Forelesning 14

Om du står overfor et NP-komplett problem, så er det viktig å kunne bevise det. Vi har sett at alt i NP kan reduseres til CIRCUIT-SAT direkte, men vi kan også vise kompletthet indirekte, ved å redusere videre fra CIRCUIT-SAT!

Pensum

- ☐ 34. NP-completeness: 34.4 og 34.5

Læringsmål

Fra pensumheftet

- [N_1] Forstå hvordan NP-kompletthet kan bevises ved én reduksjon
- [N_2] Kjenne CIRCUIT-SAT, SAT, 3-CNF-SAT, CLIQUE, VERTEX-COVER, HAM-CYCLE, TSP og SUBSET-SUM
- [N_3] Forstå NPC-bevisene for disse
- [N_4] Forstå at det binære ryggsekkproblemet er NPH
- [N_5] Forstå at *lengste enkle vei* er NPH
- [N_6] Kunne konstruere enkle NPC-bevis

Forelesningen filmes

STUDIEBAROMETERET

Forelesning 14

NP-komplette problemer

1. **CIRCUIT-SAT**
2. **SAT**
3. **3-CNF-SAT**
4. **CLIQUE**
5. **VERTEX-COVER**
6. **HAM-CYCLE**
7. **TSP**
8. **SUBSET-SUM**

Litt repetisjon

$$\mathrm{L} \in \mathbf{NPC}$$

Hvordan viser vi at L er **NP**-komplett?

- Vis at $\mathrm{L} \in \mathbf{NP}$

At sertifikat for ja-svar kan verifiseres i pol. tid

- Vis at L $\in$ **NP**
- Velg et kjent **NP**-komplett språk L$'$

- Vis at L $\in$ **NP**
- Velg et kjent **NP**-komplett språk L$'$
- Beskriv en algoritme som beregner en funksjon

$$f : \{0,1\}^* \to \{0,1\}^*$$

som transformerer instanser av L$'$ til instanser av L

Dette er altså reduksjonen fra L$'$ til L, som viser L$' \leqslant_{\mathbf{P}}$ L

- Vis at L $\in$ **NP**
- Velg et kjent **NP**-komplett språk L′
- Beskriv en algoritme som beregner en funksjon

$$f : \{0,1\}^* \to \{0,1\}^*$$

som transformerer instanser av L′ til instanser av L
- Vis at

$$x \in \mathrm{L}' \iff f(x) \in \mathrm{L}\,,$$

for alle $x \in \{0,1\}^*$

Vi må sørge for at vi får samme svar for $f(x)$

- Vis at $\mathrm{L} \in \mathbf{NP}$
- Velg et kjent **NP**-komplett språk L'
- Beskriv en algoritme som beregner en funksjon

$$f : \{0,1\}^* \to \{0,1\}^*$$

  som transformerer instanser av L' til instanser av L
- Vis at

$$x \in \mathrm{L}' \iff f(x) \in \mathrm{L}\,,$$

  for alle $x \in \{0,1\}^*$
- Vis at algoritimen som beregner f har polynomisk kjøretid

En fotnote

om NP-hardhet

- Boka bruker samme type reduksjoner for **NP**-hardhet

Karp-reduksjoner, der $x \in \mathrm{L}' \Leftrightarrow f(x) \in \mathrm{L}$

- Boka bruker samme type reduksjoner for **NP**-hardhet
- Det begrenser oss til beslutningsproblemer . . .

(Siden vi ikke får transformere svaret)

- Boka bruker samme type reduksjoner for **NP**-hardhet
- Det begrenser oss til beslutningsproblemer . . .
- . . . men gir oss altså $\mathbf{NPC} = \mathbf{NP} \cap \mathbf{NPH}$

- Boka bruker samme type reduksjoner for **NP**-hardhet
- Det begrenser oss til beslutningsproblemer . . .
- . . . men gir oss altså $\mathbf{NPC} = \mathbf{NP} \cap \mathbf{NPH}$
- Det er vanlig med mer generelle reduksjoner for **NPH** . . .

(Spesifikt Turing-reduksjoner med polynomisk kjøretid)

- Boka bruker samme type reduksjoner for **NP**-hardhet
- Det begrenser oss til beslutningsproblemer …
- … men gir oss altså $\mathbf{NPC} = \mathbf{NP} \cap \mathbf{NPH}$
- Det er vanlig med mer generelle reduksjoner for **NPH** …
- … men ikke for **NPC**!

Ukjent om man da ender med flere **NP**-komplette problemer!

- Boka bruker samme type reduksjoner for **NP**-hardhet
- Det begrenser oss til beslutningsproblemer …
- … men gir oss altså $\mathbf{NPC} = \mathbf{NP} \cap \mathbf{NPH}$
- Det er vanlig med mer generelle reduksjoner for **NPH** …
- … men ikke for **NPC**!
- Løser man et slikt **NPH**-problem, har man uansett $\mathbf{P} = \mathbf{NP}$

Altså et som er **NP**-hardt under polynomiske Turing-reduksjoner

- Boka bruker samme type reduksjoner for **NP**-hardhet
- Det begrenser oss til beslutningsproblemer …
- … men gir oss altså $\mathbf{NPC} = \mathbf{NP} \cap \mathbf{NPH}$
- Det er vanlig med mer generelle reduksjoner for **NPH** …
- … men <u>ikke</u> for **NPC**!
- Løser man et slikt **NPH**-problem, har man uansett $\mathbf{P} = \mathbf{NP}$

(For **NPH** definert slik er det altså <u>ukjent</u> om $\mathbf{NPC} = \mathbf{NP} \cap \mathbf{NPH}$)

Kort fortalt: Vi bruker ofte begrepet NP-hardhet også om mer generelle problemer

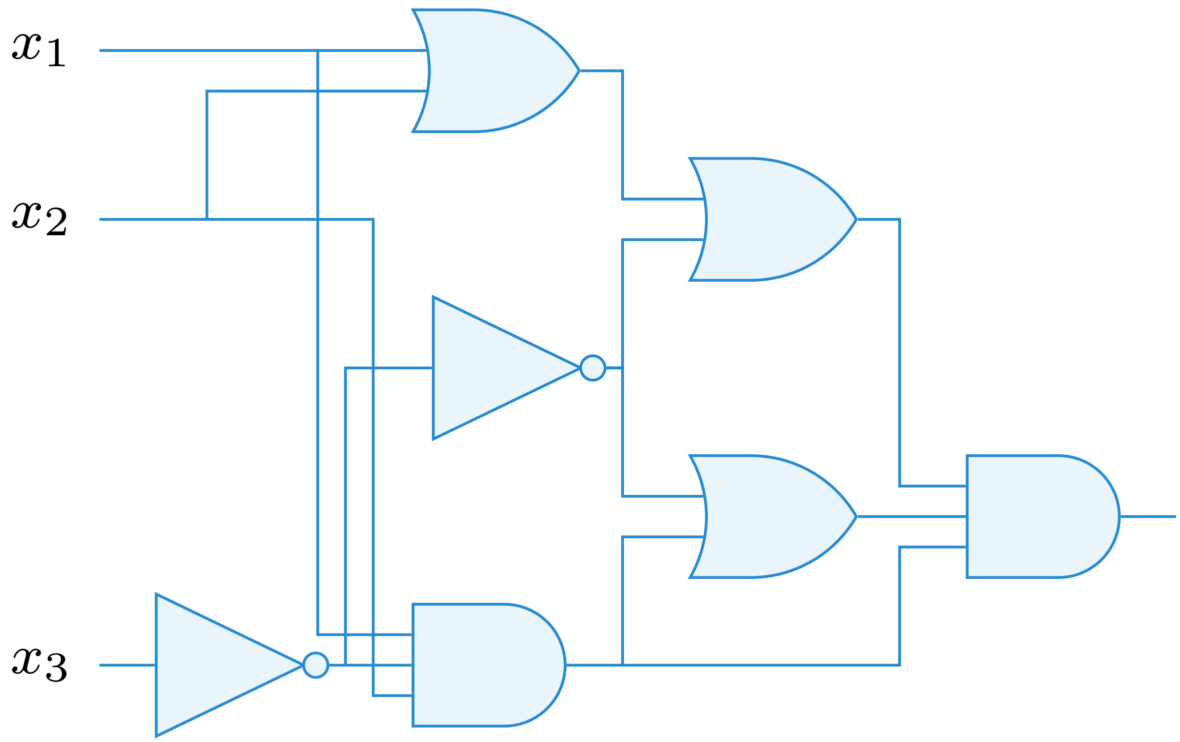
Se delkapittel 34.5.6, «Reduction strategies» for generelle råd

Her ser vi på noen spesifikke eksempler

Repetisjon

1:8

CIRCUIT-SAT



CIRCUIT-SAT

Instans: En krets med logiske porter og én utverdi

Spørsmål: Kan utverdien bli 1?

› Vi har et vilkårlig språk/problem $L \in \mathbf{NP}$

- Vi har et vilkårlig språk/problem L $\in$ **NP**
- Vi vil redusere dette til CIRCUIT-SAT

- Vi har et vilkårlig språk/problem L $\in$ **NP**
- Vi vil redusere dette til CIRCUIT-SAT
- Det eneste vi vet er at $x \in$ L kan verifiseres i polynomisk tid

- Vi har et vilkårlig språk/problem L $\in$ **NP**
- Vi vil redusere dette til CIRCUIT-SAT
- Det eneste vi vet er at $x \in$ L kan verifiseres i polynomisk tid
- Vi simulerer trinnene i verifikasjonsalgoritmen A med kretser!

- Vi har et vilkårlig språk/problem $L \in \mathbf{NP}$
- Vi vil redusere dette til CIRCUIT-SAT
- Det eneste vi vet er at $x \in L$ kan verifiseres i polynomisk tid
- Vi simulerer trinnene i verifikasjonsalgoritmen A med kretser!
- Spørsmålet blir: Kan A (for et eller annet sertifikat) svare 1?

$$x \in \{0, 1\}^*$$

Er x med i språket L? $\Longleftrightarrow$ Kan utverdien bli 1?

$\bullet \longmapsto \bullet$

L er i NP, så . . .

Det finnes en pol. alg. A, som er slik at

› $x \in \mathrm{L}$

nøyaktig når minst én $y \in \{0,1\}^*$ gir

› $\mathrm{A}(x, y) = 1$,

der $|y| = \mathrm{O}(|x|^c)$, for en eller annen c.

Er x med i språket L? $\Longleftrightarrow$ Kan utverdien bli 1?

L er i **NP**, så …

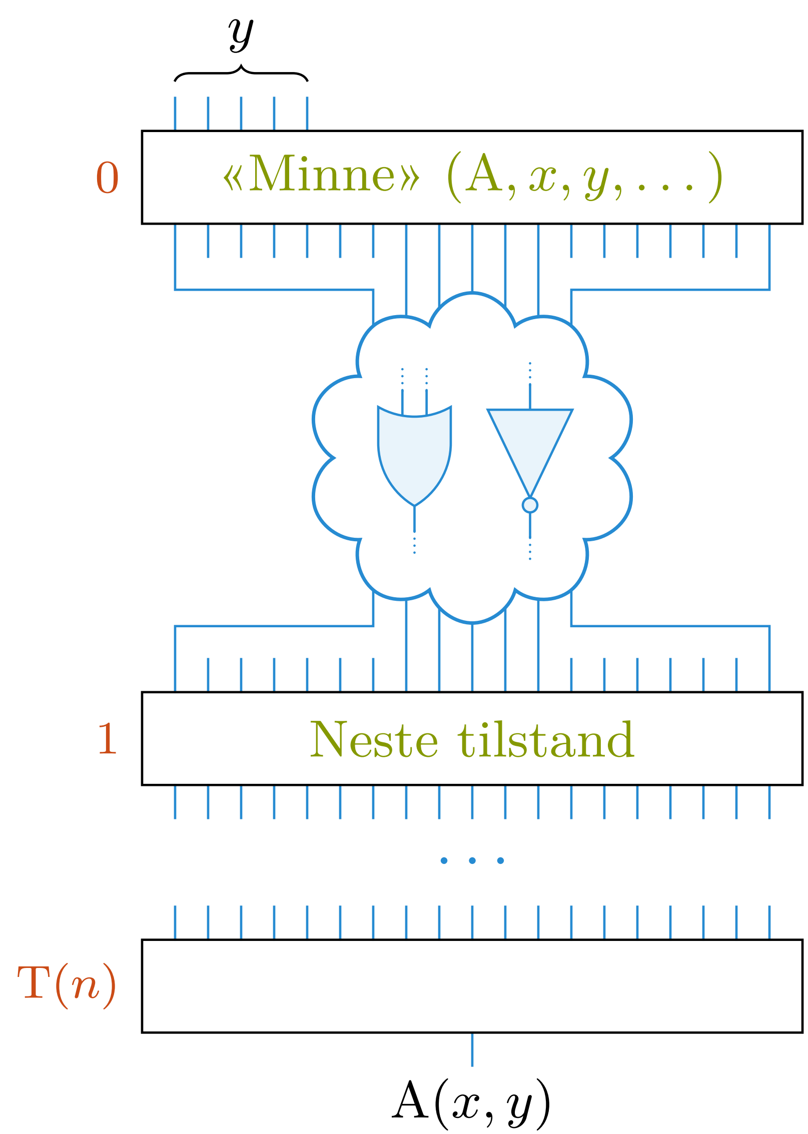
Det finnes en pol. alg. A, som er slik at

› $x \in L$

nøyaktig når minst én $y \in \{0,1\}^*$ gir

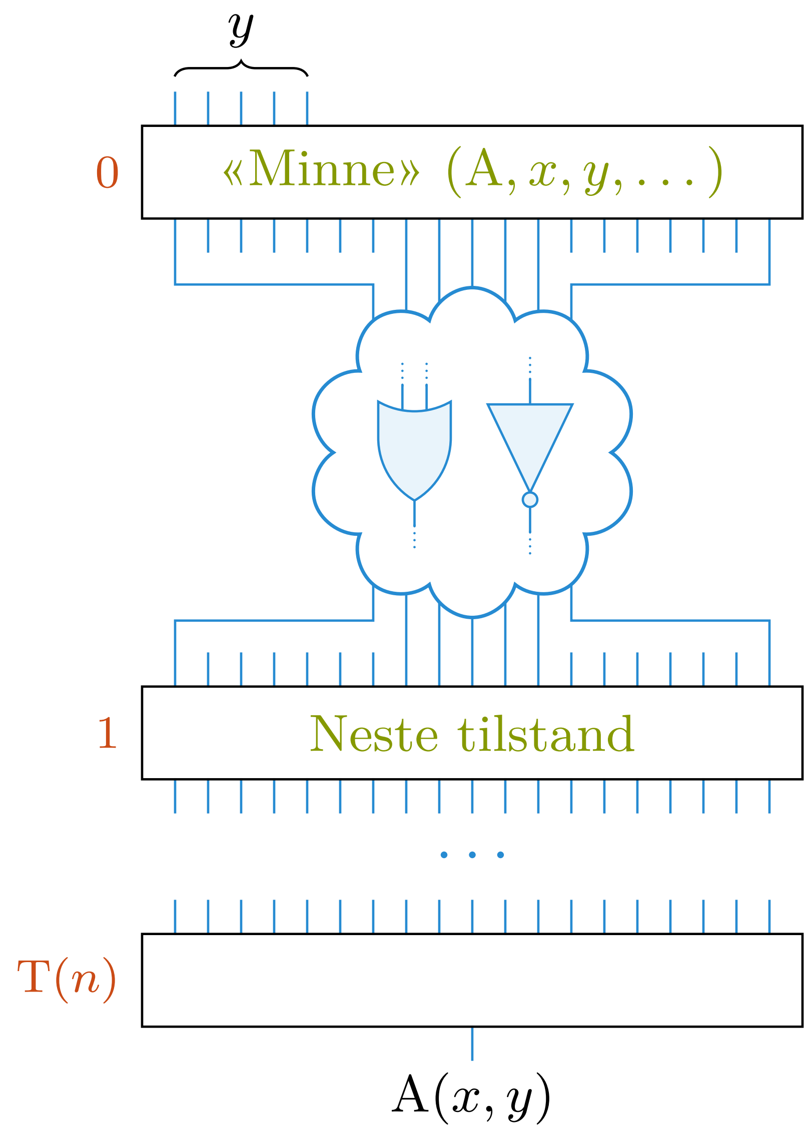
› $A(x,y) = 1$,

der $|y| = O(|x|^c)$, for en eller annen c.



Er x med i språket L? $\iff$ Kan utverdien bli 1?

$$x \in \{0,1\}^*$$



Er x med i språket L? $\iff$ Kan utverdien bli 1?

