16. december

Resume sidste gang

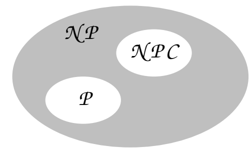
- Abstrakt problem, konkret instans, afgørlighedsproblem
- Effektiv kodning (pol. relateret til binær kodning)
- Sprog L : mængden af instanser for et afgørligheds-problem hvor svaret er 1
- $\mathcal{P} = \{L : L$ genkendes af en algoritme i polynomiel tid$\}$
- $\mathcal{NP} = \{L : L$ verificeres af en polynomieltids algoritme$\}$
- $\mathcal{P} \subseteq \mathcal{NP}$
- Q er $\mathcal{NP}$-fuldstændig hvis og kun hvis $Q \in \mathcal{NP}$ og $\forall R \in \mathcal{NP} : R \leq_{pol} Q$
- CIRCUIT-SAT er $\mathcal{NP}$-fuldstændig

Essentielle spørgsmål: $\mathcal{NP} = \mathcal{P}$ **?**

Et problem Q kaldes $\mathcal{NP}$-fuldstændigt $\Leftrightarrow$

1 $Q \in \mathcal{NP}$

2 $\forall R \in \mathcal{NP} : R \leq_{pol} Q$



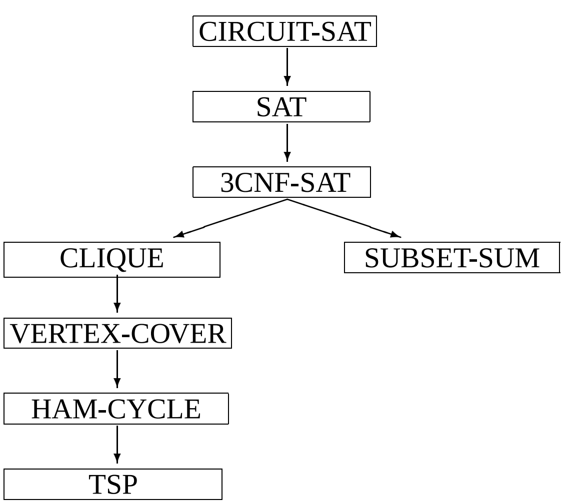
Sætning

Hvis der findes et $\mathcal{NP}$-fuldstændigt problem som er løseligt i polynomiel tid, så er $\mathcal{NP} = \mathcal{P}$.

Hvis et problem i $\mathcal{NP}$ ikke kan løses i polynomiel tid så kan ingen $\mathcal{NP}$-fuldstændige problemer løses i polynomiel tid.

Oversigt, idag

Bevise at en række problemer er $\mathcal{NP}$-fuldstændige.
Yderligere eksempler kan findes i Garey and Johnson



- Beviser ikke (2) fra grunden, men bruger reduktion
- CIRCUIT-SAT har givet os ”foden inden døre”
- men CIRCUIT-SAT er for generel

Hjælpesætning

- Hvis L er et sprog hvor $L' \leq_{pol} L$ for et problem $L' \in \mathcal{NPC}$ så er L $\mathcal{NP}$-hårdt.
- Hvis endvidere $L \in \mathcal{NP}$ så $L \in \mathcal{NPC}$

Teknik til bevis af $L \in \mathcal{NPC}$

1 Bevis at $L \in \mathcal{NP}$ (dvs kan verificeres i poly. tid)

2 Vælg et kendt $\mathcal{NP}$-fuldstændigt problem L'

3 Beskriv en algoritme f som afbilder $L' \mapsto L$

4 Bevis at f opfylder $x \in L' \Leftrightarrow f(x) \in L$ for alle $x \in \{0,1\}^*$

5 Bevis at f kører i polynomiel tid.

Opfyldning af logiske formler

En logisk formel opbygges af

- boolske variable: $x_1, \ldots, x_n$
- boolske operatorer:

