldots, G_n$ fest, ob es von rechts nach links oder von links nach rechts durchlaufen wird.
- Klauselknoten c_j können wir in die Rundreise einbauen, indem wir eine Variable x_i auswählen, die c_j erfüllt, und c_j durch einen kleinen Abstecher vom Diamantengadget G_i aus besuchen.



D-Ham-Cycle (6c): Korrektheit



- Wenn c_j für $x_i = 1$ erfüllt ist, so ist x_i positiv in c_j enthalten. Ein Besuch von c_j beim Durchlaufen des Diamantengadgets G_i von rechts nach links ist möglich.
- Wenn c_j für $x_i = 0$ erfüllt ist, so ist x_i in negierter Form in c_j enthalten. Ein Besuch von c_j beim Durchlaufen des Diamantengadgets G_i von links nach rechts ist möglich.
- Also können alle Klauselknoten in die Rundreise eingebunden werden.

D-Ham-Cycle (7): Schluss

4. (Polynomielle Zeit):
Die Funktion f ist in Polynomialzeit berechenbar.

- Die Konstruktion verwendet n Diamantengadgets mit je $O(m)$ Knoten
- Die Konstruktion verwendet m Klauselknoten

5. (Korrektheit):
f ist Reduktion: $x \in L^* \Leftrightarrow f(x) \in L$

$\varphi \in \text{SAT} \Leftrightarrow f(\varphi) = \langle G \rangle \in \text{D-Ham-Cycle}$

NP-Vollständigkeit von Ham-Cycle (ungerichtet)

Ham-Cycle (1): Definition

Problem: Hamiltonkreis (Ham-Cycle)

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

Frage: Besitzt G einen Hamiltonkreis?

Satz

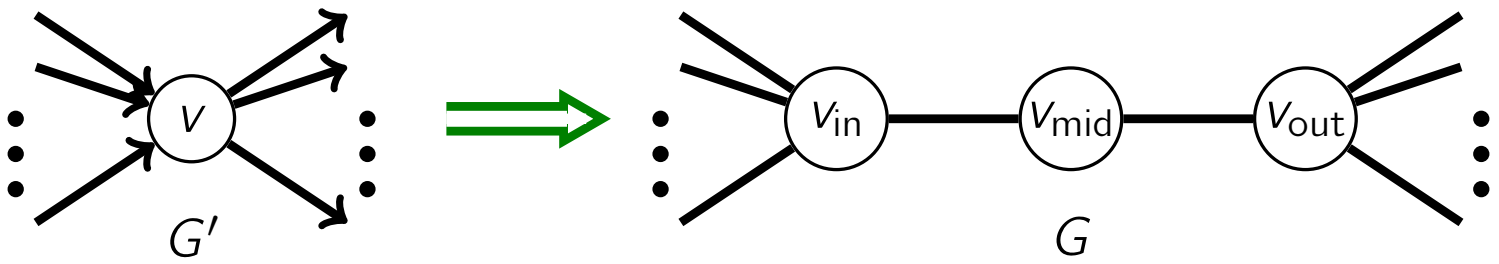
Ham-Cycle ist NP-vollständig.

Beweis:

- Wir zeigen D-Ham-Cycle $\leq_p$ Ham-Cycle
- Es sei $G' = (V', A')$ eine Instanz von D-Ham-Cycle
- Wir konstruieren in polynomieller Zeit einen ungerichteten Graphen $G = (V, E)$, sodass gilt: $G' \in$ D-Ham-Cycle $\Leftrightarrow$ $G \in$ Ham-Cycle

Ham-Cycle (2): Reduktion

- Es sei $G' = (V', A')$ eine Instanz von D-Ham-Cycle
- Der ungerichtete Graph G ensteht aus G' durch lokale Ersetzung:



Interpretation:

- v_{in} ist der Eingangsknoten für v_{mid}
- v_{out} ist der Ausgangsknoten für v_{mid}

Ham-Cycle (3): Korrektheit

G' hat gerichteten Hamiltonkreis $\Leftrightarrow$ G hat Hamiltonkreis

(A) Jeder Hamiltonkreis in G' kann offensichtlich in einen Hamiltonkreis in G transformiert werden.

(B) Wie sieht es mit der Umkehrrichtung aus?

- Jeder Hamiltonkreis in G besucht den Knoten v_{mid} zwischen den beiden Knoten v_{in} und v_{out}
- Entweder: $v_{\text{in}} - v_{\text{mid}} - v_{\text{out}}$ Oder: $v_{\text{out}} - v_{\text{mid}} - v_{\text{in}}$
- Von v_{out} aus kann man nur Knoten vom Typ u_{in} erreichen (und dazu muss der gerichtete Graph die entsprechende gerichtete Kante von v nach u enthalten)
- Daher kann jeder Hamiltonkreis in G in einen gerichteten Hamiltonkreis in G' übersetzt werden.

Ham-Cycle (4): Übung

Übung

Zeigen Sie: Ham-Cycle $\leq_p$ D-Ham-Cycle

Hinweis: Verwenden Sie lokale Ersetzungen

NP-Vollständigkeit des TSP

TSP (1): Definitionen

Problem: Travelling Salesman Problem (TSP)

Eingabe: Städte $1, \ldots, n$; Distanzen $d(i,j)$; eine Zahl γ

Frage: Gibt es eine Rundreise (TSP-Tour) mit Länge höchstens γ?

Zwei Spezialfälle:

Problem: Δ-TSP

Eingabe: Städte $1, \ldots, n$; symmetrische Distanzen $d(i,j)$ mit Dreiecksungleichung $d(i,j) \leq d(i,k) + d(k,j)$; eine Zahl γ

Frage: Gibt es eine Rundreise (TSP-Tour) mit Länge höchstens γ?

Problem: $\{1,2\}$-TSP

Eingabe: Städte $1, \ldots, n$; symmetrische Distanzen $d(i,j) \in \{1,2\}$; eine Zahl γ

Frage: Gibt es eine Rundreise (TSP-Tour) mit Länge höchstens γ?

TSP (2): Beweis der NP-Schwere

Satz

TSP und Δ-TSP und $\{1,2\}$-TSP sind NP-schwer.

- Es genügt zu zeigen, dass $\{1,2\}$-TSP NP-schwer ist.
- Wir zeigen: Ham-Cycle $\leq_p$ $\{1,2\}$-TSP

- Aus einem ungerichteten Graphen $G = (V, E)$ für Ham-Cycle konstruieren wir eine TSP Instanz.
- Jeder Knoten $v \in V$ wird zu einer Stadt
- Der Abstand zwischen Stadt u und Stadt v beträgt

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

- Wir setzen $\gamma := |V|$
- Der Graph G hat genau dann einen Hamiltonkreis, wenn die konstruierte TSP Instanz eine Tour mit Länge $\leq \gamma$ hat.

Landkarte mit Karp's 20 Reduktionen