2:8

SAT

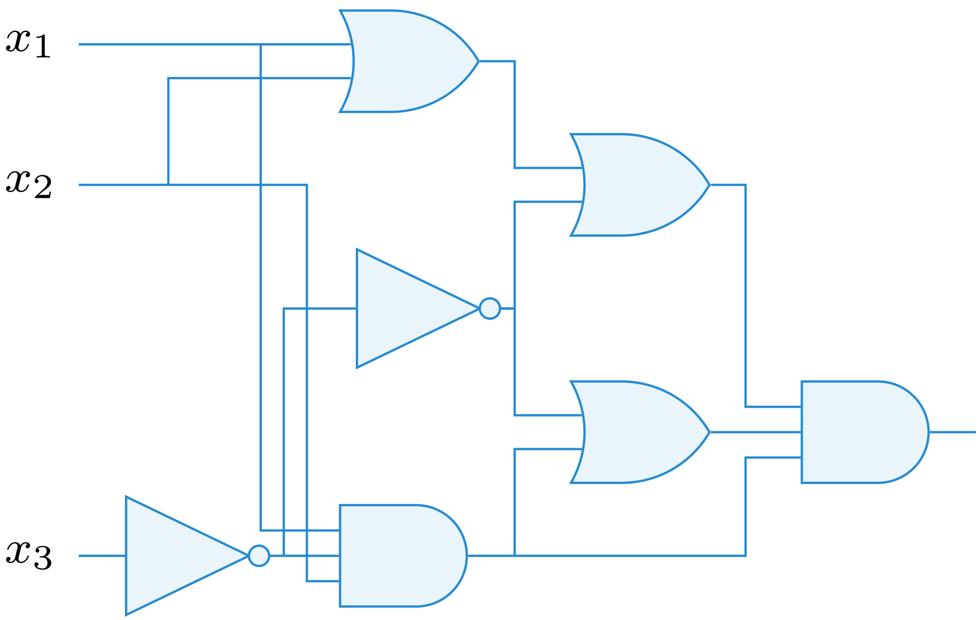
$$\phi = ((x_1 \rightarrow x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$$

SAT
Instans: En logisk formel
Spørsmål: Kan formelen være sann?

- Direkte oversettelse av logisk krets?

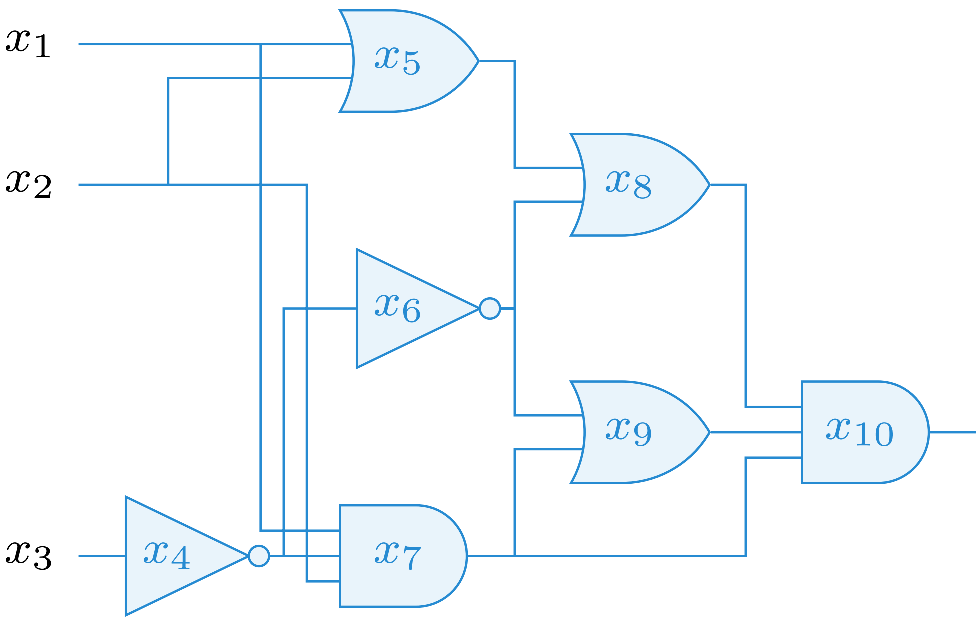
- Direkte oversettelse av logisk krets?
- Kan gi eksponentielt stor formel!

$\bullet \longmapsto \bullet$



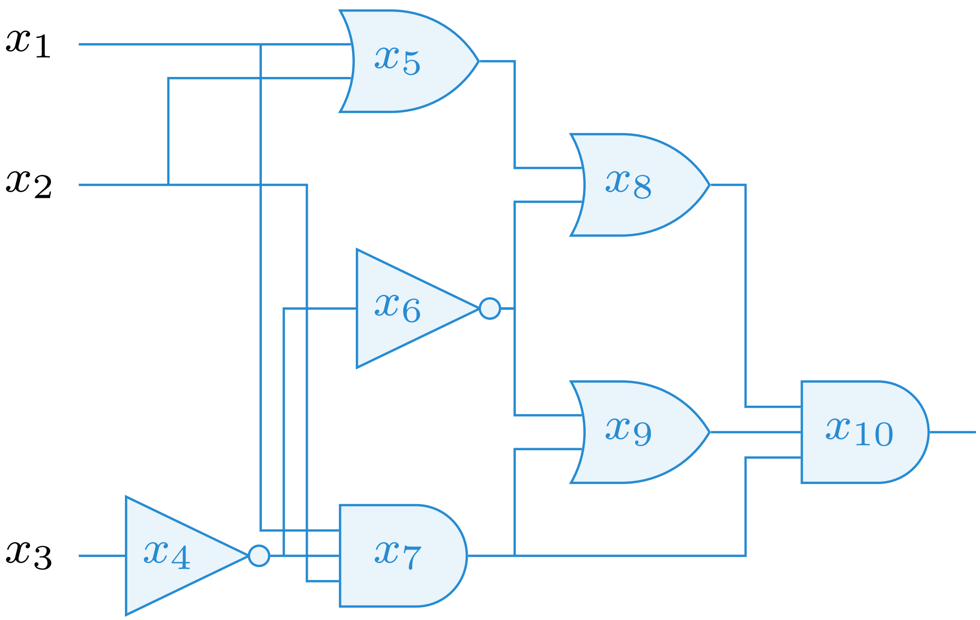
$\phi =$

Kan utverdien bli 1? $\iff$ Kan ϕ være sann?



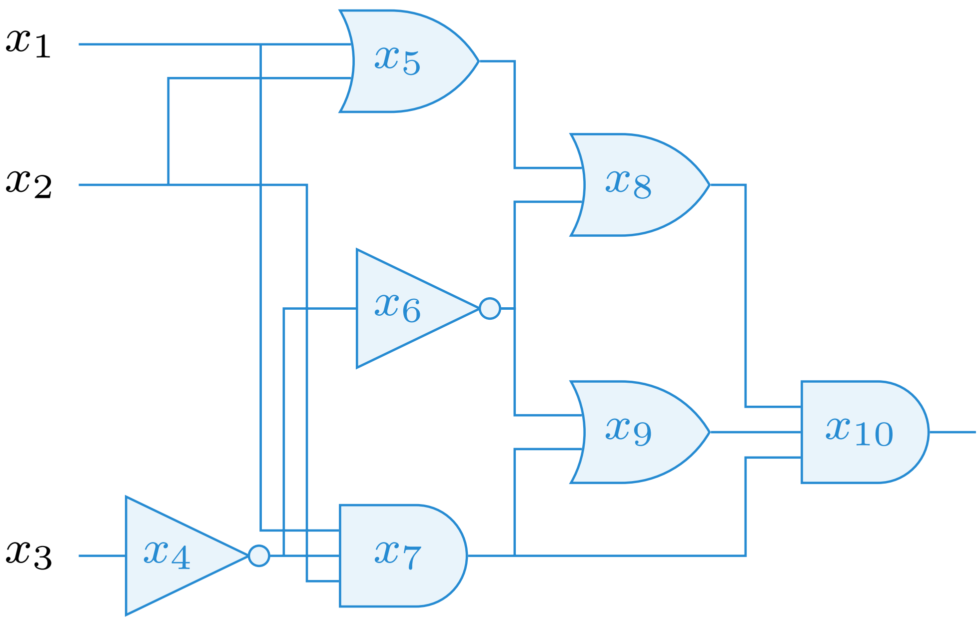
$\phi =$

Kan utverdien bli 1? $\Longleftrightarrow$ Kan ϕ være sann?



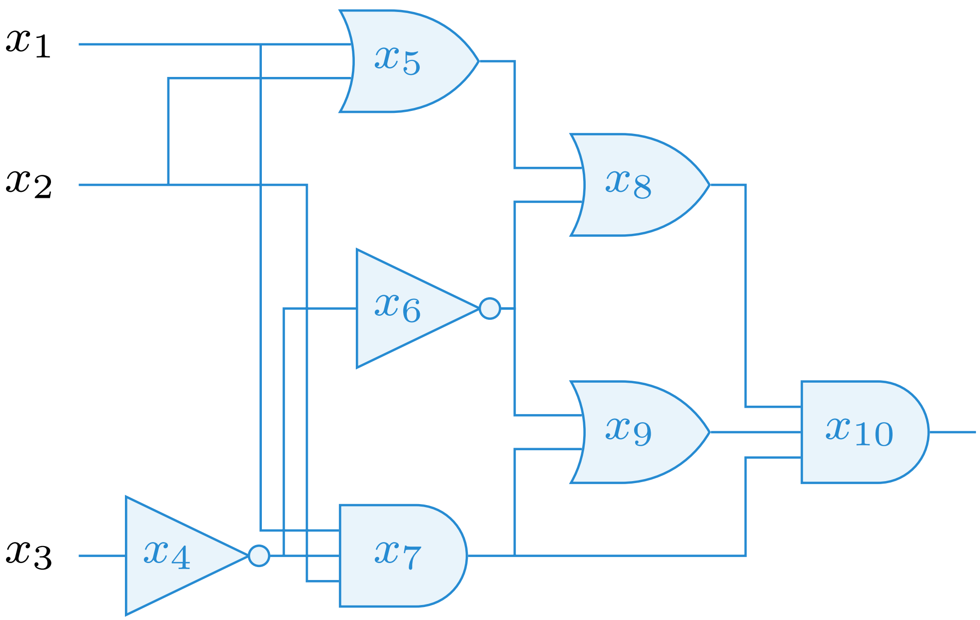
$$\phi = x_{10}$$

Kan utverdien bli 1? $\iff$ Kan ϕ være sann?



$$\begin{aligned}\phi = x_{10} &\wedge (x_4 \leftrightarrow \quad) \\ &\wedge (x_5 \leftrightarrow \quad) \\ &\wedge (x_6 \leftrightarrow \quad) \\ &\wedge (x_7 \leftrightarrow \quad) \\ &\wedge (x_8 \leftrightarrow \quad) \\ &\wedge (x_9 \leftrightarrow \quad) \\ &\wedge (x_{10} \leftrightarrow \quad)\end{aligned}$$

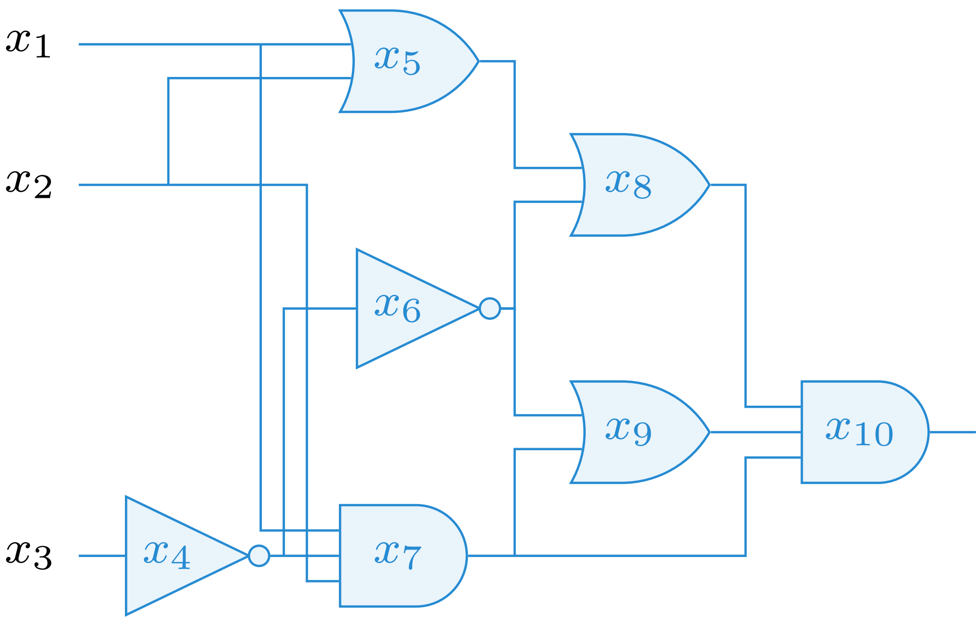
Kan utverdien bli 1? $\iff$ Kan ϕ være sann?



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

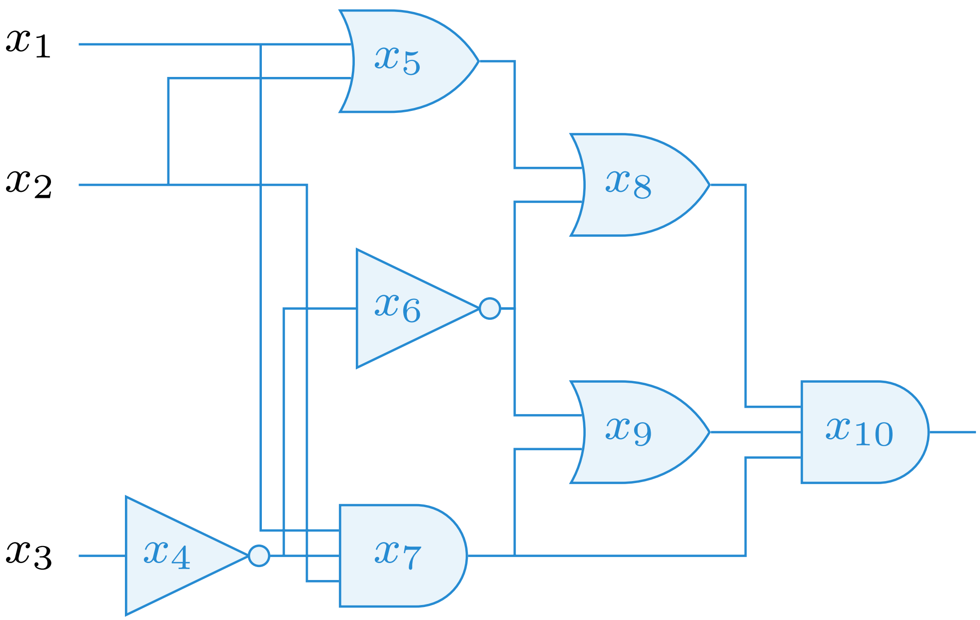
Kan utverdien bli 1? $\Longleftrightarrow$ Kan ϕ være sann?

$\bullet \mapsto \bullet$



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

Kan utverdien bli 1? $\Longleftrightarrow$ Kan ϕ være sann?



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

Kan utverdien bli 1? $\iff$ Kan ϕ være sann?

3:8

3-CNF-SAT

CNF-form

Konjunksjon av disjunksjoner av literaler

CNF-form

Konjunksjon av disjunksjoner av literaler

For eksempel $(x \vee y) \wedge (\neg z \vee \neg y \vee w \vee u) \vee (\neg v \vee \neg x \vee \neg z)$

3-CNF-form

Konjunksjon av disjunksjoner av 3 literaler

3-CNF-form

Konjunksjon av disjunksjoner av 3 literaler

For eksempel $(x \vee y \vee \neg z) \wedge (\neg y \vee w \vee u) \vee (\neg v \vee \neg x \vee \neg z)$

$$\begin{aligned}\phi = {} & (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ & (\neg x_1 \vee x_2 \vee x_3) \wedge \\ & (x_1 \vee x_2 \vee x_3)\end{aligned}$$

3-CNF-SAT
Instans: En logisk formel på 3-CNF-form
Spørsmål: Kan formelen være sann?