$\neg$	NOT
$\wedge$	AND
$\vee$	OR
$\Rightarrow$	medfører
$\Leftrightarrow$	hvis og kun hvis

- paranteser

Eksempel

$$\phi = ((x_1 \Rightarrow x_2) \vee \neg((\neg x_1 \Leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$$

Opfyldende tildeling af sandhedsværdier:

$$x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1$$

Afgørlighedsproblem

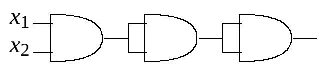
$$\text{SAT} = \{< \phi >: \phi \text{ kan opfyldes}\}$$

SAT er $\mathcal{NP}$-fuldstændig

Bevisets gang

1. Bevis at SAT $\in \mathcal{NP}$ (dvs kan verificeres i poly. tid)
2. Vælg et kendt $\mathcal{NP}$-fuldstændigt problem L'
3. Beskriv en algoritme f som afbilder CIRCUIT-SAT $\mapsto$ SAT
4. Bevis at f opfylder $x \in$ CIRCUIT-SAT $\Leftrightarrow f(x) \in$ SAT for alle $x \in \{0,1\}^*$
5. Bevis at f kører i polynomiel tid.

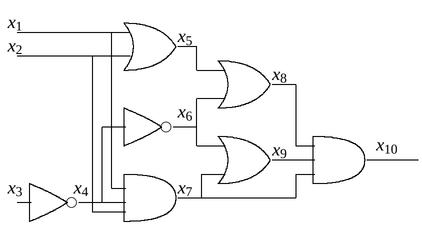
Naiv algoritme f udtrykker output af gate ved input



Polynomiel algoritme f

- Sæt boolske variable på alle ledninger
- Opskriv formel som viser sammenhæng mellem indgange og udgang for hver formel som kræver output = 1.

SAT er $\mathcal{NP}$-fuldstændig



Tilhørende formel

$$\begin{aligned}
\phi = x_{10} \; & \wedge \; (x_4 \Leftrightarrow \neg x_3) \\
& \wedge \; (x_5 \Leftrightarrow (x_1 \vee x_2)) \\
& \wedge \; (x_6 \Leftrightarrow \neg x_4) \\
& \wedge \; (x_7 \Leftrightarrow (x_1 \wedge x_2 \wedge x_4)) \\
& \wedge \; (x_8 \Leftrightarrow (x_5 \vee x_6)) \\
& \wedge \; (x_9 \Leftrightarrow (x_6 \vee x_7)) \\
& \wedge \; (x_{10} \Leftrightarrow (x_7 \wedge x_8 \wedge x_9))
\end{aligned}$$

3CNF-SAT er $\mathcal{NP}$-fuldstændig

CNF er Konjunktiv Normal Form

- Konjunktion: $a \wedge b$
- Disjunktion: $a \vee b$

En formel er på CNF hvis den er en konjunktion af en række disjunktioner af logiske variable eller disses negationer.

- literal: variabel x eller negation $\neg x$.
- clausul: et OR-udtryk af literaler
- CNF-formel: et AND-udtryk af clausuler

3CNF-formel har præcis 3 *forskellige* literaler pr. clausul (bruges ved bevis af CLIQUE)

