Anmerkungen zu P

- Die meisten praktisch interessanten Probleme in P lassen sich in $O(n^3)$ Schritten oder schneller lösen.
- Das Laufzeitverhalten polynomieller Algorithmen lässt sich häufig weiter verbessern (Beispiel: Matrizenmultiplikation).

- Wenn jede Gleichungsanwendung in CE-S auf dem Computer polynomiellen Aufwand hat, können wir Gleichungsanwendung als konstant zählen, ohne dabei die Klasse P zu verlassen.

  Aber:

  Es kann sich bei einer CE-S-Spezifikation ein anderer Aufwand ergeben (als bei einem entsprechenden Computer-Programm), wenn der Aufwand einer Gleichungsanwendung nicht konstant ist.

- Gleichungsanwendungen haben nicht immer konstanten Aufwand.

Ein Problem in NP

- SAT (Erfüllbarkeitsproblem der Aussagenlogik)
  - Eingabe: Aussagenlogische Formel f.
  - Ausgabe: T gdw eine Belegung der Variablen in f mit 1 oder 0 existiert, so dass f gilt.
  - Lösung 1: Probiere alle 2^k Belegungen $\in O(2^k)$
  - Lösung 2: Rate richtige Belegung $\in O(k)$

Anwendungsgebiete: Entwurf und Analyse von Schaltkreisen, Model Checkers, Robotik, . . .

Aussagenlogische Formel

Sei Var eine Menge von Variablen.

Menge WFF aller aussagenlogischen Formeln

1. $v \in Var \Longrightarrow v \in WFF$
2. $S_1, S_2 \in WFF \Longrightarrow$
   $(S_1 \vee S_2), (S_1 \wedge S_2), \neg S_1 \in WFF$

Beispiel

Ein Gerät mit vier $0/1$-Schaltern befindet sich bei folgenden Schalterstellungen in einem betriebssicheren Zustand:

1. Wenn A und B gleich 0, dann C gleich 1
2. A oder C gleich 1
3. A, B, C gleich 1 oder B, C, D gleich 1 oder A, D gleich 1

$P = NP$-Problem

- Lässt sich das Erfüllbarkeitsproblem der Aussagenlogik deterministisch in polynomieller Zeit lösen?
- Entsprechende Frage für viele praktisch relevante Probleme (Maschinenbelegung, Tourenplanung, Handelsreisende, Färbung, Lagerhaltung, . . .)

---

* P: alle polynomiell lösbaren (Entscheidungs-)Probleme
NP: alle nichtdeterministisch polynomiell lösbaren (Entscheidungs-)Probleme

Anmerkungen zu NP

- Zur Lösung von Problemen aus NP muss meist eine Anzahl von$Lösungskandidaten$ durchsucht werden.
- Diese Anzahl steigt mit wachsender Eingabe exponentiell an.
- Ein nichtdeterministischer Algorithmus löst diese Probleme in polynomieller Zeit durch Raten einer richtigen Lösung.
- Die Überprüfung, ob ein gegebener $Lösungskandidat$ eine korrekte Lösung ist, kann mit einem deterministischen Algorithmus in $O(n^k)$ Schritten gelöst werden.
- Es wird allgemein vermutet, dass NP größer als P ist.