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse
- ϕ' er en konjunksjon av ledd med maks 3 literaler

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse
- ϕ' er en konjunksjon av ledd med maks 3 literaler
  - Dvs.: de to argumentene, samt resultatet av operatoren

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse
- ϕ' er en konjunksjon av ledd med maks 3 literaler
  - Dvs.: de to argumentene, samt resultatet av operatoren
- Hvert ledd gjøres om til CNF vha. en sannhetstabell

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse
- ϕ' er en konjunksjon av ledd med maks 3 literaler
  - Dvs.: de to argumentene, samt resultatet av operatoren
- Hvert ledd gjøres om til CNF vha. en sannhetstabell
- CNF $\phi'' \to$ 3-CNF ϕ''':

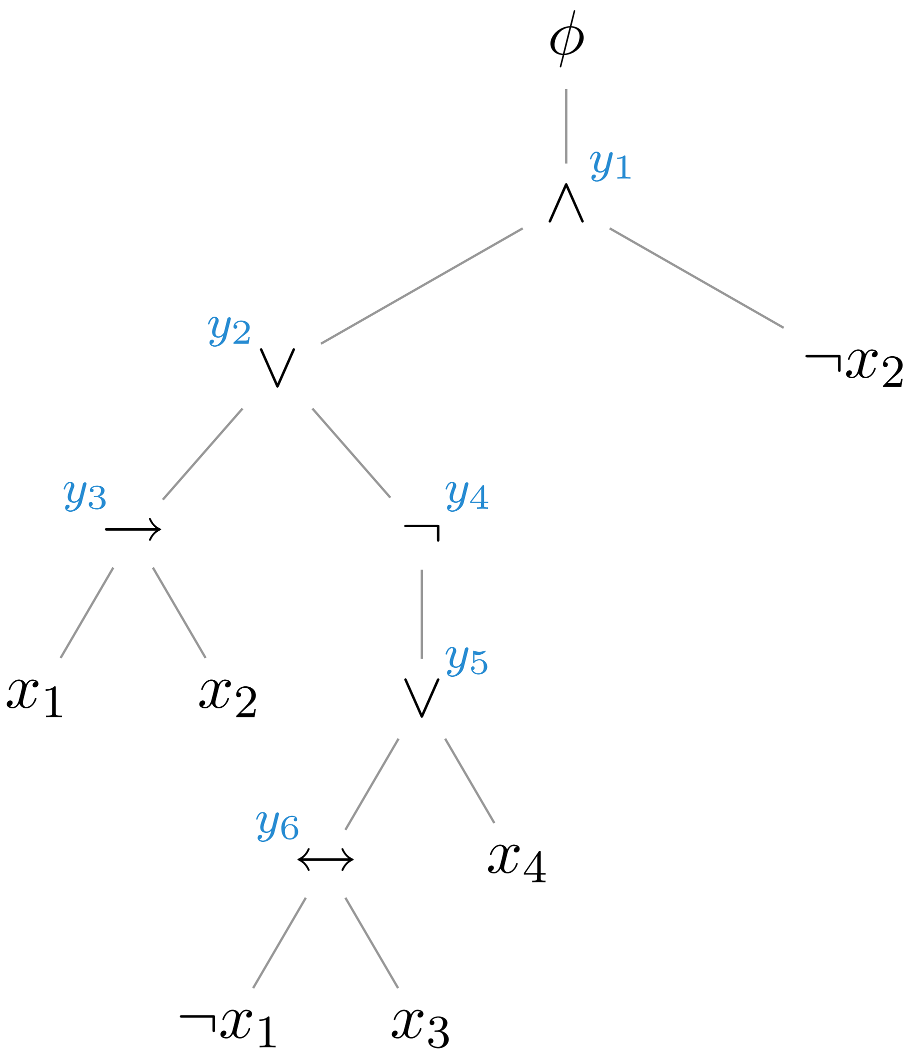
- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse
- ϕ' er en konjunksjon av ledd med maks 3 literaler
  - Dvs.: de to argumentene, samt resultatet av operatoren
- Hvert ledd gjøres om til CNF vha. en sannhetstabell
- CNF $\phi'' \rightarrow$ 3-CNF ϕ''':
  - $(x \vee y)$ gjøres om til $(x \vee y \vee z) \wedge (x \vee y \vee \neg z)$

- Vi kan bruke ca. samme reduksjon, på syntakstreet til ϕ!
- Vi får da en formel ϕ' av pol. størrelse
- ϕ' er en konjunksjon av ledd med maks 3 literaler
  - Dvs.: de to argumentene, samt resultatet av operatoren
- Hvert ledd gjøres om til CNF vha. en sannhetstabell
- CNF $\phi'' \to$ 3-CNF ϕ''':
  - $(x \vee y)$ gjøres om til $(x \vee y \vee z) \wedge (x \vee y \vee \neg z)$
  - Tilsvarende blir (x) til fire nye disjunksjoner

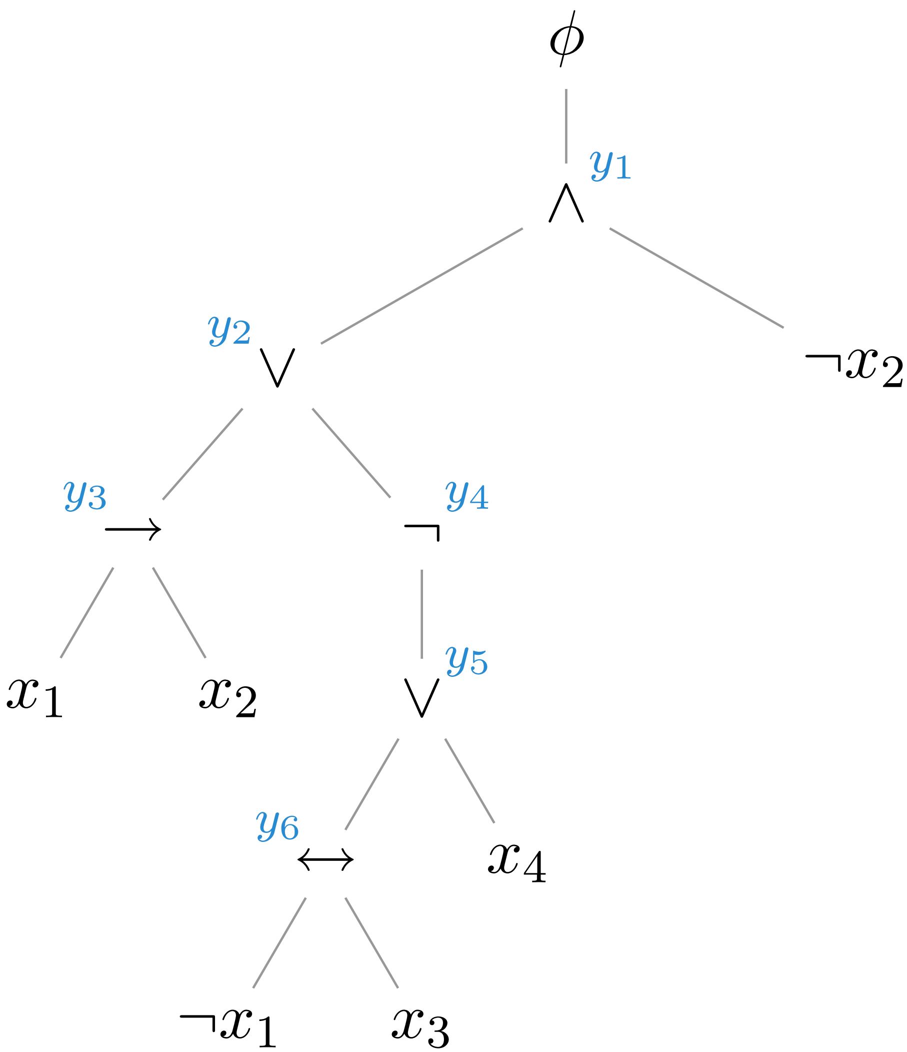
$$\phi = ((x_1 \to x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$$

Kan ϕ være sann? $\iff$ Kan ϕ''' være sann?

$\bullet \longmapsto \bullet$

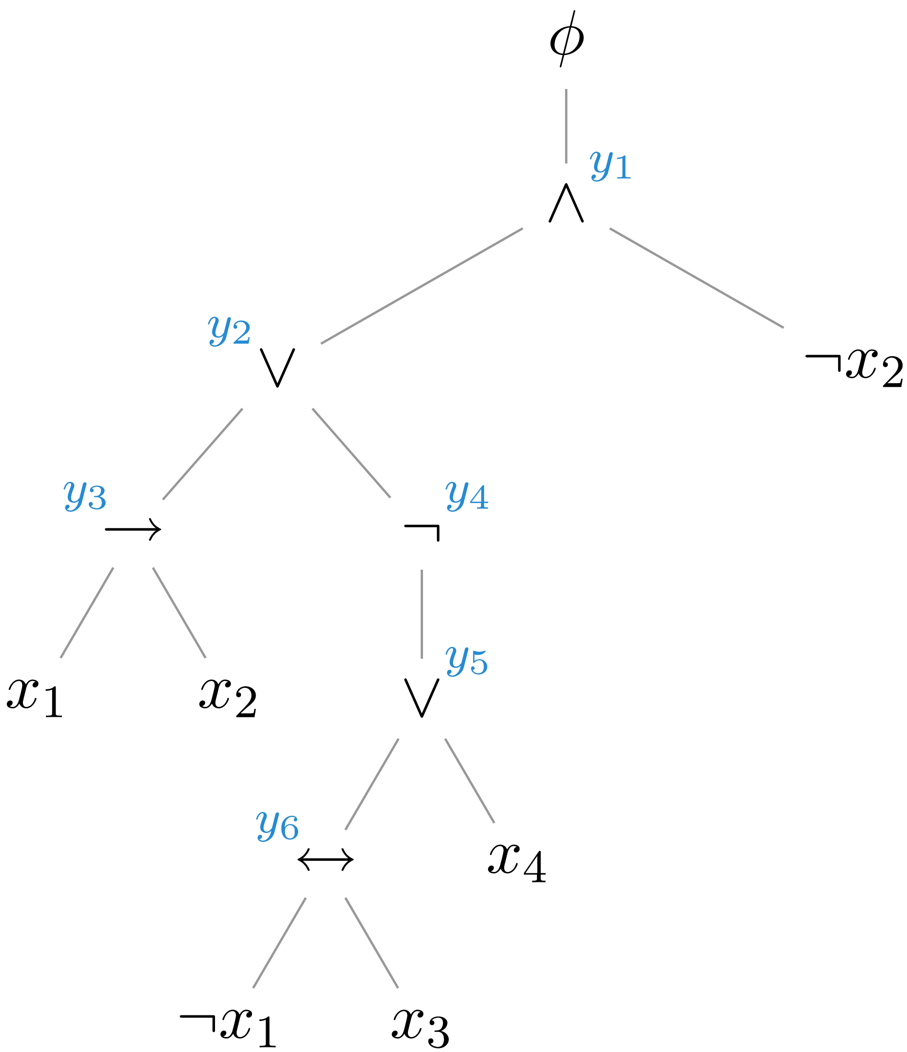


Kan ϕ være sann? $\Longleftrightarrow$ Kan ϕ''' være sann?



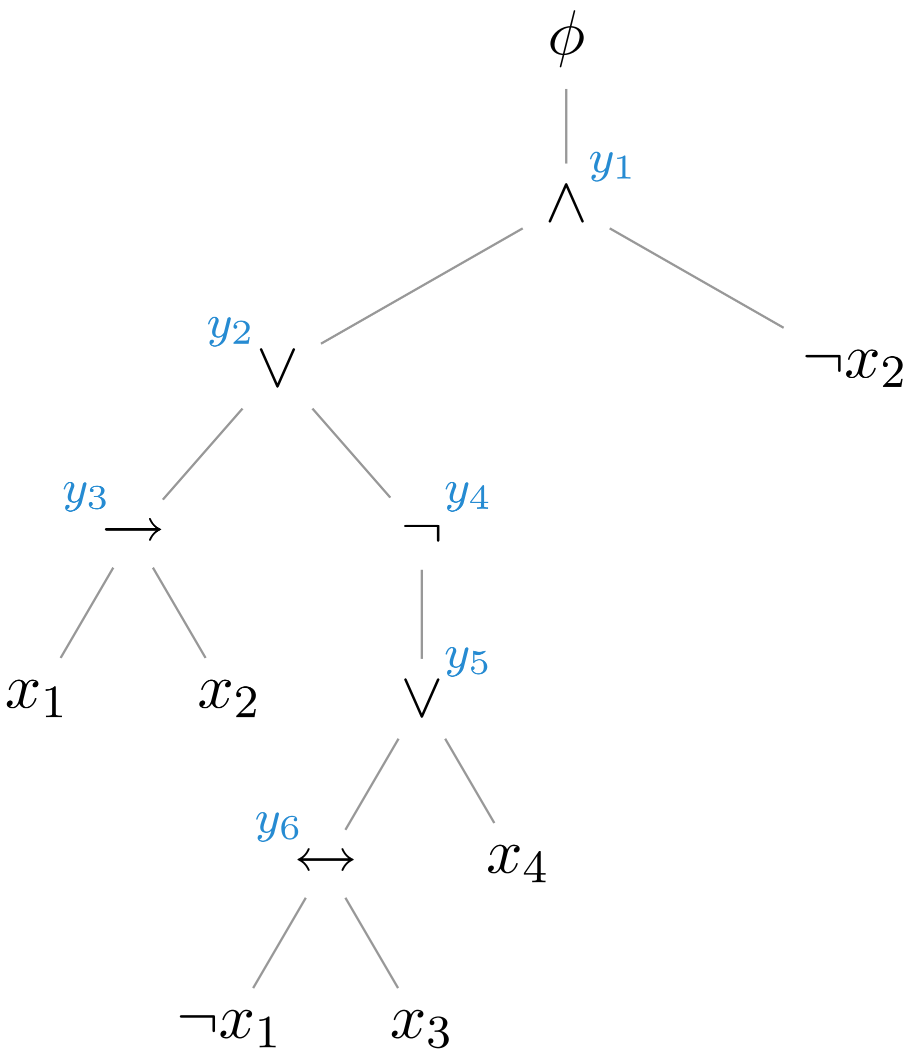
$$
\begin{aligned}
\phi' \quad = \quad & y_1 \wedge (y_1 \leftrightarrow (y_2 \wedge \neg x_2)) \\
& \wedge (y_2 \leftrightarrow (y_3 \vee y_4)) \\
& \wedge (y_3 \leftrightarrow (x_1 \rightarrow x_2)) \\
& \wedge (y_4 \leftrightarrow \neg y_5) \\
& \wedge (y_5 \leftrightarrow (y_6 \vee x_4)) \\
& \wedge (y_6 \leftrightarrow (\neg x_1 \leftrightarrow x_3))
\end{aligned}
$$

Kan ϕ være sann? $\Longleftrightarrow$ Kan ϕ''' være sann?



$$
\begin{aligned}
\phi' \;=\;& y_1 \wedge \big(y_1 \leftrightarrow (y_2 \wedge \neg x_2)\big)\\
& \wedge \big(y_2 \leftrightarrow (y_3 \vee y_4)\big)\\
& \wedge \big(y_3 \leftrightarrow (x_1 \rightarrow x_2)\big)\\
& \wedge \big(y_4 \leftrightarrow \neg y_5\big)\\
& \wedge \big(y_5 \leftrightarrow (y_6 \vee x_4)\big)\\
& \wedge \big(y_6 \leftrightarrow (\neg x_1 \leftrightarrow x_3)\big)\\
\phi'' \;=\;& \text{CNF, vha. sannhetstabeller}
\end{aligned}
$$

Kan ϕ være sann? $\iff$ Kan ϕ''' være sann?

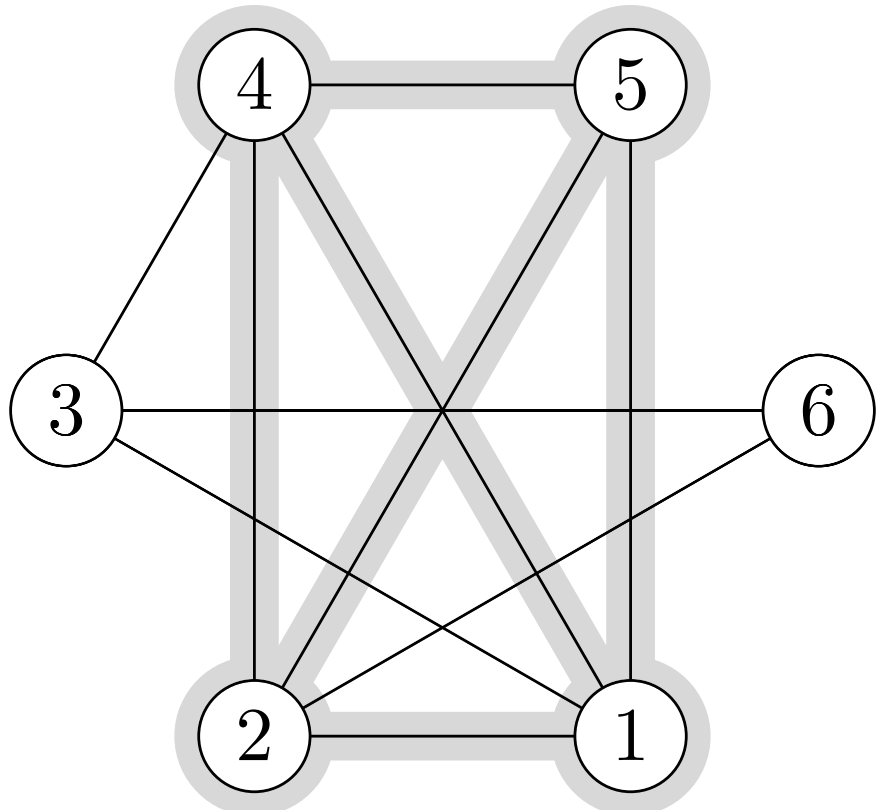


$$
\begin{aligned}
\phi' &= y_1 \wedge (y_1 \leftrightarrow (y_2 \wedge \neg x_2)) \\
&\quad \wedge (y_2 \leftrightarrow (y_3 \vee y_4)) \\
&\quad \wedge (y_3 \leftrightarrow (x_1 \rightarrow x_2)) \\
&\quad \wedge (y_4 \leftrightarrow \neg y_5) \\
&\quad \wedge (y_5 \leftrightarrow (y_6 \vee x_4)) \\
&\quad \wedge (y_6 \leftrightarrow (\neg x_1 \leftrightarrow x_3)) \\
\phi'' &= \text{CNF, vha. sannhetstabeller} \\
\phi''' &= \text{3-CNF, vha. dummy-variable}
\end{aligned}
$$

Kan ϕ være sann? $\Longleftrightarrow$ Kan ϕ''' være sann?

4:8

CLIQUE



CLIQUE

Instans: En urettet graf G og et positivt heltall k

Spørsmål: Har G en en komplett delgraf med k noder?