Eksempel

$$\phi = (x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

Afgørlighedsproblem

$$\text{3CNF-SAT} = \{< \phi >: \phi \text{ kan opfyldes}\}$$

3CNF-SAT er $\mathcal{NP}$-fuldstændig

1 Bevis at 3CNF-SAT $\in \mathcal{NP}$

2 Vælg et kendt $\mathcal{NP}$-fuldstændigt problem SAT

3 Beskriv en algoritme f som afbilder SAT $\mapsto$ 3CNF-SAT

4 Bevis at f opfylder $x \in \text{SAT} \Leftrightarrow f(x) \in \text{3CNF-SAT}$ for alle $x \in \{0,1\}^*$

5 Bevis at f kører i polynomiel tid.

Reduktions-algoritme f: givet logisk udtryk

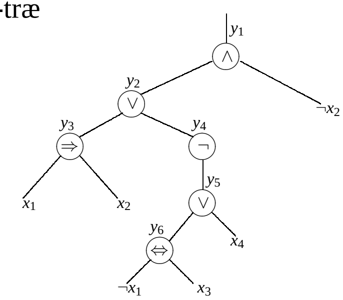
- Byg parse-træ til evaluering — fan-in ≤ 2
- Indfør nye variable y_i til at angive output af hver intern knude
- Opskriv ækvivalent udtryk ϕ' som AND-udtryk
- Hvert AND-udtryk omskrives til CNF
- Clausuler med for få literaler suppleres op til 3

3CNF-SAT er $\mathcal{NP}$-fuldstændig

Givet instans af SAT

$$\phi = ((x_1 \Rightarrow x_2) \vee \neg((\neg x_1 \Leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$$

Generer parse-træ



Opskriv ækvivalent AND-udtryk

$$\begin{aligned}
\phi' &= y_1 \wedge (y_1 \Leftrightarrow (y_2 \wedge \neg x_2)) \\
&\quad \wedge (y_2 \Leftrightarrow (y_3 \vee y_4)) \\
&\quad \wedge (y_3 \Leftrightarrow (x_1 \Rightarrow x_2)) \\
&\quad \wedge (y_4 \Leftrightarrow \neg y_5) \\
&\quad \wedge (y_5 \Leftrightarrow (y_6 \vee x_4)) \\
&\quad \wedge (y_6 \Leftrightarrow (\neg x_1 \Leftrightarrow x_3)) \\
&= \phi'_0 \wedge \phi'_1 \wedge \ldots \wedge \phi'_m
\end{aligned}$$

Hvis hver clausul ϕ'_i er på CNF, så er ϕ' på CNF.

3CNF-SAT er $\mathcal{NP}$-fuldstændig

Sandhedstabel for ϕ'_1

y_1	y_2	x_2	$(y_1 \Leftrightarrow (y_2 \wedge \neg x_2))$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	1

ϕ'_1 er falsk når

$$\begin{aligned}
\neg\phi'_1 &= (y_1 \wedge y_2 \wedge x_2) \\
&\vee (y_1 \wedge \neg y_2 \wedge x_2) \\
&\vee (y_1 \wedge \neg y_2 \wedge \neg x_2) \\
&\vee (\neg y_1 \wedge y_2 \wedge \neg x_2)
\end{aligned}$$

De-Morgans lov

$$\begin{aligned}
\phi'_1 &= (\neg y_1 \vee \neg y_2 \vee \neg x_2) \\
&\wedge (\neg y_1 \vee y_2 \vee \neg x_2) \\
&\wedge (\neg y_1 \vee y_2 \vee x_2) \\
&\wedge (y_1 \vee \neg y_2 \vee x_2)
\end{aligned}$$

Nu er $\phi'' = \phi'_0 \wedge \phi'_1 \wedge \phi'_2 \wedge \ldots \wedge \phi'_m$ på CNF

Hver clausul har *højst* 3 literaler

Alle literaler i clausul er *forskellige*

3CNF-SAT er $\mathcal{NP}$-fuldstændig

Clausuler med 2 literaler:

$$C_i = (l_1 \vee l_2)$$

erstattes af

$$C_i = (l_1 \vee l_2 \vee p) \wedge (l_1 \vee l_2 \vee \neg p)$$

hvor p er en ny variabel.

Clausuler med 1 literal:

$$C_i = (l_1)$$

erstattes af

$$C_i = (l_1 \vee p \vee q) \wedge (l_1 \vee p \vee \neg q) \wedge (l_1 \vee \neg p \vee q) \wedge (l_1 \vee \neg p \vee \neg q)$$

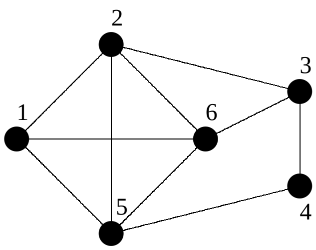
hvor p, q er nye variable.