Probleme in NP

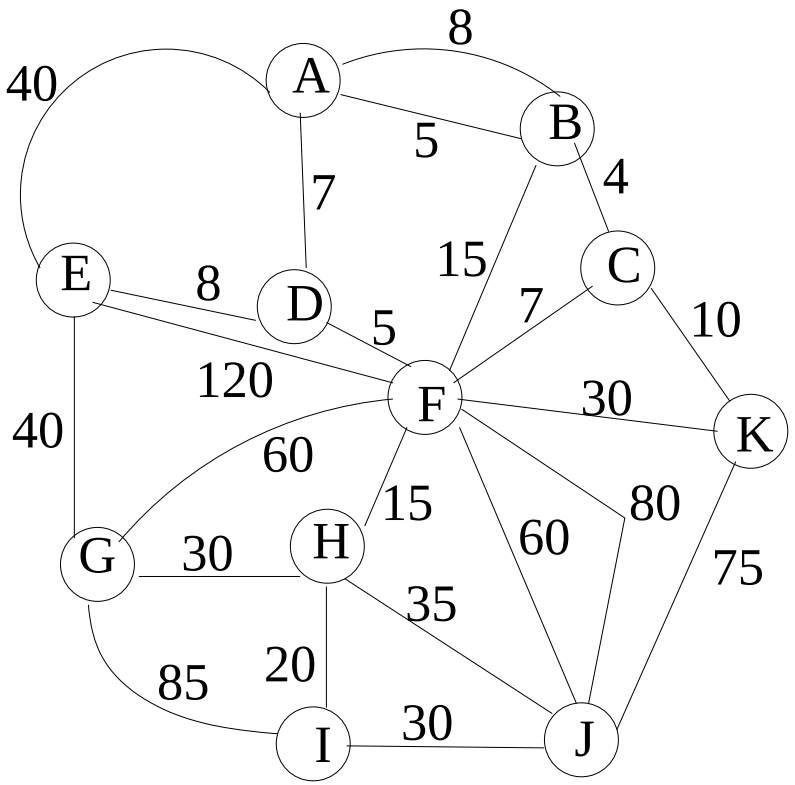
- 3SAT (Spezialfall von SAT)
  - Eingabe: Aussagenlogische Formel f der Form

$$c_1 \wedge c_2 \wedge \cdots \wedge c_n,$$

    so dass
    - c_i: Klausel der Form $L_1 \vee L_2 \vee L_3$,
    - L_j: Literal der Form x oder $\neg x$
    - x: Variable.
  - Ausgabe: T gdw eine Belegung der Variablen in f mit 1 oder 0 existiert, so dass f gilt.

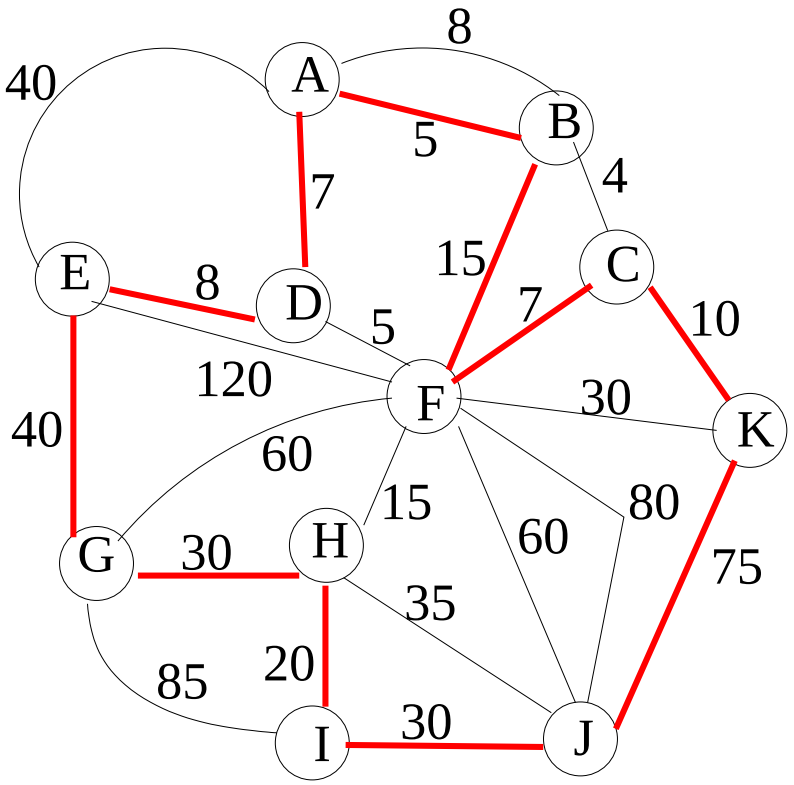
- TSP (Traveling Salesperson Problem, Entscheidungsproblemvariante)
  - Eingabe: Ungerichteter Graph G mit natürlichen Zahlen als Kantenmarkierungen und eine Zahl k.