- Vi vil redusere fra 3-CNF-SAT

- Vi vil redusere fra 3-CNF-SAT
- Lag én node i G for hver literal i formelen

- Vi vil redusere fra 3-CNF-SAT
- Lag én node i G for hver literal i formelen
- Ingen kanter mellom literaler fra samme disjunksjon

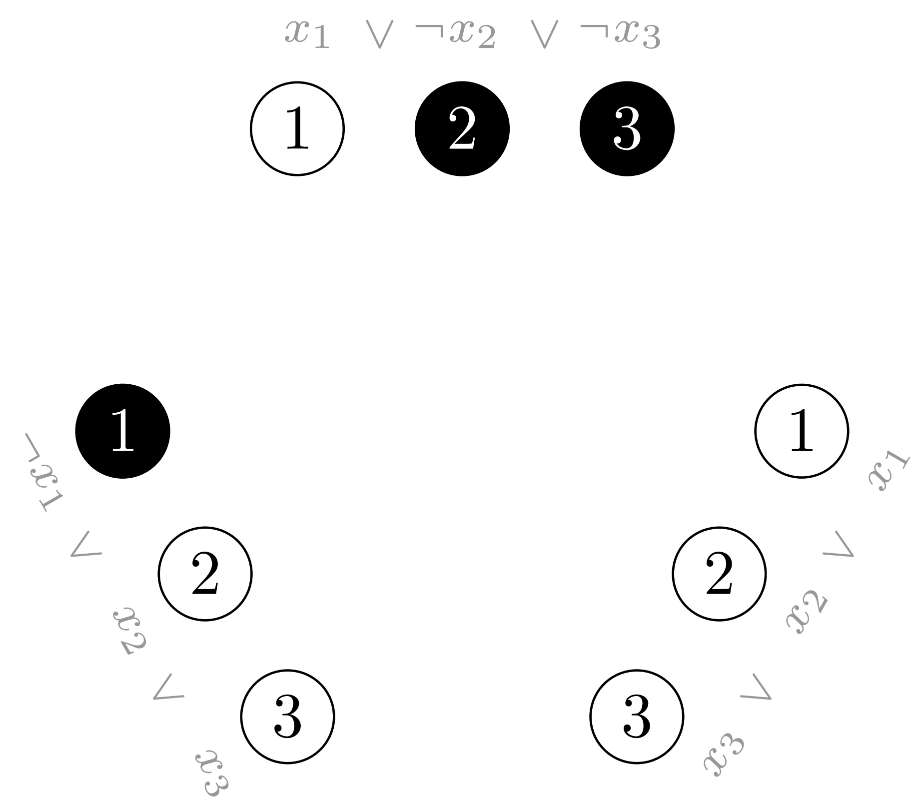
- Vi vil redusere fra 3-CNF-SAT
- Lag én node i G for hver literal i formelen
- Ingen kanter mellom literaler fra samme disjunksjon
- Ellers: Kanter mellom literaler som kan være sanne samtidig

- Vi vil redusere fra 3-CNF-SAT
- Lag én node i G for hver literal i formelen
- Ingen kanter mellom literaler fra samme disjunksjon
- Ellers: Kanter mellom literaler som kan være sanne samtidig
- La k være antall disjunksjoner

$$\phi = (\ x_1 \lor \neg x_2 \lor \neg x_3\ ) \land$$
$$(\neg x_1 \lor \ x_2 \lor \ x_3\ ) \land$$
$$(\ x_1 \lor \ x_2 \lor \ x_3\ )$$

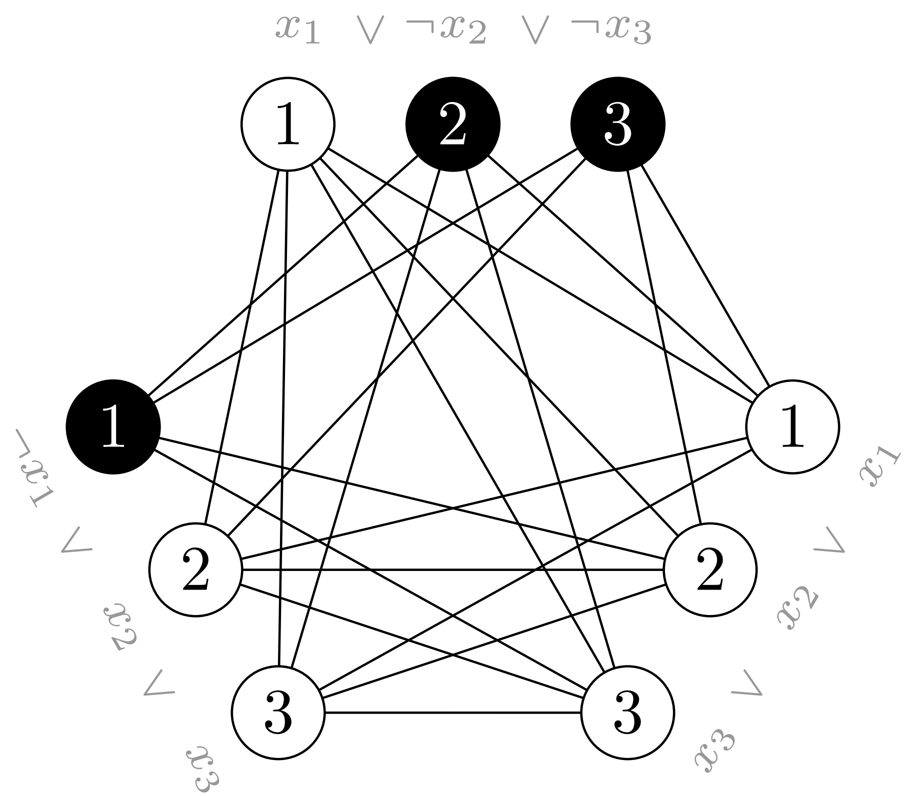
Kan ϕ være sann? $\iff$ Finnes en k-klikk?

$$\begin{aligned}\phi = (\ x_1 \vee \neg x_2 \vee \neg x_3\ ) \wedge \\ (\neg x_1 \vee\ x_2 \vee\ x_3\ ) \wedge \\ (\ x_1 \vee\ x_2 \vee\ x_3\ )\end{aligned}$$



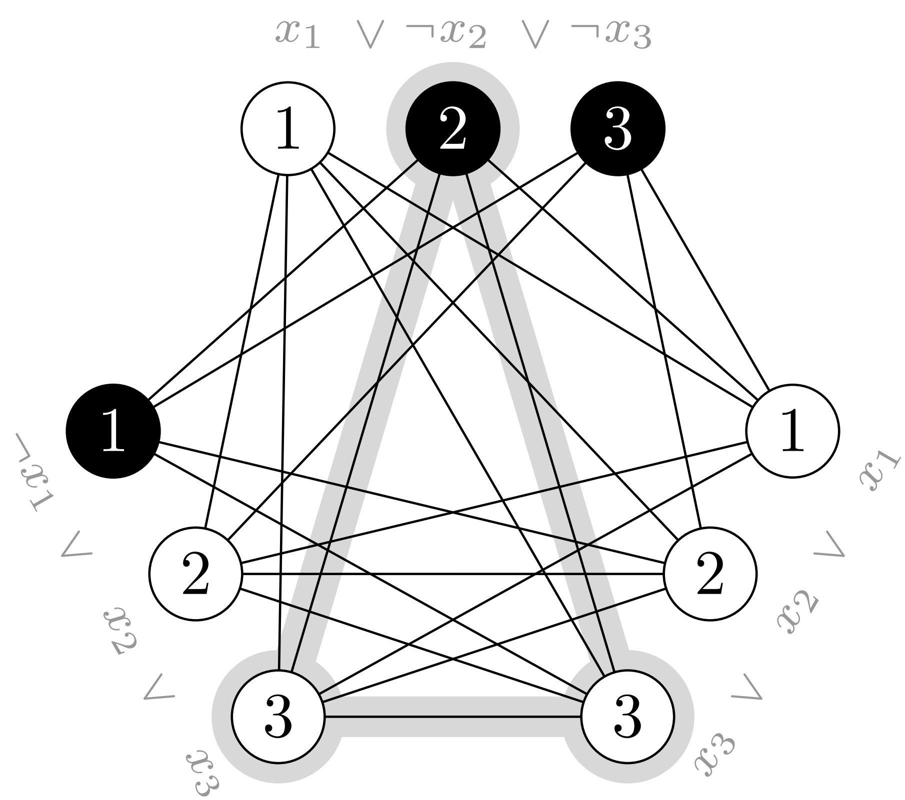
Kan ϕ være sann? $\Longleftrightarrow$ Finnes en k-klikk?

$$\begin{aligned}\phi = &(\ \ x_1 \lor \neg x_2 \lor \neg x_3) \land \\ &(\neg x_1 \lor \ \ x_2 \lor \ \ x_3) \land \\ &(\ \ x_1 \lor \ \ x_2 \lor \ \ x_3)\end{aligned}$$



Kan ϕ være sann? $\iff$ Finnes en k-klikk?

$$\begin{aligned}\phi = (\ \ x_1 &\vee \neg x_2 \vee \neg x_3\,) \wedge\\ (\neg x_1 &\vee\ \ x_2 \vee\ \ x_3\,) \wedge\\ (\ \ x_1 &\vee\ \ x_2 \vee\ \ x_3\,)\end{aligned}$$

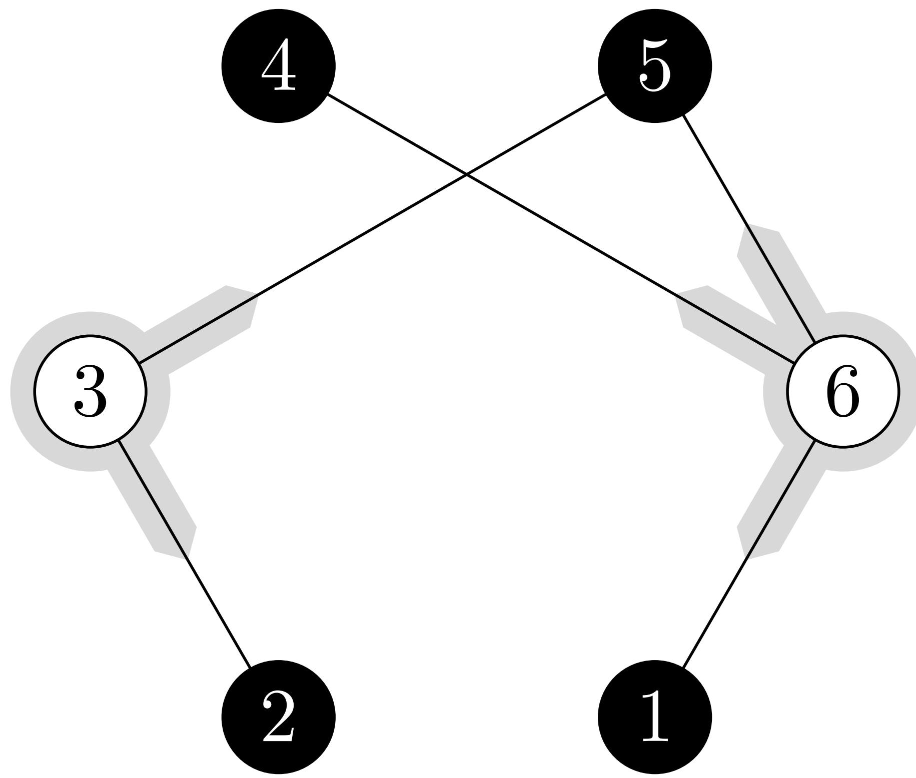


Tilsv. $x_1, x_2, x_3 = -, 0, 1$

Kan ϕ være sann? $\iff$ Finnes en k-klikk?

5:8

VERTEX-COVER



VERTEX-COVER

Instans: En urettet graf G og et positivt heltall k

Spørsmål: Har G en et nodedekke med k noder?

Dvs., k noder som tilsammen ligger inntil alle kantene

Vi antar at G ikke har noen isolerte noder; slike noder kan uansett fjernes

- En klikk er en komplett delgraf

- En klikk er en komplett delgraf
- Tilsvarer en uavhengig mengde (kantfri delgraf) i komplementet $\bar{\mathrm{G}} = (\mathrm{V}, \bar{\mathrm{E}})$

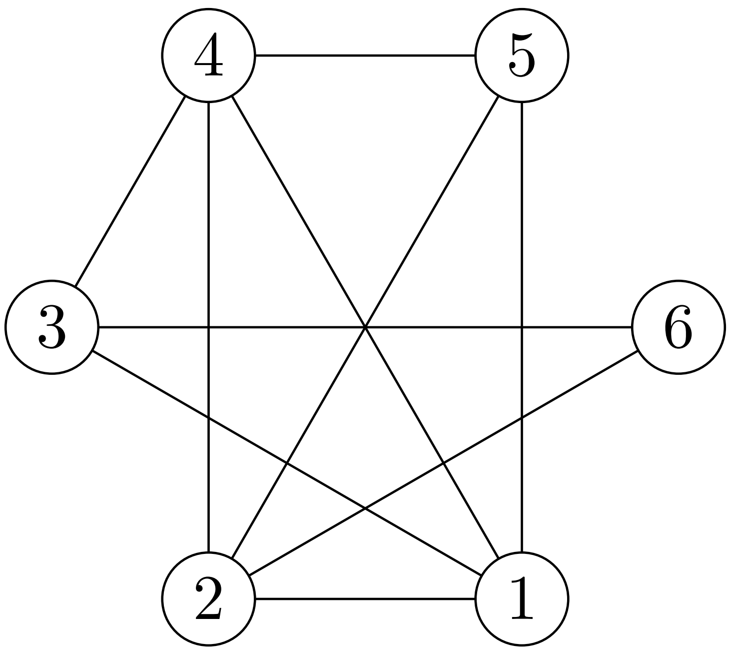
- En klikk er en komplett delgraf
- Tilsvarer en uavhengig mengde (kantfri delgraf) i komplementet $\bar{\mathrm{G}} = (\mathrm{V}, \bar{\mathrm{E}})$
- Nodene utenfor en uavhengig mengde utgjør et nodedekke

- En klikk er en komplett delgraf
- Tilsvarer en uavhengig mengde (kantfri delgraf) i komplementet $\bar{G} = (V, \bar{E})$
- Nodene utenfor en uavhengig mengde utgjør et nodedekke
- Hvis G har en k-klikk …

- En klikk er en komplett delgraf
- Tilsvarer en uavhengig mengde (kantfri delgraf) i komplementet $\bar{G} = (V, \bar{E})$
- Nodene utenfor en uavhengig mengde utgjør et nodedekke
- Hvis G har en k-klikk …
  - … så har $\bar{G} = (V, \bar{E})$ en uavh. mengde med k noder …

- En klikk er en komplett delgraf
- Tilsvarer en uavhengig mengde (kantfri delgraf) i komplementet $\bar{G} = (V, \bar{E})$
- Nodene utenfor en uavhengig mengde utgjør et nodedekke
- Hvis G har en k-klikk ...
  - ... så har $\bar{G} = (V, \bar{E})$ en uavh. mengde med k noder ...
  - ... og dermed også et $(|V| - k)$-nodedekke

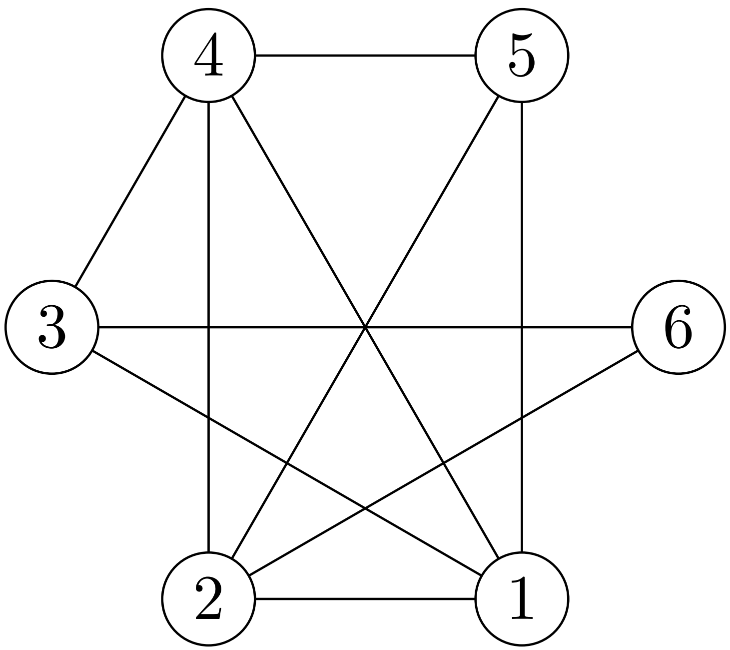
- En klikk er en komplett delgraf
- Tilsvarer en uavhengig mengde (kantfri delgraf) i komplementet $\bar{G} = (V, \bar{E})$
- Nodene utenfor en uavhengig mengde utgjør et nodedekke
- Hvis G har en k-klikk …
  - … så har $\bar{G} = (V, \bar{E})$ en uavh. mengde med k noder …
  - … og dermed også et $(|V| - k)$-nodedekke
- Samme resonnement holder i motsatt retning



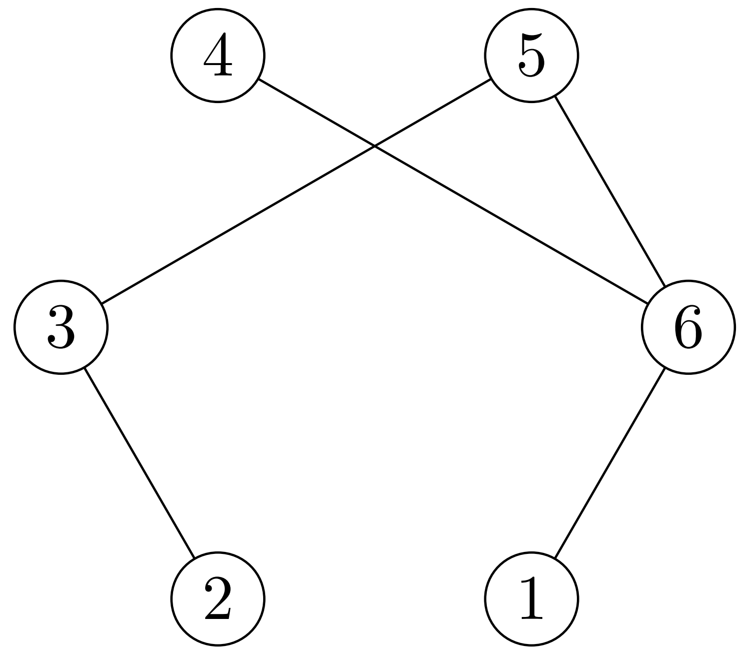
G

$\bar{G}$

Finnes en k-klikk? $\iff$ Finnes et $(|V|-k)$-dekke?