Klike i en graf

En klike i en ikke-orienteret graf $G = (V, E)$ er en delmængde $V' \subset V$ af knuder således at $(v_i, v_j) \in E$ for alle $v_i, v_j \in V'$.

Afgørlighedsproblem

$$\text{CLIQUE} = \{< G, k >: G \text{ har en klike af størrelse } k\}$$



CLIQUE er $\mathcal{NP}$-fuldstændig

1. Bevis at CLIQUE $\in \mathcal{NP}$
2. Vælg et kendt $\mathcal{NP}$-fuldstændigt problem 3CNF-SAT
3. Beskriv en algoritme f som afbilder 3CNF-SAT $\mapsto$ CLIQUE
4. Bevis at f opfylder $x \in$ 3CNF-SAT $\Leftrightarrow f(x) \in$ CLIQUE for alle $x \in \{0,1\}^*$
5. Bevis at f kører i polynomiel tid.

Reduktions-algoritme f: Givet udtryk på 3CNF

$$\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$$

hver clausul C_r har 3 *forskellige* literaler (l_1^r, l_2^r, l_3^r).

For eksempel

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

som kan tilfredsstilles med $x_1 = 0, x_2 = 0, x_3 = 1$.

CLIQUE er $\mathcal{NP}$-fuldstændig

Motivation

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

- knude repræsenterer literal: Literal sand $\Leftrightarrow$ knude vælges i klike
- sikrer at kun en literal fra hver klausul vælges ved ikke at have kanter mellem disse.
- graf skal være konsistent: ej kant mellem x og $\neg x$.
- klike størrelse er k.
- i udtrykket ϕ skal mindst en literal i hver klausul være sand $\Leftrightarrow$ knude vælges.

CLIQUE er $\mathcal{NP}$-fuldstændig

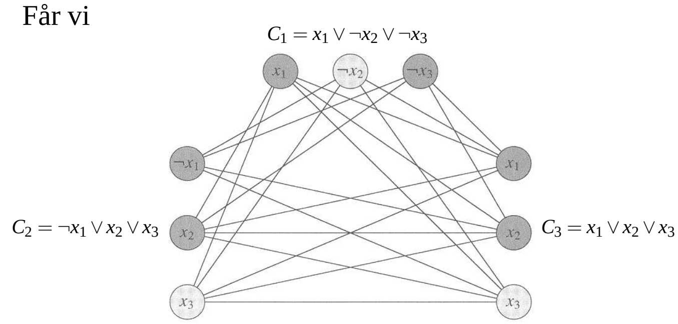
Konstruerer graf

- For hver clausul $C_r = (l_1^r \vee l_2^r \vee l_3^r)$ indfør 3 knuder v_1^r, v_2^r, v_3^r.
- Indfør en kant mellem v_i^r og v_j^s hvis
  - $r \neq s$ dvs. l_i^r og l_j^s er i forskellige clausuler
  - l_i^r er ikke negation af l_j^s (literaler er *konsistente*).
- k (klike størrelse) er antal clausuler.

For det givne eksempel

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

Får vi



Maksimal klike $\neg x_2$, x_3 og x_3. Variable svarende til knuder som ikke er i kliken kan sættes arbitrært i ϕ

CLIQUE er $\mathcal{NP}$-fuldstændig

ϕ er en reduktion:

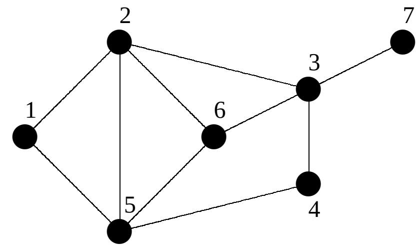
- Antag ϕ har en tilfredsstillende tildeling.
  Så findes k literaler fra hver sin clausul som alle er sande. Vælg disse som klike.
  - Klike størrelse er korrekt
  - Der findes kanter mellem alle par af knuder i kliken
- Antag G har en klike af størrelse k.
  Sæt literal til sand, hvis tilhørende knude er valgt. Hvis nogle variable ikke defineres på denne vis, kan de sættes arbitrært.
  - Ingen konflikter opstår (dvs. f.eks. $x_1 = 1$ og $\neg x_1 = 1$)
  - En literal i hver clausul er sand

Knudeoverdækning

En knudeoverdækning af en ikke orienteret graf $G = (V, E)$ er en delmængde af $V' \subseteq V$ så

$$(u,v) \in E \Rightarrow u \in V' \text{ eller } v \in V' \text{ (eller begge)}$$

Størrelsen af en knudeoverdækning er $k = |V'|$.
Knudeoverdækningsproblemet søger den mindste overdækning i grafen.



Afgørlighedsproblem

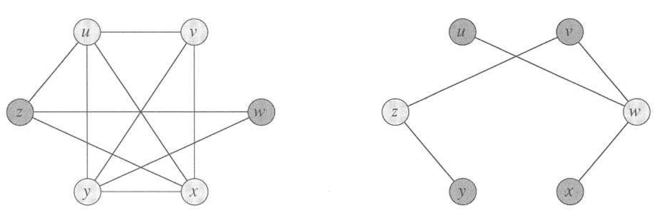
$$\text{VERTEX-COVER} = \{< G, k > : \text{ grafen } G \text{ har en knudeoverdækning af størrelse } k\}$$

VERTEX-COVER er $\mathcal{NP}$-fuldstændig

1. Bevis at VERTEX-COVER $\in \mathcal{NP}$
2. Vælg et kendt $\mathcal{NP}$-fuldstændig problem CLIQUE
3. Beskriv en algoritme f som afbilder
   CLIQUE $\mapsto$ VERTEX-COVER
4. Bevis at f opfylder
   $x \in$ CLIQUE $\Leftrightarrow$
   $f(x) \in$ VERTEX-COVER for alle $x \in \{0,1\}^*$
5. Bevis at f kører i polynomiel tid.

Reduktions-algoritme f : CLIQUE $\Rightarrow$ VERTEX-COVER

$$f :< G, k > \mapsto < \overline{G}, |V| - k >$$



(a) (b)

VERTEX-COVER er $\mathcal{NP}$-fuldstændig

Reduktion

- Antag at G har klike $V' \subseteq V$ med $|V'| = k$.
  Vil vise at $V - V'$ er en knudeoverdækning i $\overline{G}$ af størrelse $|V| - k$
- Antag at $\overline{G}$ har en knudeoverdækning $V' \subseteq V$ hvor $|V'| = |V| - k$.
  Vil vise at $V - V'$ er en klike af størrelse $|V| - |V'| = k$.

Resume

- Teknik til at bevise at problem $P \in \mathcal{NPC}$
- CIRCUIT-SAT er $\mathcal{NP}$-fuldstændig
- SAT er $\mathcal{NP}$-fuldstændig
- 3CNF-SAT er $\mathcal{NP}$-fuldstændig
- CLIQUE er $\mathcal{NP}$-fuldstændig
- VERTEX-COVER er $\mathcal{NP}$-fuldstændig

Overraskende resultater

- Alle ovenstående problemer er "polynomielt ækvivalente"
- Reduktion kan gøres mellem vidt forskellige problemer
- Siden klassen af $\mathcal{NP}$-fuldstændige problemer indeholder tusinder af problemer, har vi grund til at tro at en polynomiel algoritme ikke findes for nogen af dem.