  Beispiel für G:



- Ausgabe: T gdw G einen Rundweg besitzt, der jeden Knoten genau einmal besucht und höchstens k lang ist.

  Beispiel: T für k $= 250$



Rundweg: 247

Ordnung auf NP durch Reduktion

Eine Reduktion von

$$dp_1 \colon A_1^* \to BOOL \quad \text{auf} \quad dp_2 \colon A_2^* \to BOOL$$

ist eine CE-S-Operation $red \colon A_1^* \to A_2^*$, so dass

1. $T^{red} \in O(n^l)$ für ein $l \in \mathbb{N}$
2. für alle $w \in A_1^*$ gilt: $dp_1(w) = dp_2(red(w))$.

Schreibweise: $dp_1 \leq dp_2$

Theorem

$dp_1 \leq dp_2$ und $dp_2 \in P$ impl. $dp_1 \in P$.

NP-Vollständigkeit

Definition

$dp_0 \in NP$ heißt NP-vollständig, falls $dp \leq dp_0$ für alle $dp \in NP$.

Theorem

dp_0 NP-vollständig und $dp_0 \in P$ impl. $NP \subseteq P$.

Beweis der NP-Vollständigkeit eines Problems dp mittels Reduktion

Theorem

dp ist NP-vollständig.

Beweis:

1. Zeige, dass dp in NP liegt.
2. Wähle ein NP-vollständiges Problem und reduziere es auf dp.

Anmerkung: Ist nur möglich, wenn man bereits ein NP-vollständiges Problem hat.

NP-vollständige Probleme

Theorem (Cook und Levin)
Das Erfüllbarkeitsproblem der Aussagenlogik ist NP-vollständig.
Beweisidee: Reduziere jedes Problem aus NP auf SAT.

Theorem
3SAT ist NP-vollständig.
Beweisidee: Reduziere SAT auf 3SAT.

Theorem

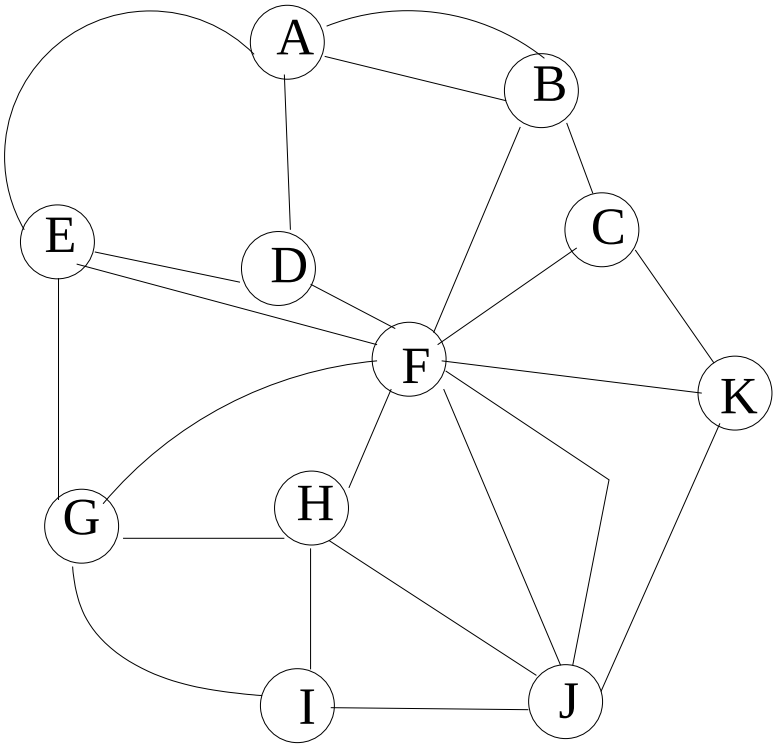
TSP ist NP-vollständig.

Beweisidee: Reduziere HC auf TSP (HC ist ein Spezialfall von TSP und NP-vollständig, wie wir noch sehen werden...)