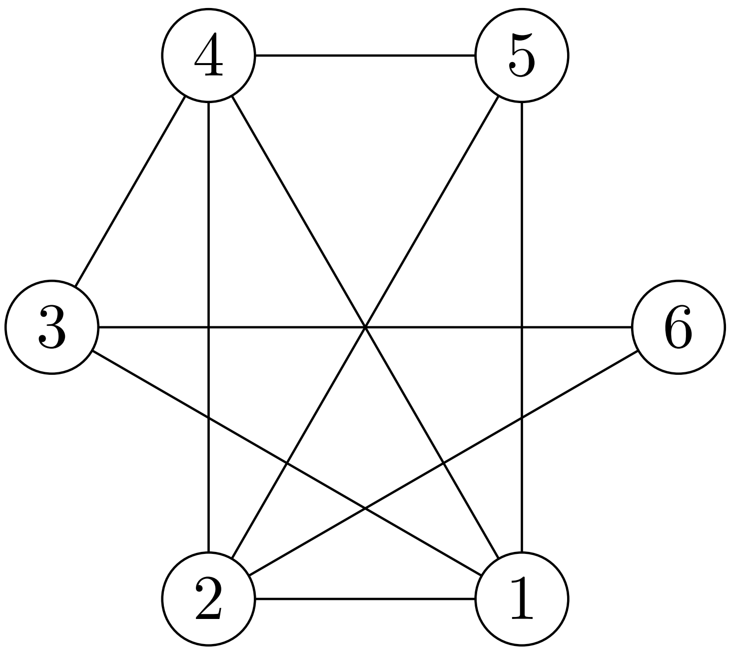


G

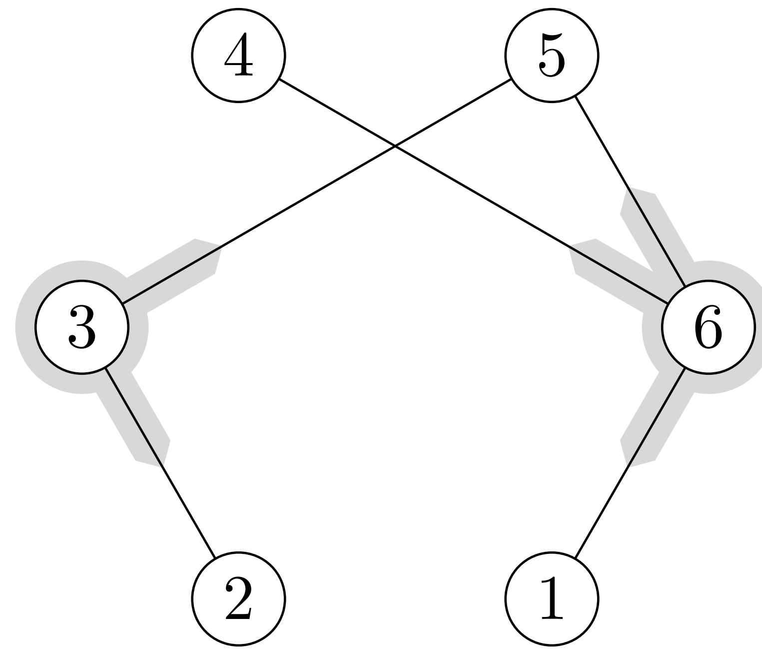


$\bar{\mathrm{G}}$

Finnes en k-klikk? $\Longleftrightarrow$ Finnes et $(|\mathrm{V}|-k)$-dekke?

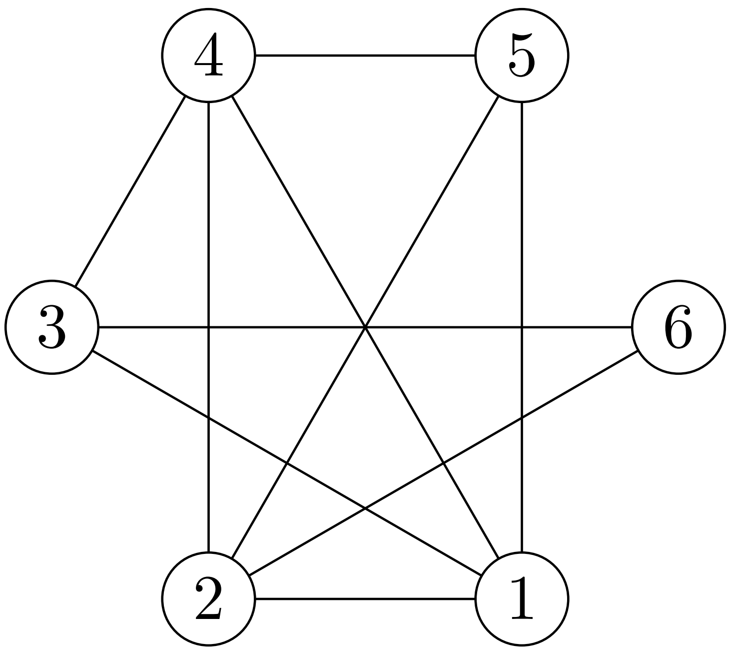


G

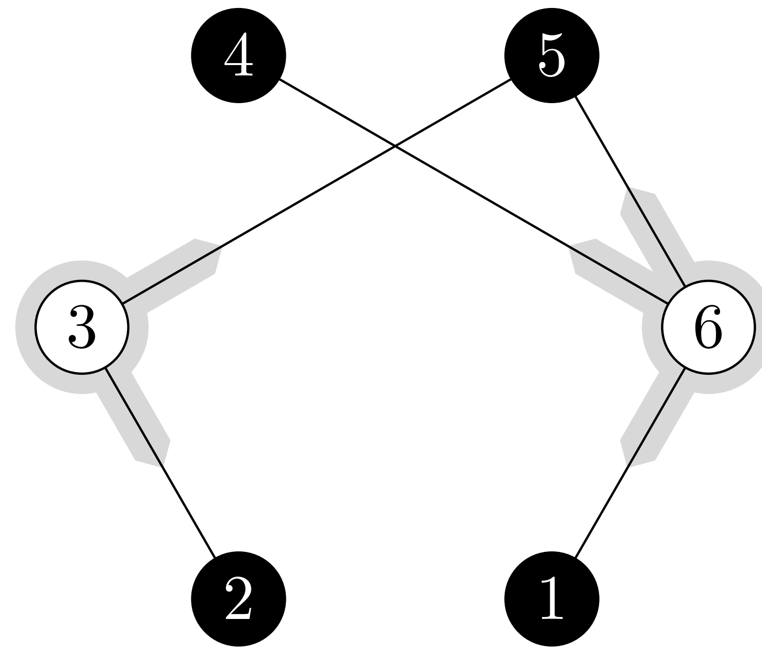


$\bar{G}$

Finnes en k-klikk? $\iff$ Finnes et $(|V|-k)$-dekke?

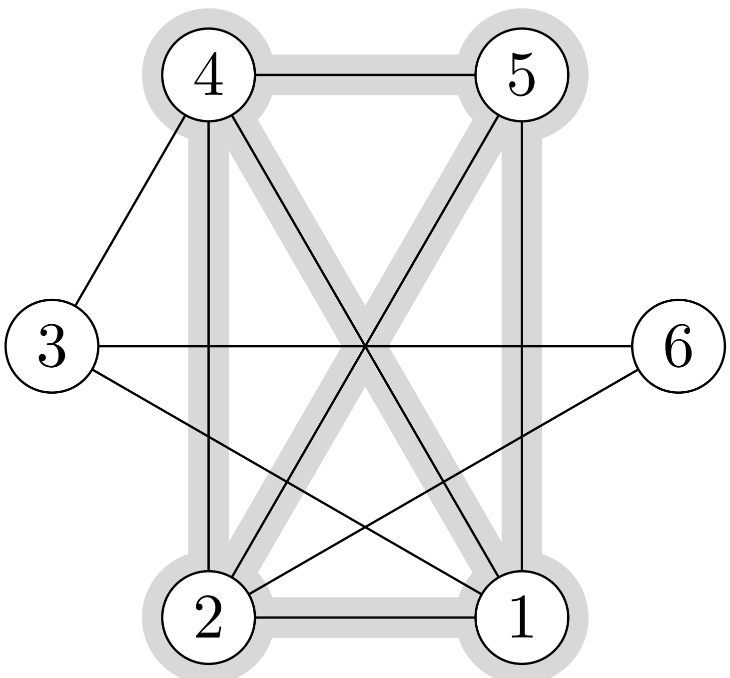


G

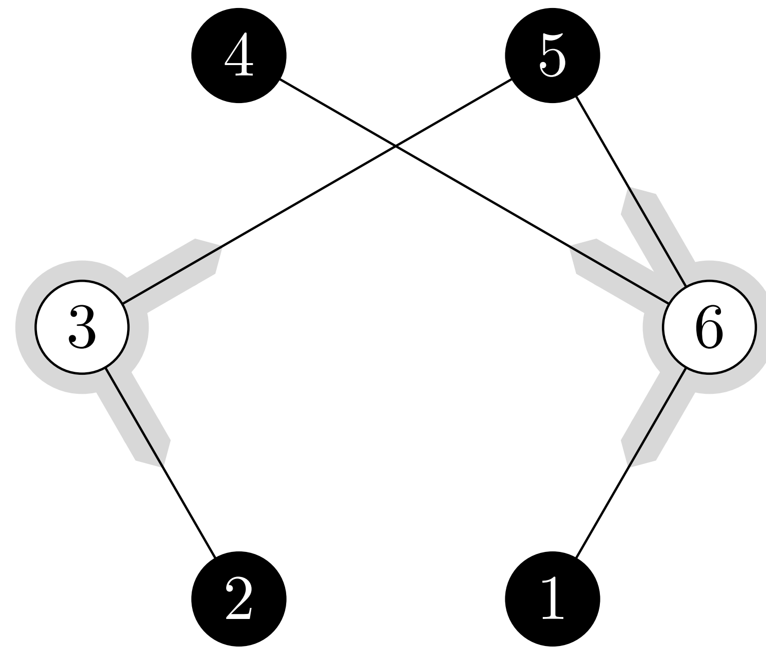


$\bar{G}$

Finnes en k-klikk? $\iff$ Finnes et $(|V|-k)$-dekke?



G

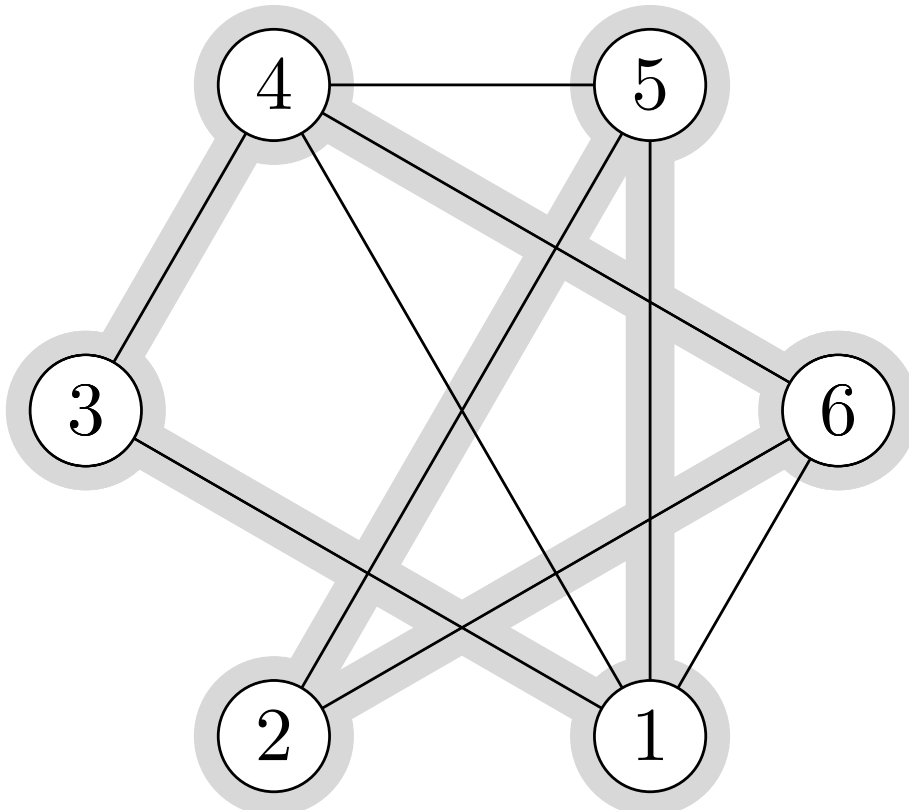


$\bar{\text{G}}$

Finnes en k-klikk? $\Longleftrightarrow$ Finnes et $(|\text{V}|-k)$-dekke?

6:8

HAM-CYCLE



HAM-CYCLE

Instans: En urettet graf G

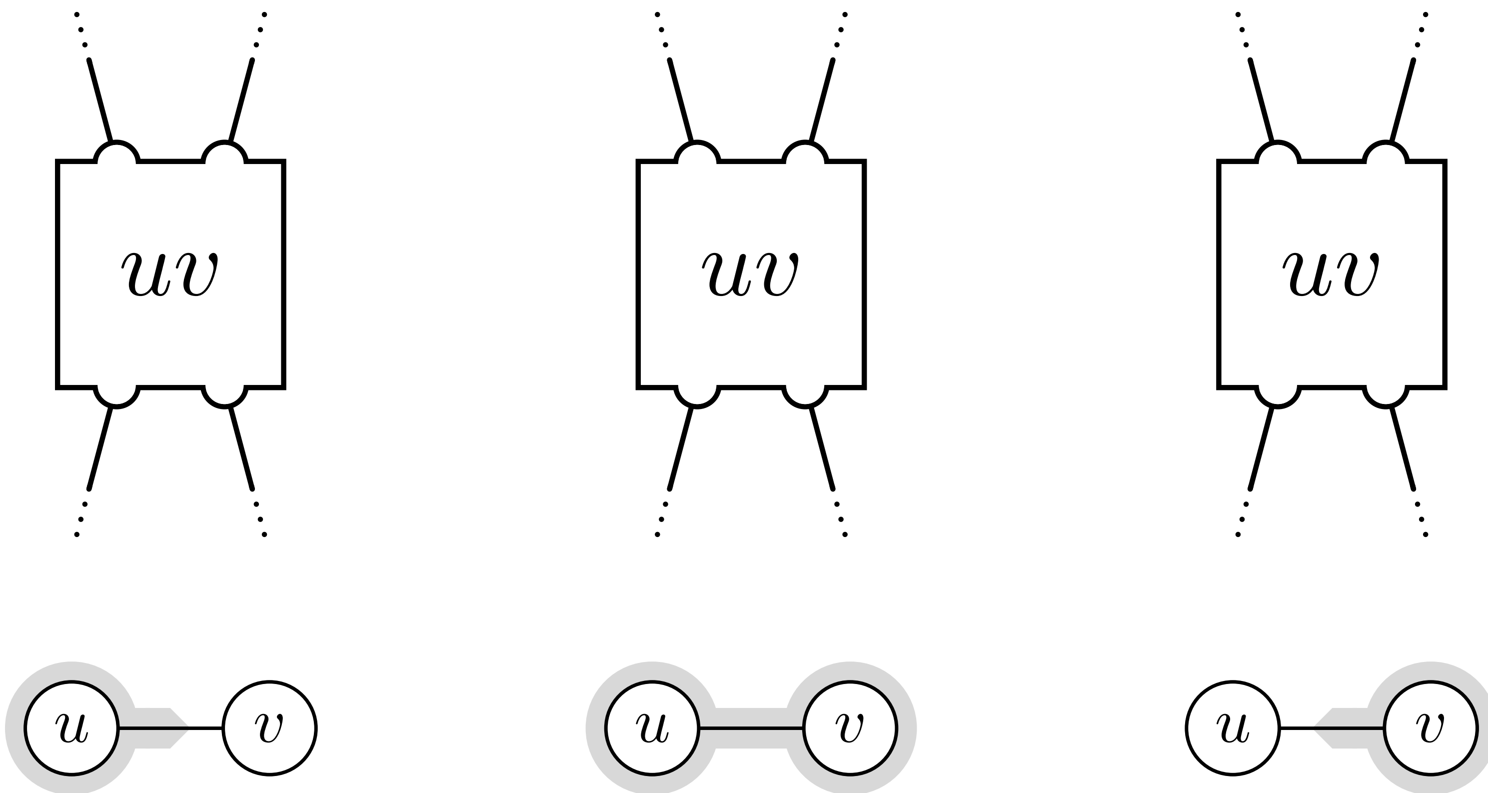
Spørsmål: Finnes det en sykel som inneholder alle nodene?