To ekstra beviser

I projektopgave P3 skal bruges at MAX-CUT er $\mathcal{NP}$-fuldstændi

- Viser 3CNF-SAT $\leq_{pol}$ 3CNF-NAESAT
- Viser 3CNF-NAESAT $\leq_{pol}$ MAXCUT

Skematisk bevis, læs selv detaljer

3CNF-NAESAT er $\mathcal{NP}$-fuldstændig

$$\phi = (x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

Afgørlighedsproblem

$$\text{3CNF-NAESAT} = \left\{ <\phi>: \begin{array}{l} \text{hver clausul indeholder} \\ \text{sand og falsk literal} \end{array} \right\}$$

Bemærk at ϕ automatisk bliver sand

Bevis

1 Bevis at 3CNF-NAESAT $\in \mathcal{NP}$

2 Vælg et kendt $\mathcal{NP}$-fuldstændigt problem: 3CNF-SAT

3 Beskriv en algoritme f som afbilder 3CNF-SAT $\mapsto$ 3CNF-NAESAT

4 Bevis at f opfylder $x \in$ 3CNF-SAT $\Leftrightarrow f(x) \in$ 3CNF-NAESAT for alle $x \in \{0,1\}^*$

5 Bevis at f kører i polynomiel tid.

Reduktion

Givet 3CNF-SAT udtryk

$$\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$$

hvor

$$C_i = (l_1^i \vee l_2^i \vee l_3^i)$$

Konstruer 3CNF-NAESAT udtryk

$$\psi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$$

ved at erstatte hver clausul C_i med

$$C_i = (l_1^i \vee l_2^i \vee z_i) \wedge (\neg z_i \vee l_3^i \vee b)$$

hvor z_i er en ny variabel for hver klausul, og b er en ny variabel for hele udtrykket ψ.

Eksempel

Givet 3CNF-SAT udtryk

$$\phi = (x_1 \vee \neg x_4 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

konstruer

$$\begin{aligned} \psi \;=\; & (x_1 \vee \neg x_4 \vee z_1) \wedge (\neg z_1 \vee \neg x_2 \vee b) \wedge \\ & (x_3 \vee x_2 \vee z_2) \wedge (\neg z_2 \vee x_4 \vee b) \wedge \\ & (\neg x_1 \vee \neg x_3 \vee z_3) \wedge (\neg z_3 \vee \neg x_4 \vee b) \end{aligned}$$

$x \in$ **3CNF-SAT** $\Rightarrow f(x) \in$ **3CNF-NAESAT**

Antag at ϕ er sand, udvid løsning til

1 Hvis $l_1^i = l_2^i =$ TRUE så $z_i =$ FALSE

2 Hvis $l_1^i = l_2^i =$ FALSE så $z_i =$ TRUE

3 Hvis $l_1^i \neq l_2^i$ så $z_i =$ FALSE

4 $b =$ FALSE

$$\begin{aligned}\psi \;=\; & (x_1 \vee \neg x_4 \vee z_1) \wedge (\neg z_1 \vee \neg x_2 \vee b) \wedge \\ & (x_3 \vee x_2 \vee z_2) \wedge (\neg z_2 \vee x_4 \vee b) \wedge \\ & (\neg x_1 \vee \neg x_3 \vee z_3) \wedge (\neg z_3 \vee \neg x_4 \vee b)\end{aligned}$$

$x \in$ **3CNF-SAT** $\Leftarrow f(x) \in$ **3CNF-NAESAT**

Givet udtryk

$$\psi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$$

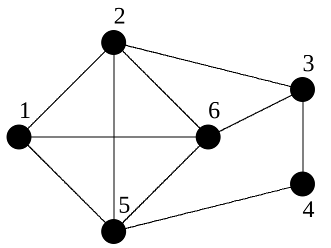
tag variable $x_1, \ldots, x_n$ og tildel dem til ϕ.

- Hvis ϕ er sand så slut
- Hvis ϕ er falsk, så findes klausul $C_i = (l_1^i \vee l_2^i \vee l_3^i)$ hvor $(l_1^i \vee l_2^i \vee l_3^i)$ alle er falske. Da b er falsk, og en af z_i og $\neg z_i$ altid er falsk, fandtes en klausul i Ψ hvor alle literaler var falske.