HC

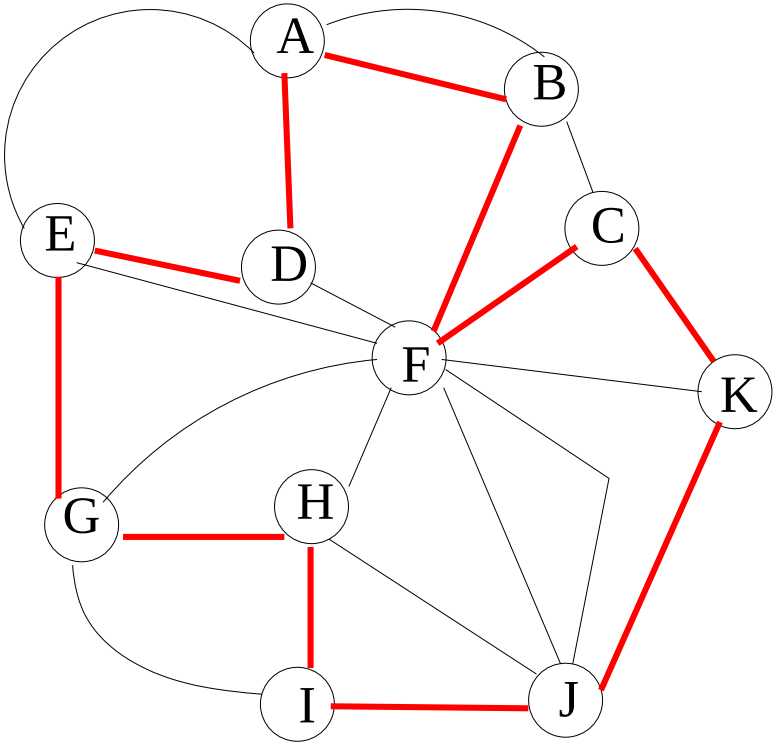
- HC (Hamiltonian Circuit Problem)
  - Eingabe: Ein ungerichteter Graph G.

    Beispiel:



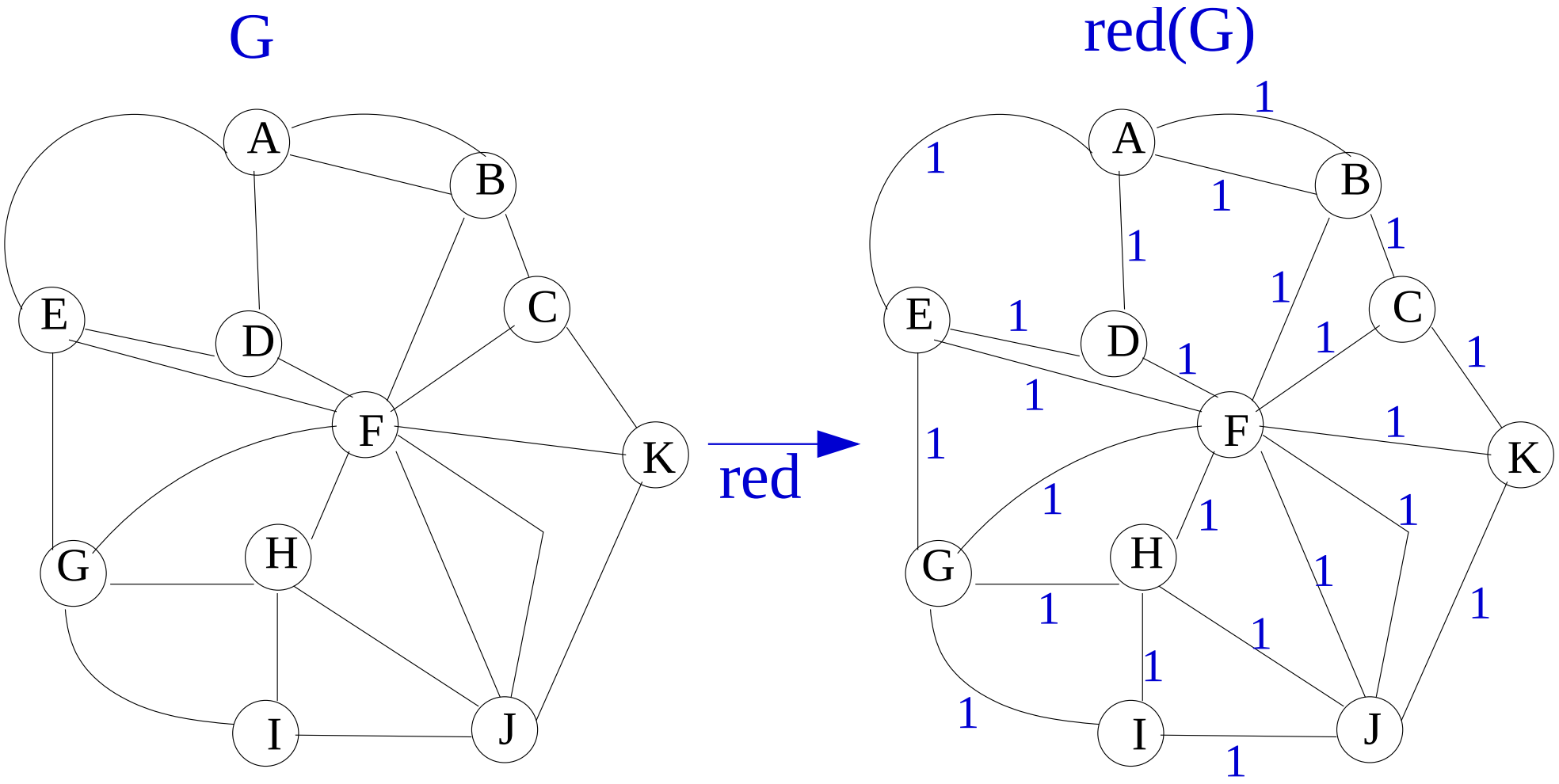
- Ausgabe: T gdw G einen Rundweg besitzt, der jeden Knoten genau einmal besucht.