- Vi reduserer fra VERTEX-COVER

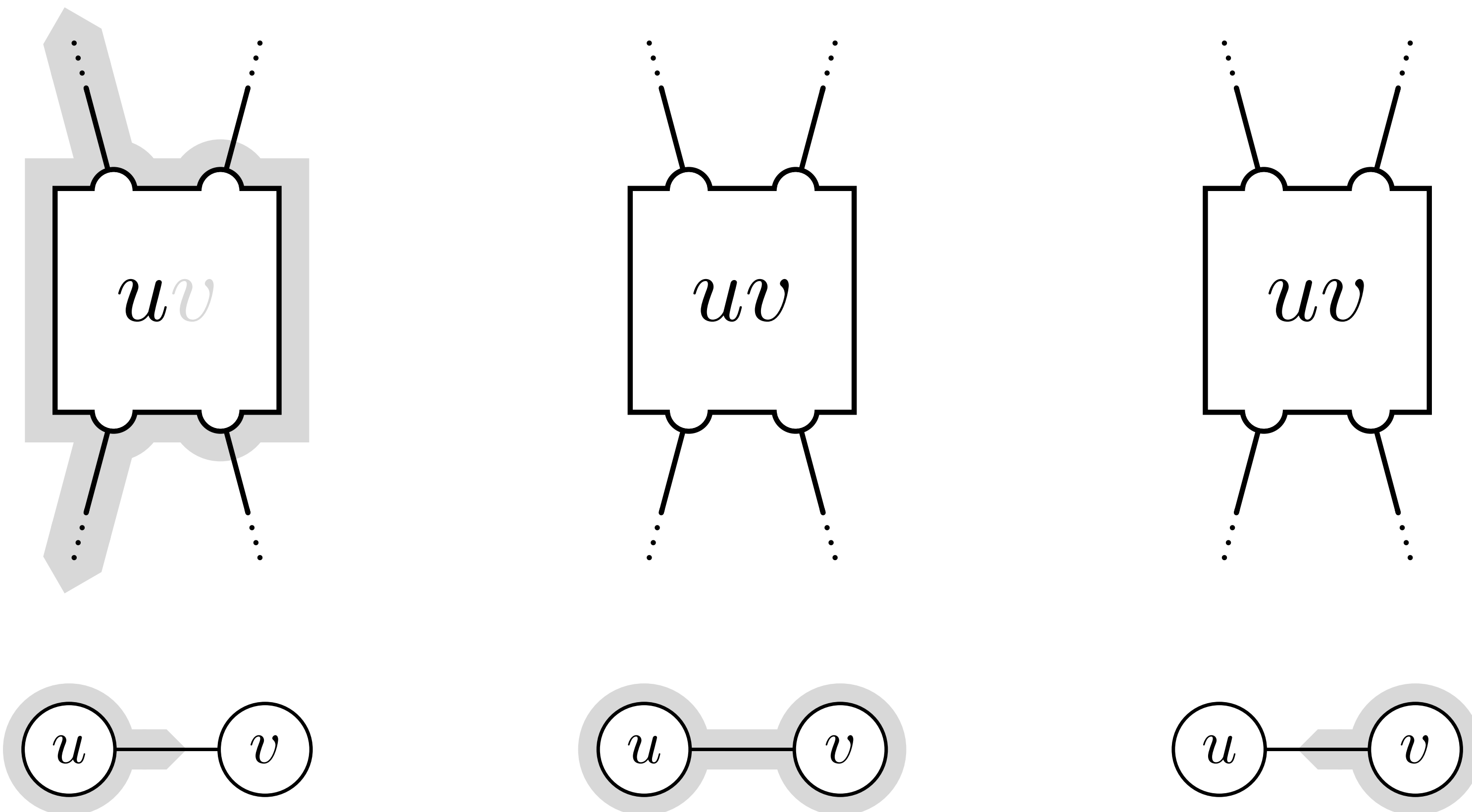
- Vi reduserer fra VERTEX-COVER
- Sykelen må kunne velge noder til nodedekket

- Vi reduserer fra VERTEX-COVER
- Sykelen må kunne velge noder til nodedekket
- En kant (u, v) kan dekkes av u eller v eller begge

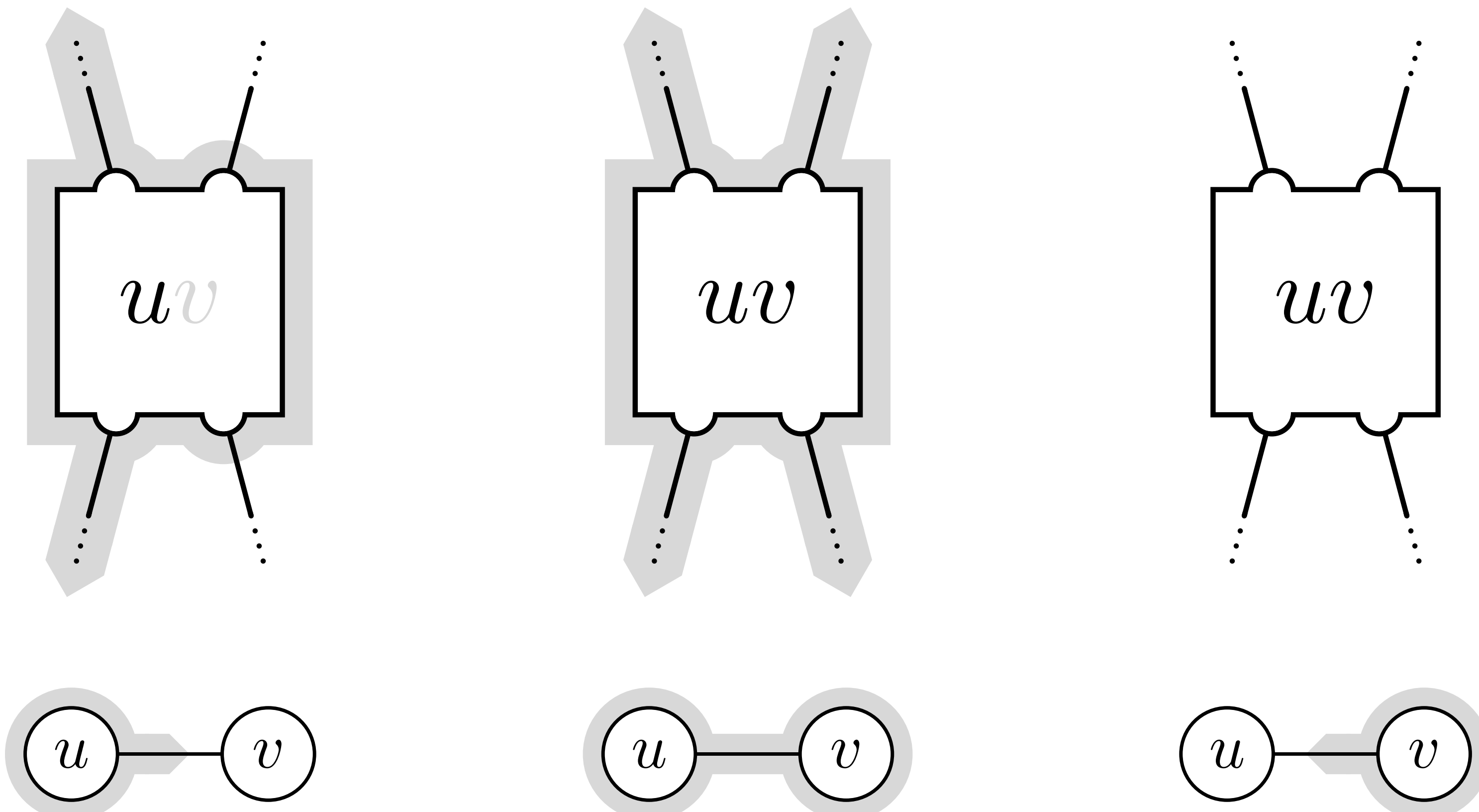
- Vi reduserer fra VERTEX-COVER
- Sykelen må kunne velge noder til nodedekket
- En kant (u, v) kan dekkes av u eller v eller begge
- Vi lager oss en innretning (*widget*) som hjelper oss med dette



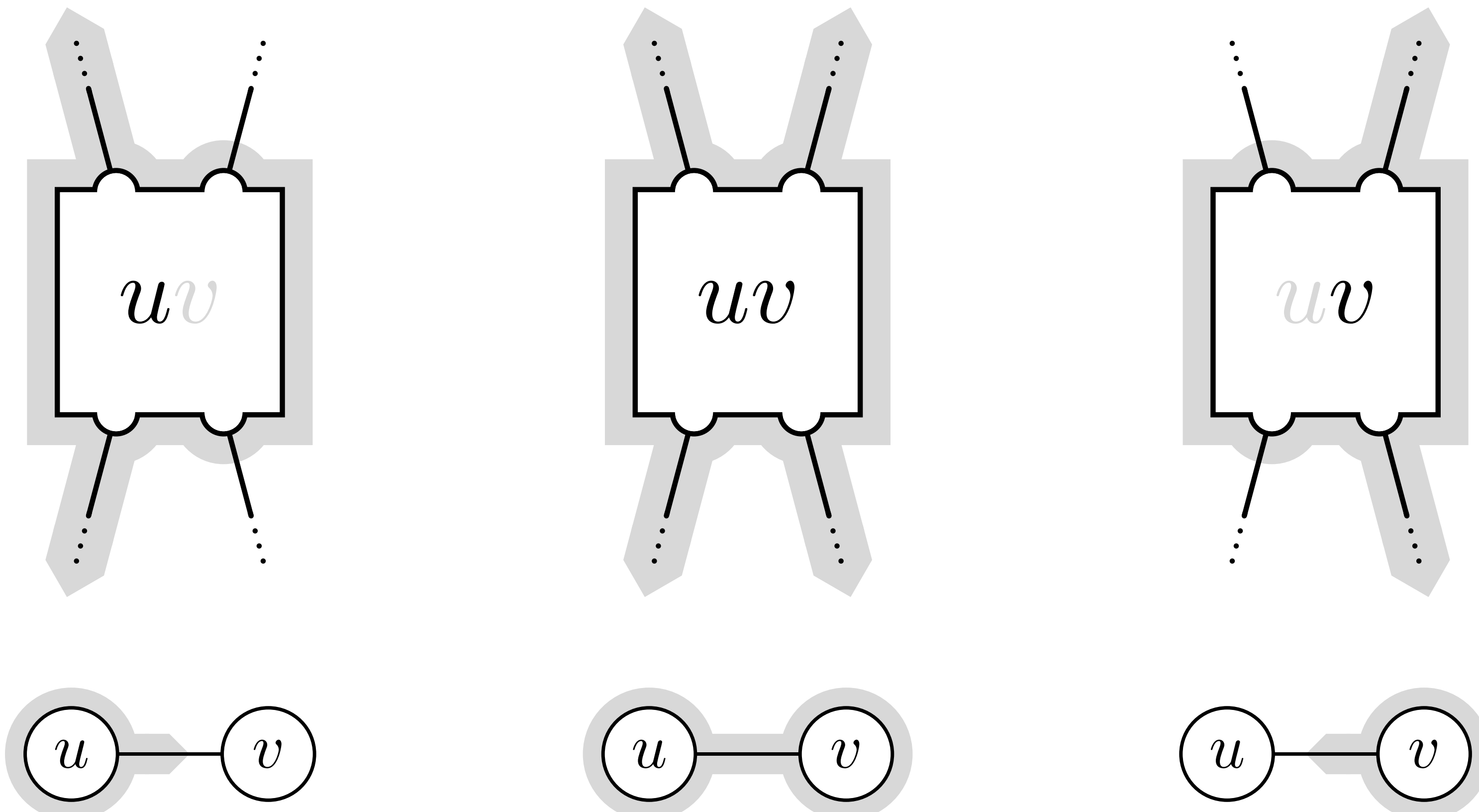
En innretning uv representerer kanten mellom u og v



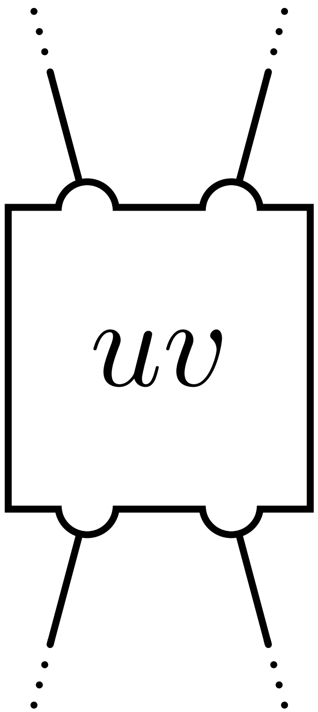
Om sykelen går igjennom på venstre side, velges u



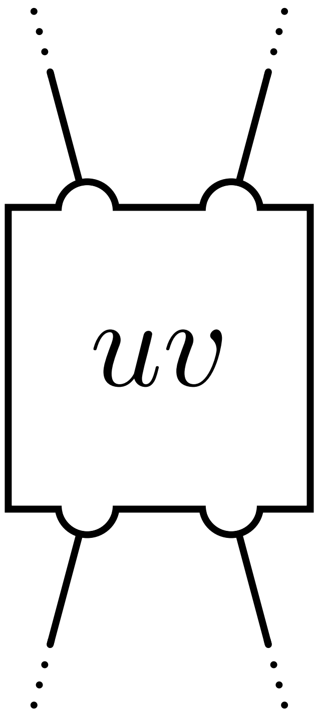
Om sykelen går igjennom på begge sider, velges begge

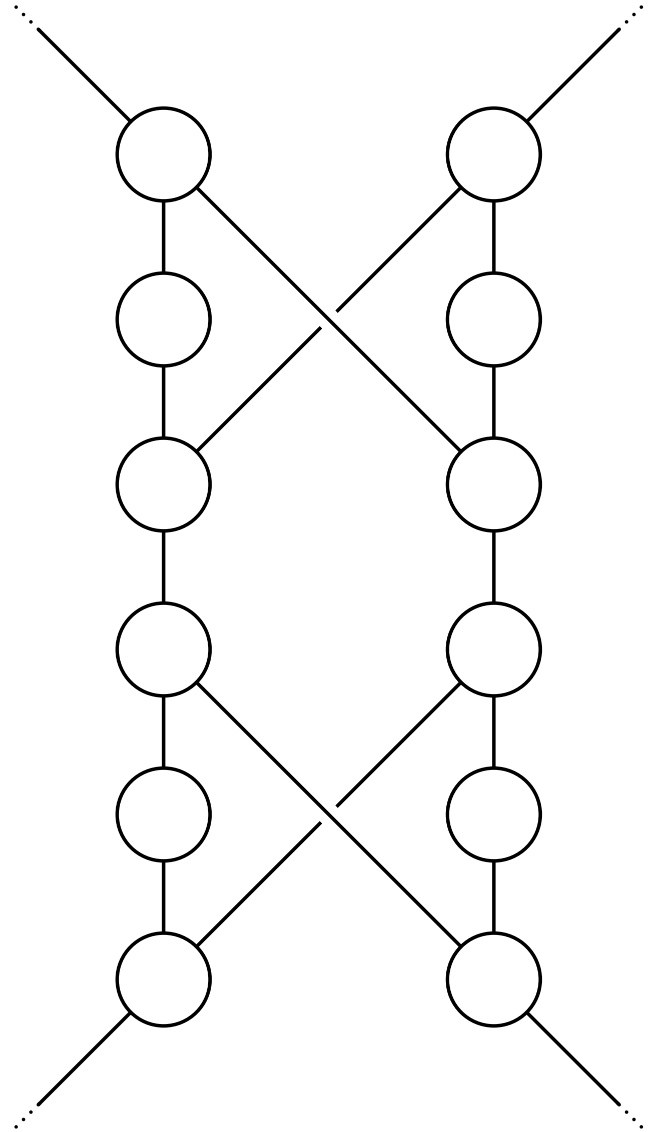


Om sykelen går igjennom på høyre side, velges v

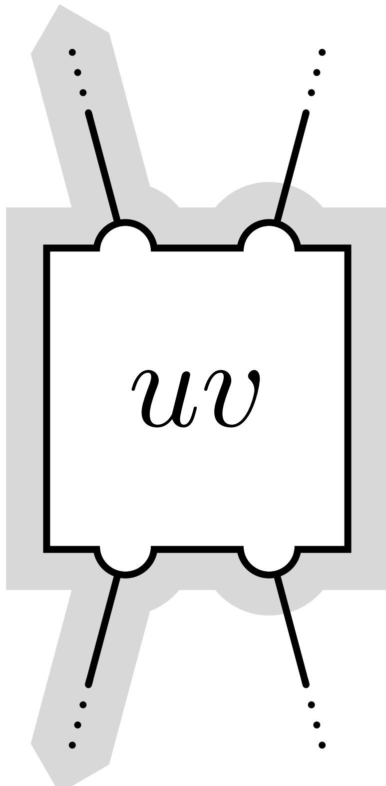


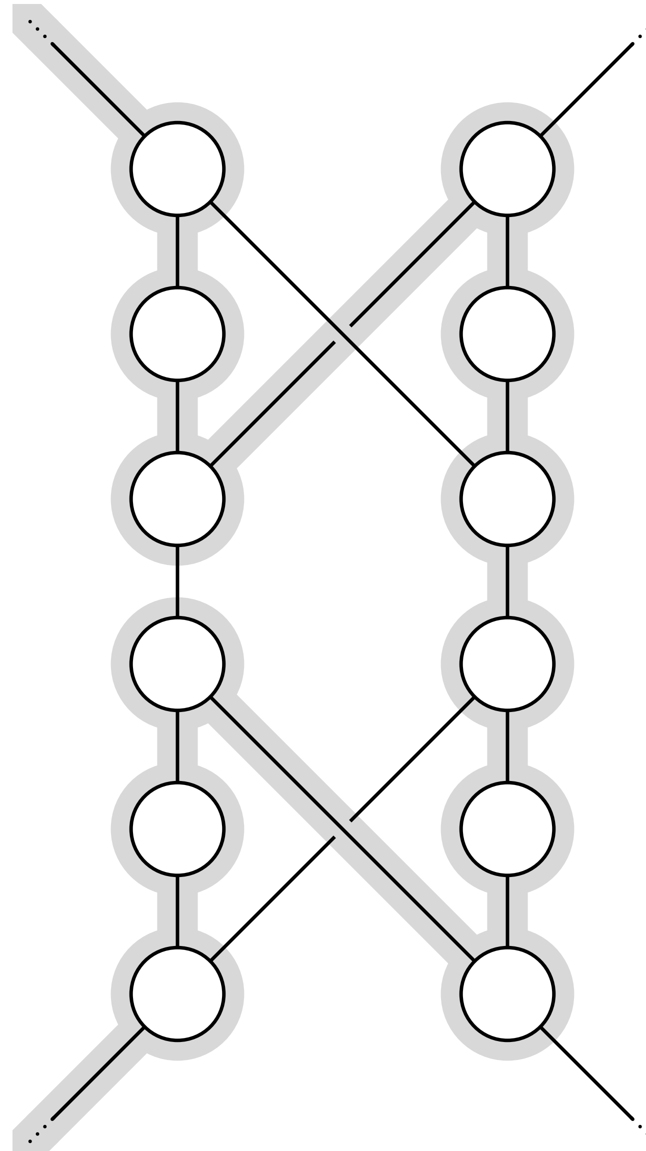
Hvordan lager vi disse?