MAX-CUT

Afgørlighedsproblem

$$\text{MAX-CUT} = \left\{ < G, K >: \begin{array}{l} \text{der findes et cut } S, T \text{ med} \\ \text{mindst } K \text{ kanter mellem } S \text{ og } T \end{array} \right\}$$



Hvis $< G, K >$ er givet ved ovenstående graf, og $K = 6$ findes en ja-instans $S = \{1, 2, 3\}$, $T = \{4, 5, 6\}$.

MAX-CUT er $\mathcal{NP}$-fuldstændig

1 Bevis at MAX-CUT $\in \mathcal{NP}$

2 Vælg et kendt $\mathcal{NP}$-fuldstændigt problem 3CNF-NAESAT

3 Beskriv en algoritme f som afbilder 3CNF-NAESAT $\mapsto$ MAX-CUT

4 Bevis at f opfylder $x \in$ 3CNF-NAESAT $\Leftrightarrow f(x) \in$ MAX-CUT for alle $x \in \{0,1\}^*$

5 Bevis at f kører i polynomiel tid.

Reduktion

Reduktions-algoritme f: Givet udtryk på 3CNF

$$\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$$

hver clausul C_r har 3 *forskellige* literaler (l_1^r, l_2^r, l_3^r).

For eksempel

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

tilfredsstilles med $x_1 = 0, x_2 = 0, x_3 = 1$.

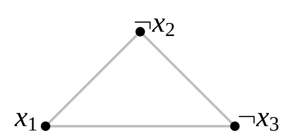
$$\phi = (0 \vee 1 \vee 0) \wedge (1 \vee 0 \vee 1) \wedge (0 \vee 0 \vee 1)$$

Reduktion

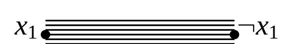
$$\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

- hver variabel x_i repræsenteres af to knuder x_i og $\neg x_i$

- for hver klausul $(x_1 \vee \neg x_2 \vee \neg x_3)$ tegnes trekant



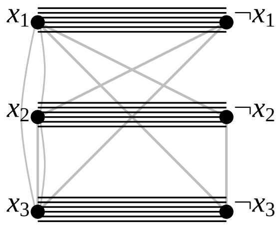
- mellem hvert par af knuder x_i og $\neg x_i$ tegnes $2n_i$ kanter, hvor n_i er antal forekomster af x_i og $\neg x_i$ i ϕ



- cutstørrelse sættes til $K = 8k$ hvor k er antal klausuler

Eksempel på reduktion

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



$K = 8k = 24$

Fortolkning af snit (S,T)

- Knuder i S svarer til at variabel er TRUE
- Knuder i T svarer til at variabel er FALSE
- Vi skal sikre at x_i og $\neg x_i$ er i forskellige mængder
- Hvis en trekant (klausul) splittes op af (S,T) så er en literal sand og en anden falsk
- En trekant (klausul) bidrager med 2 kanter uanset hvordan den splittes op

MAX-CUT medfører 3CNF-NAESAT

Antag at MAX-CUT med $G = (V,E)$ og $K = 8k$ er sand

Første $6k$ kanter

- Vi kan antage at snittet adskiller variable fra deres negation da en literal højst bidrager med n_i kanter i klausulerne, hvilket ikke overstiger antal kanter mellem x_i og $\neg x_i$.
- Antal kanter mellem x_i og $\neg x_i$ for $i = 1,\ldots,m$ er $6k$ da antal literaler er $3k$.

Resterende $2k$ kanter

- Må komme fra klausuler (trekanter)
- Hver klausul bidrager med 2 kanter i snit, så alle må være blevet splittet
- Opsplitning af en trekant betyder at en literal er sand og en literal er falsk

3CNF-NAESAT medfører MAX-CUT

Simpel (prøv selv)