  Beispiel:



Beobachtung: HC ist in NP.

Reduktion von HC auf TSP



Theorem

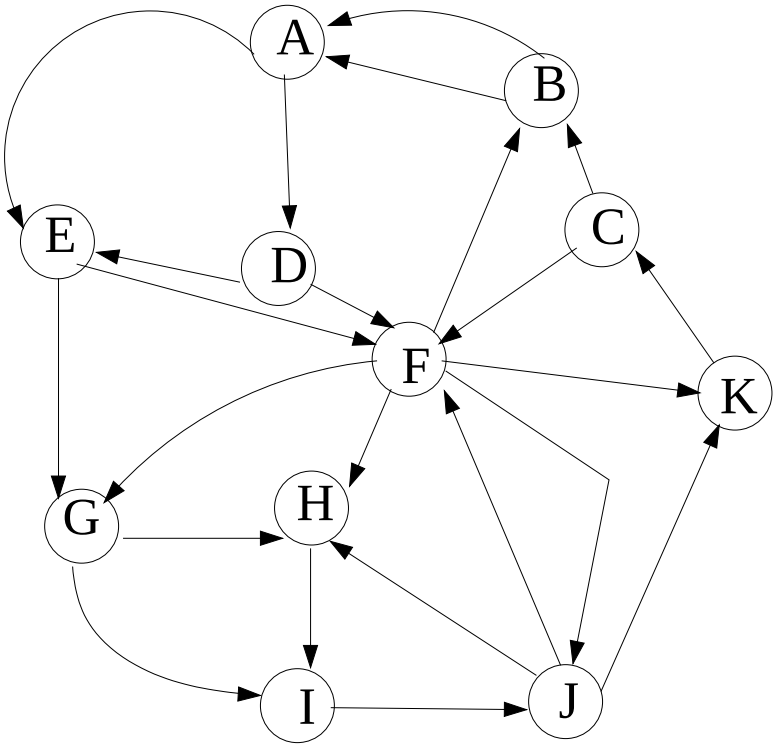
HC ist NP-vollständig.