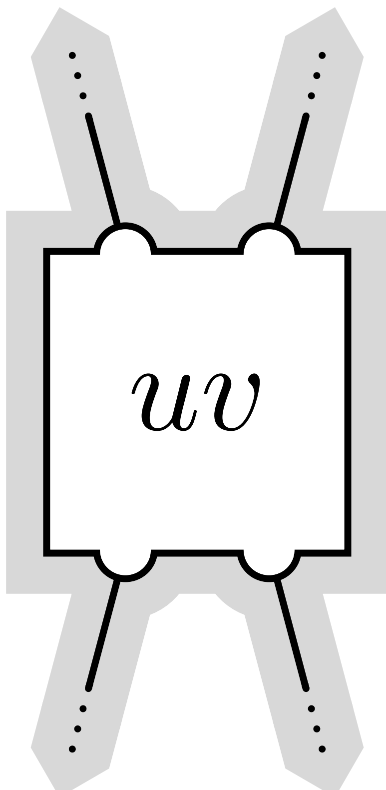


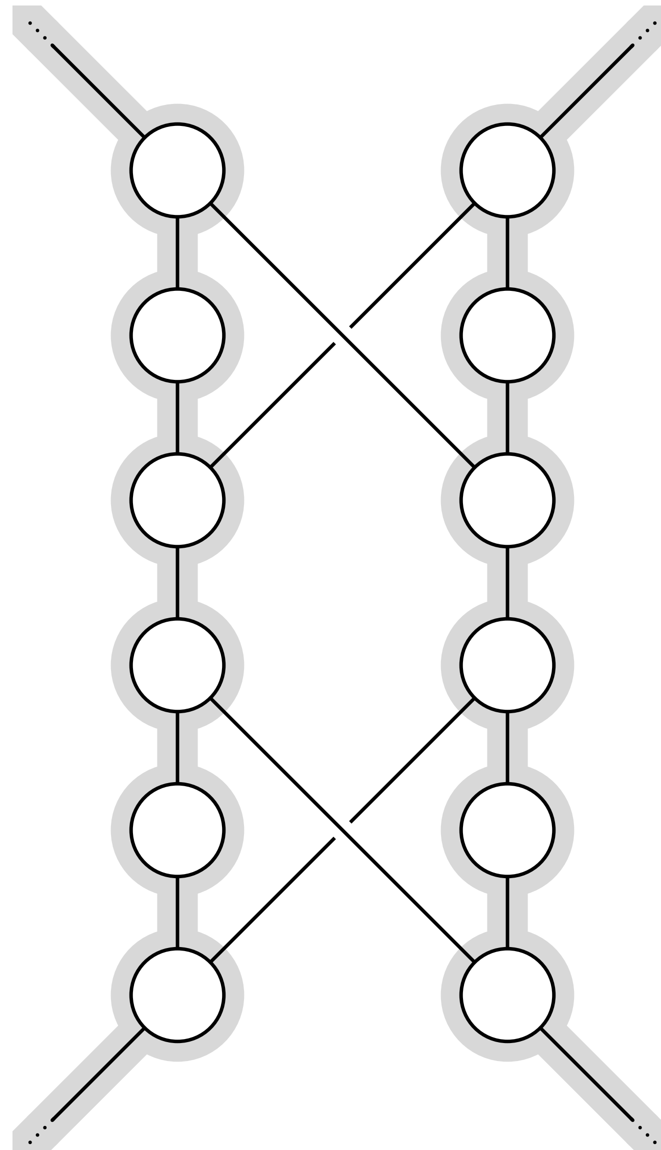


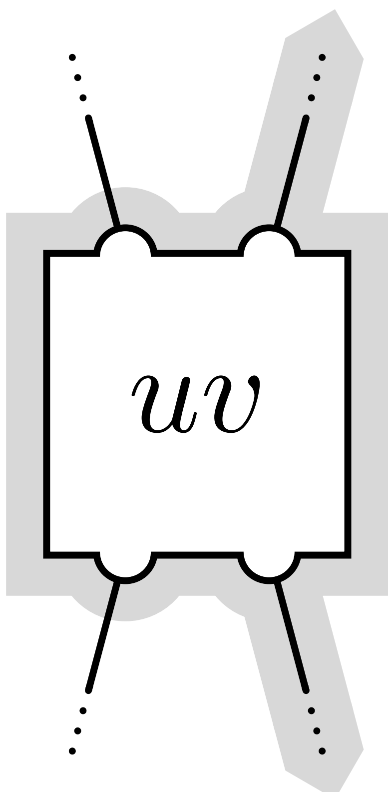
Kan uansett besøke alle

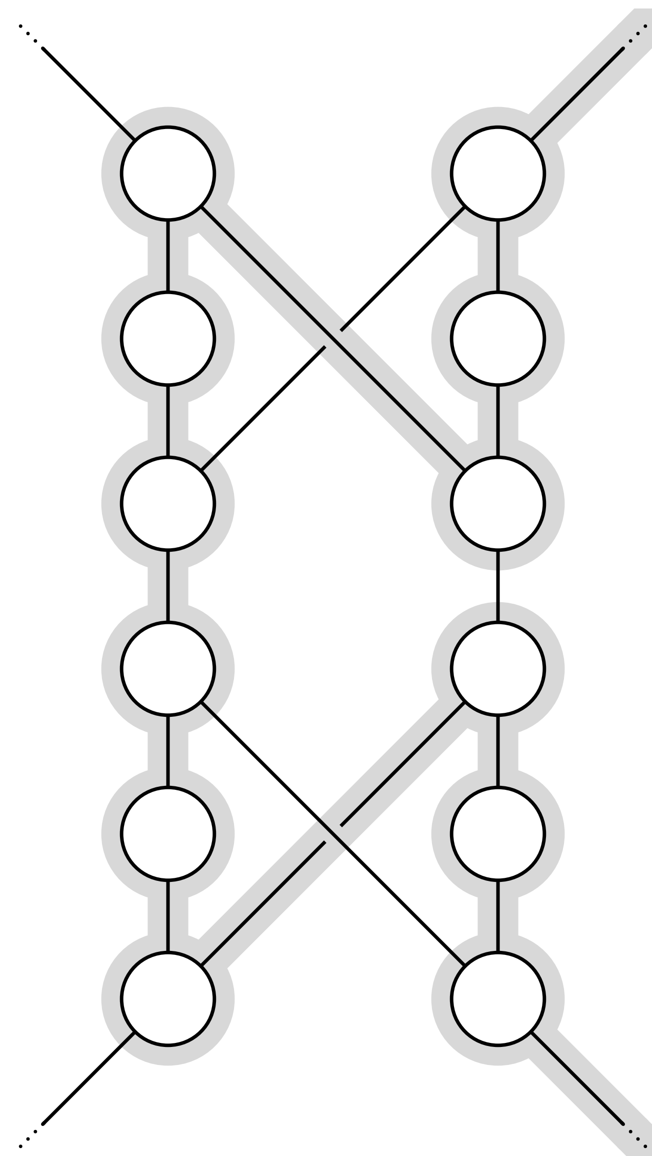

uv




uv




uv



- Lag «nabolister» ved hjelp av innretningene våre

Husk: Hver innretning representerer én av kantene i grafen

- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig

Vi lager oss en liste med ut-kanter (altså av innretninger) for v

- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig
  - Koble nederste innretningsnode til øverste for neste

På den siden av innretningen som representerer v

- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig
  - Koble nederste innretningsnode til øverste for neste
- Koble k «selektornoder» til første øverste og siste nederste

Selektornode i representerer den i-ende noden i nodedekket

- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig
  - Koble nederste innretningsnode til øverste for neste
- Koble k «selektornoder» til første øverste og siste nederste
- Hver selektornode kan da velge seg en naboliste

Velger node og «dekker» kanter (sender sykel gjennom innretninger)

- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig
  - Koble nederste innretningsnode til øverste for neste
- Koble k «selektornoder» til første øverste og siste nederste
- Hver selektornode kan da velge seg en naboliste
  - Den «velger» ved å sende hamiltonsykelen til den første, øverste innretningsnoden i nabolista

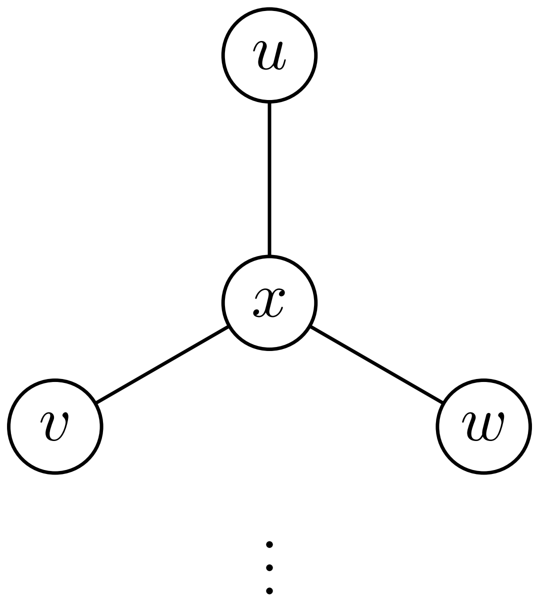
For v sin nabokant-liste: Går gjennom innretninger i lista på v-siden

- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig
  - Koble nederste innretningsnode til øverste for neste
- Koble k «selektornoder» til første øverste og siste nederste
- Hver selektornode kan da velge seg en naboliste
  - Den «velger» ved å sende hamiltonsykelen til den første, øverste innretningsnoden i nabolista
- Første og siste innretning i lista er koblet til alle k selektornoder

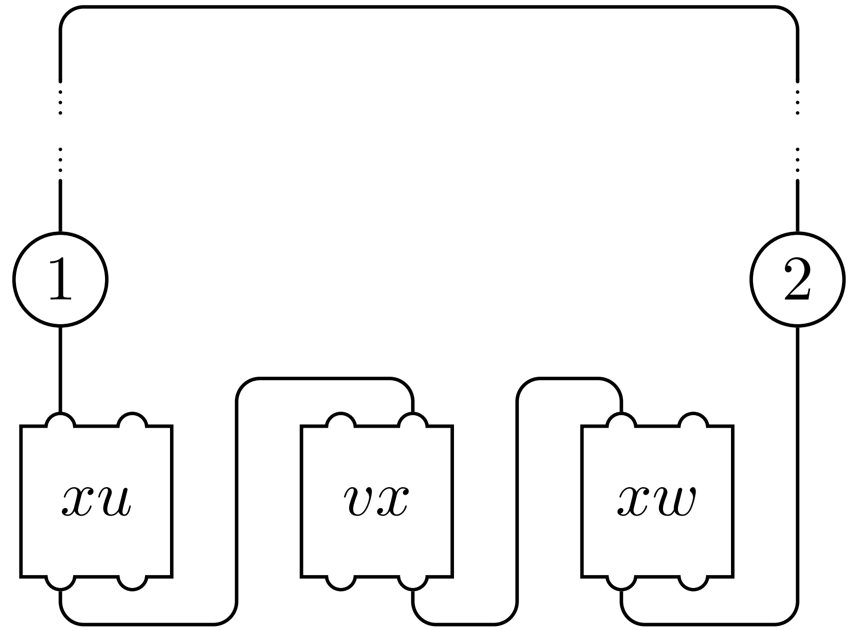
Siste innretning sender sykelen videre til neste (ev. første) selektornode

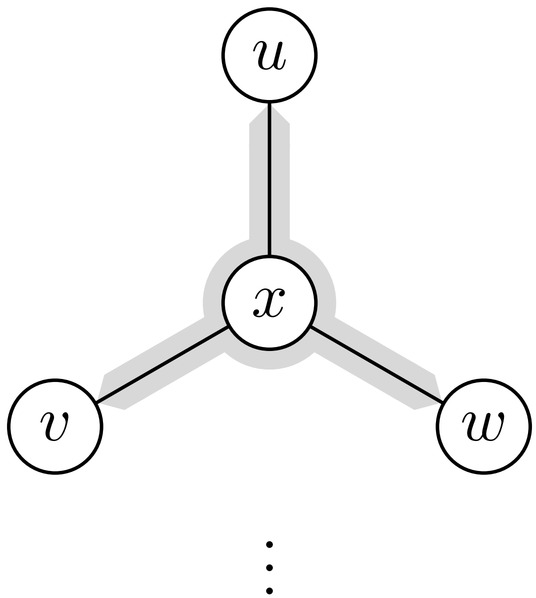
- Lag «nabolister» ved hjelp av innretningene våre
  - Ordne utkanter for hver originale node vilkårlig
  - Koble nederste innretningsnode til øverste for neste
- Koble k «selektornoder» til første øverste og siste nederste
- Hver selektornode kan da velge seg en naboliste
  - Den «velger» ved å sende hamiltonsykelen til den første, øverste innretningsnoden i nabolista
- Første og siste innretning i lista er koblet til alle k selektornoder
- Vi har en hamiltonsykel hvis og bare hvis hver naboliste velges

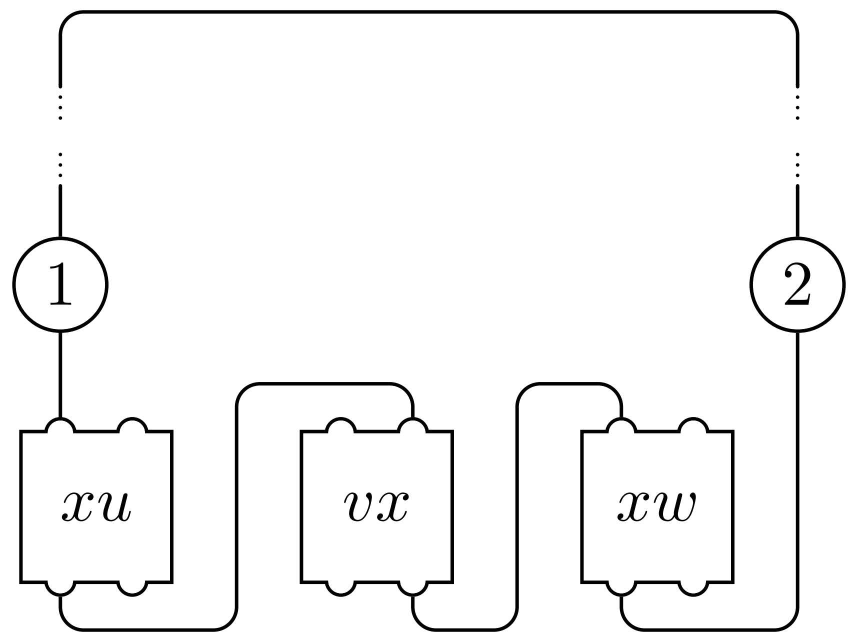
Vi er innom selektornodene, og går gjennom nabolister mellom dem