Beweisidee: Reduziere DHC auf HC. (DHC ist HC für gerichtete Graphen und NP-vollständig, wie wir noch sehen werden...)

DHC

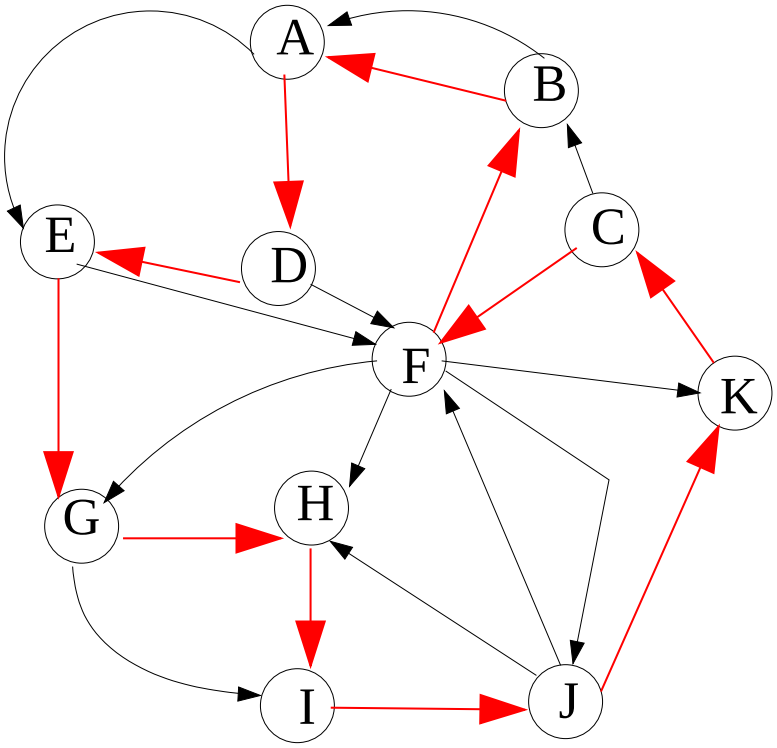
- DHC (Directed Hamiltonian Circuit Problem)
  - Eingabe: Ein gerichteter Graph G.

    Beispiel:



- Ausgabe: T gdw G einen Rundweg (in Pfeilrichtung) besitzt, der jeden Knoten genau einmal besucht.

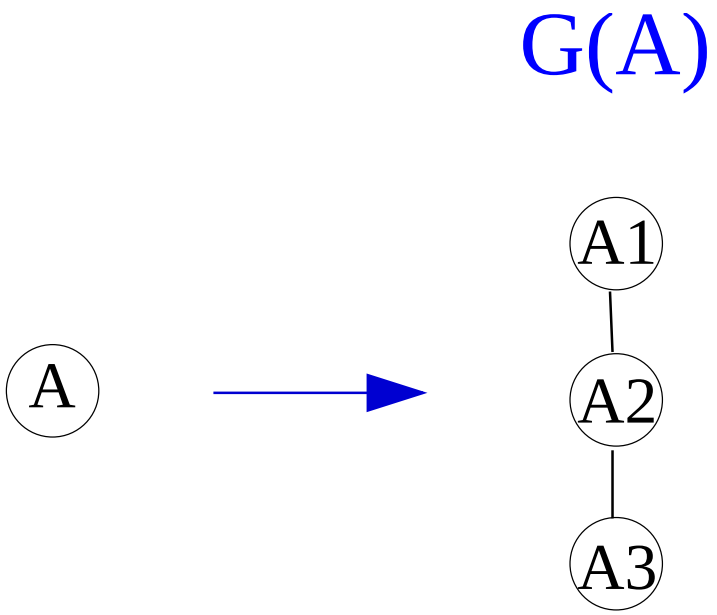
  Beispiel:



Beobachtung: DHC ist in NP.

Reduktion von DHC auf HC

1. Für jeden Knoten in G konstruiere drei miteinander verbundene Knoten wie folgt:



2. Für jede Kante von A nach B ziehe eine Kante in $red(G)$ von $A3$ nach $B1$.

Skizze