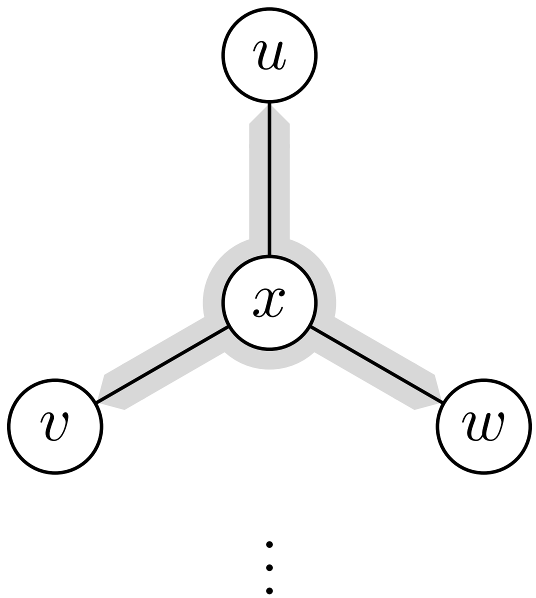
Vi velger x med selektor 1



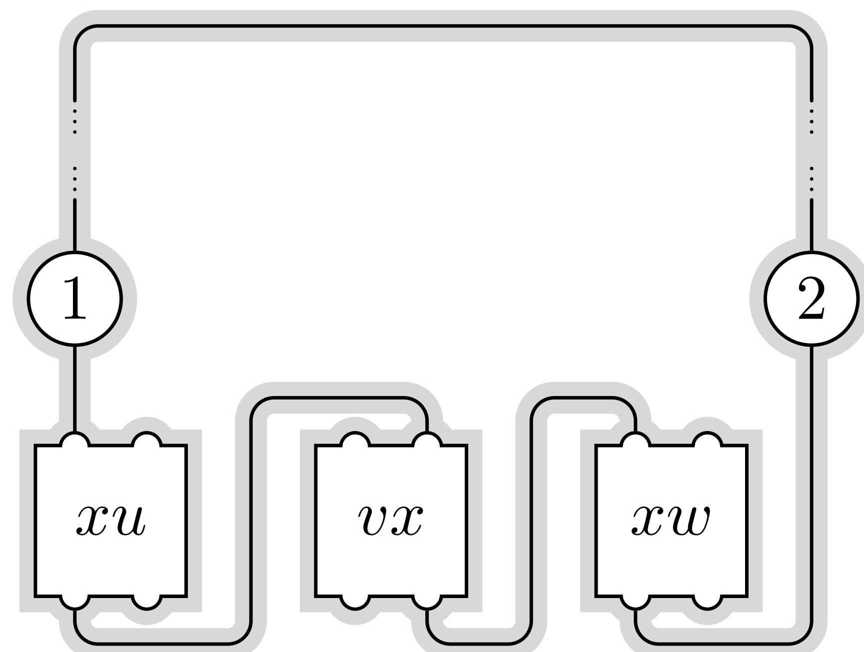




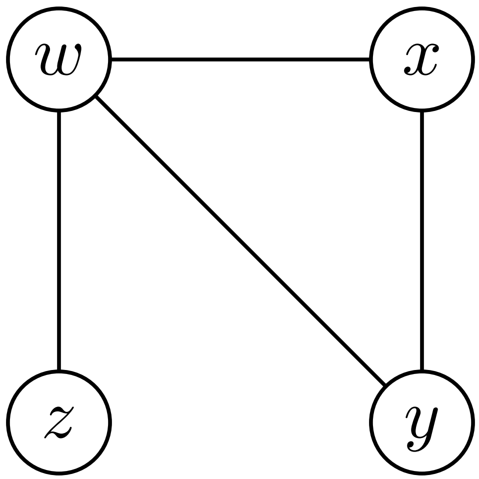
Dekker kanter xu, xv og xw



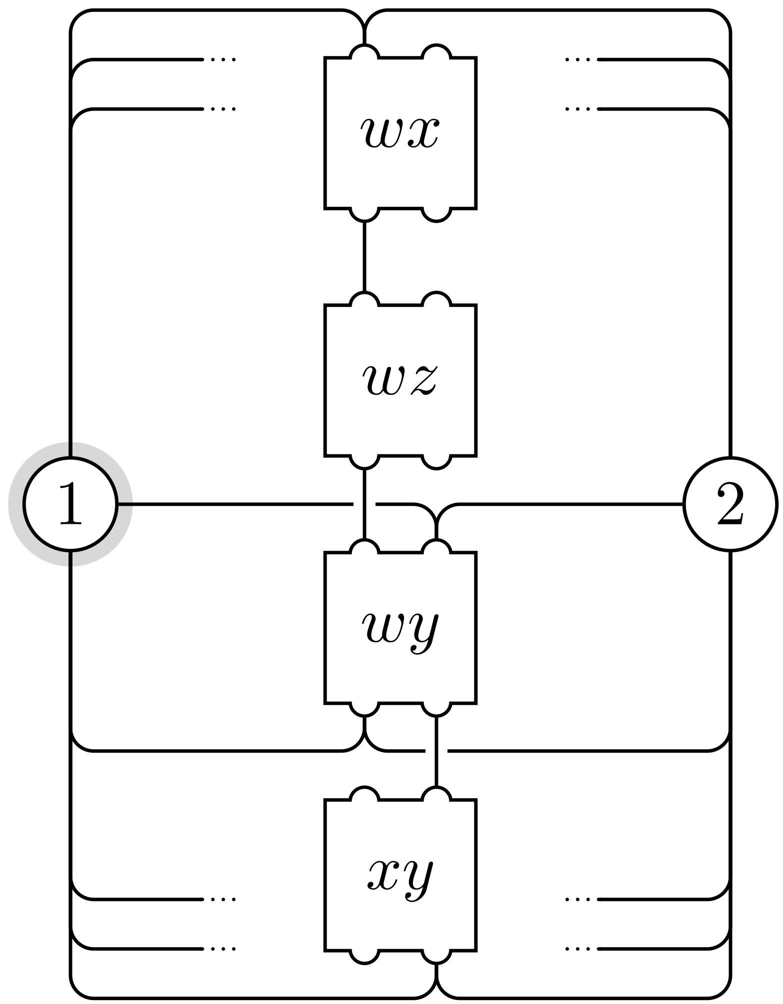
Dekker kanter xu, xv og xw



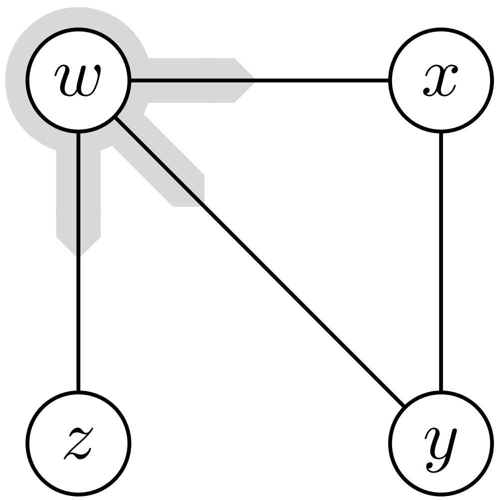
Sykel via tilsvarende widgets



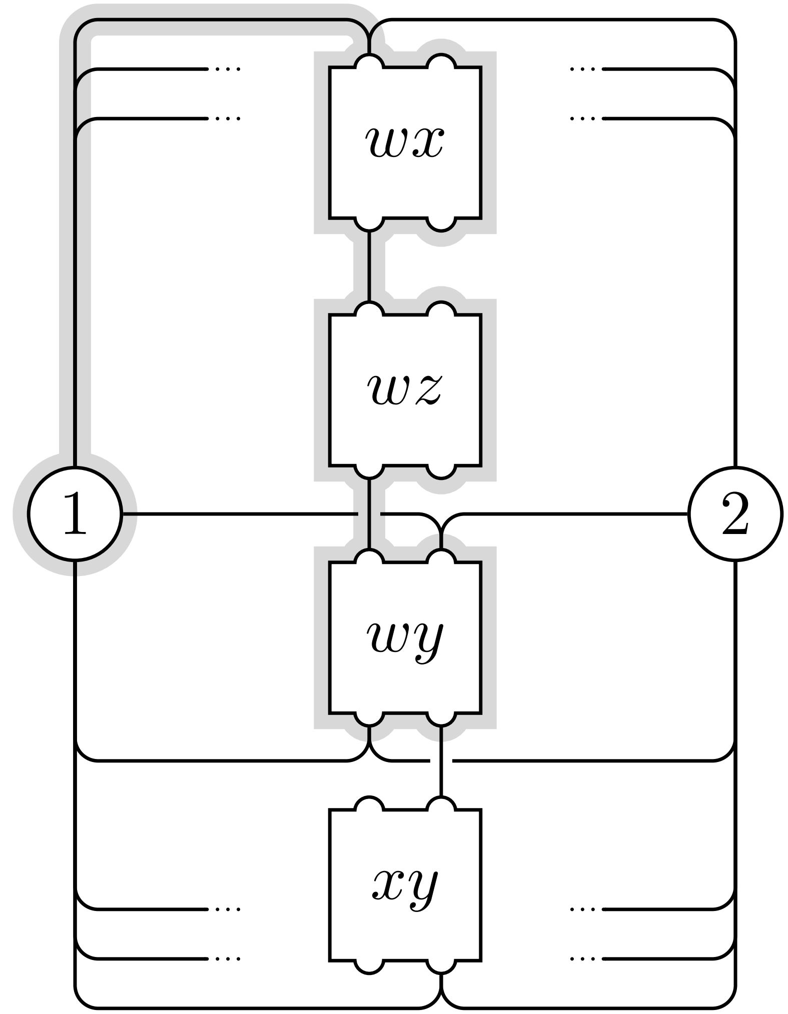
$k = 2$



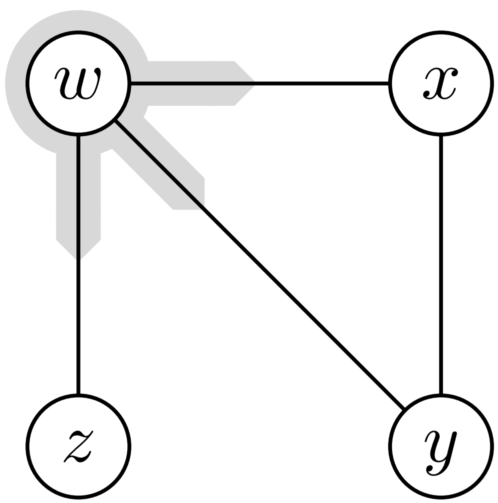
Finnes et k-dekke? $\Longleftrightarrow$ Finnes en hamiltonsykel?



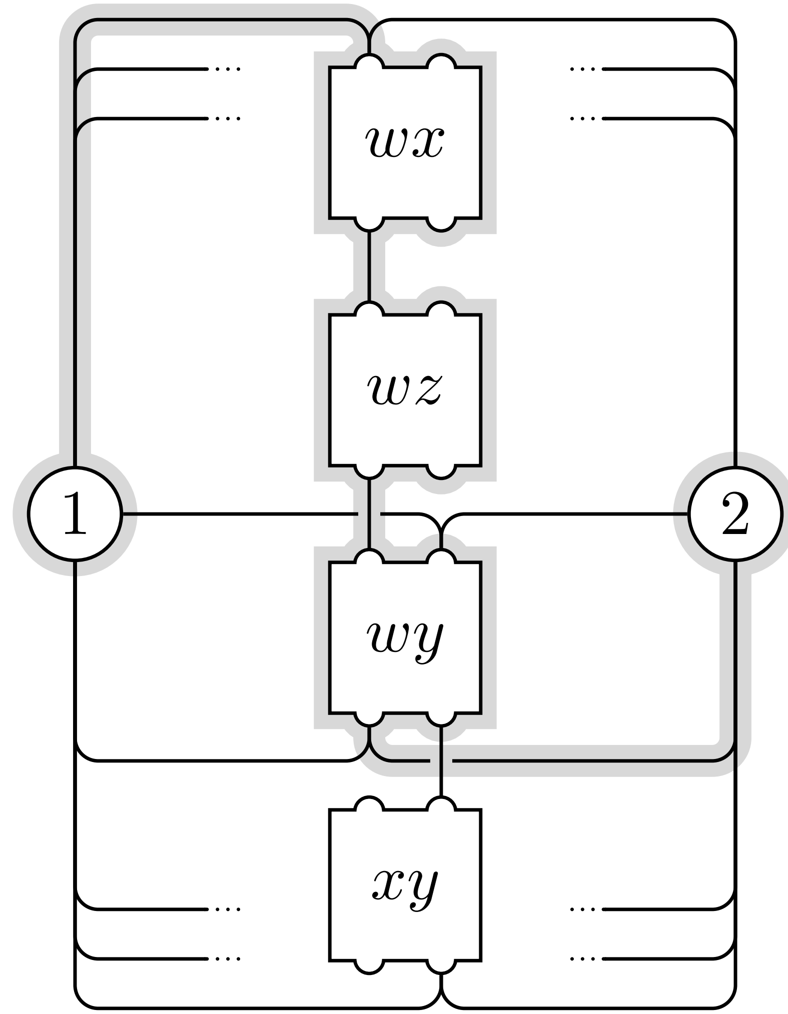
$k = 2$



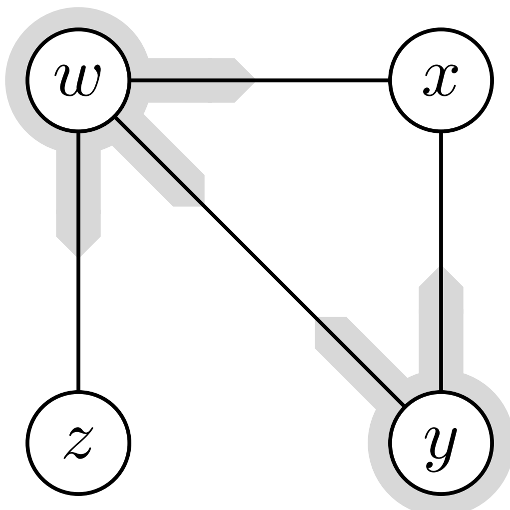
Finnes et k-dekke? $\iff$ Finnes en hamiltonsykel?



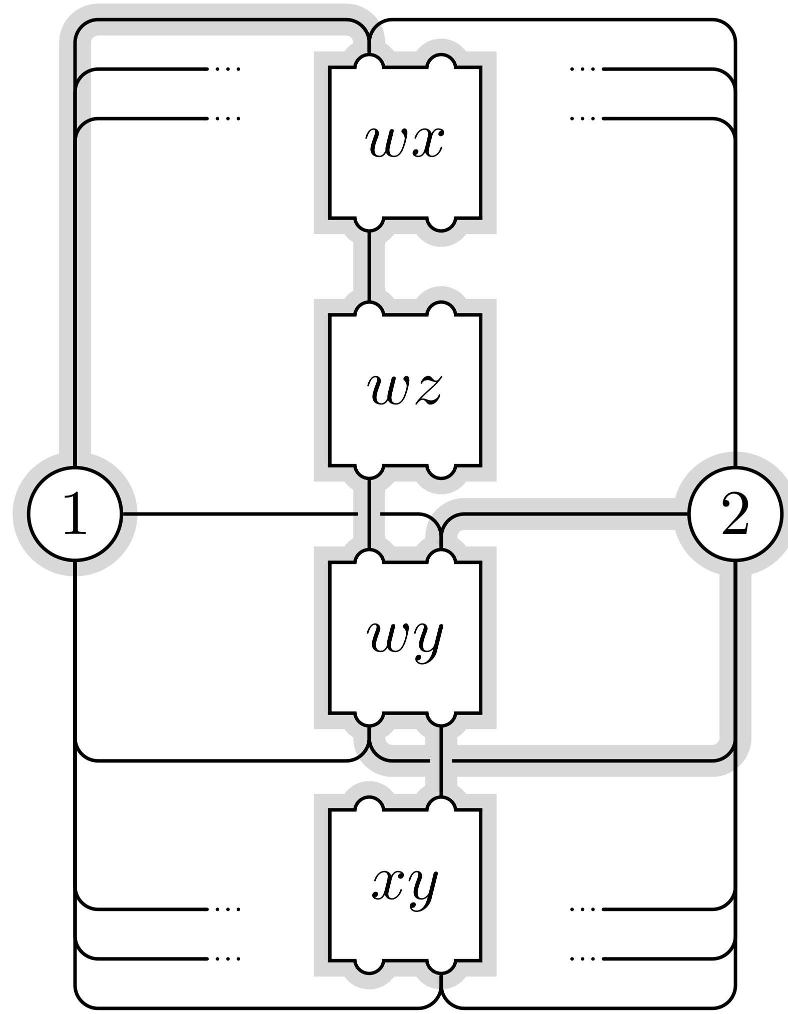
$k = 2$



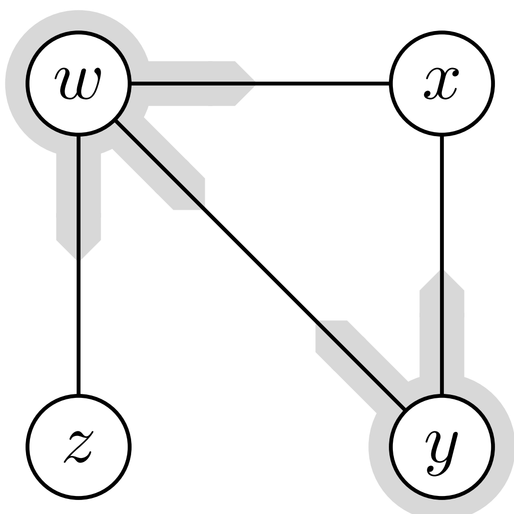
Finnes et k-dekke? $\iff$ Finnes en hamiltonsykel?



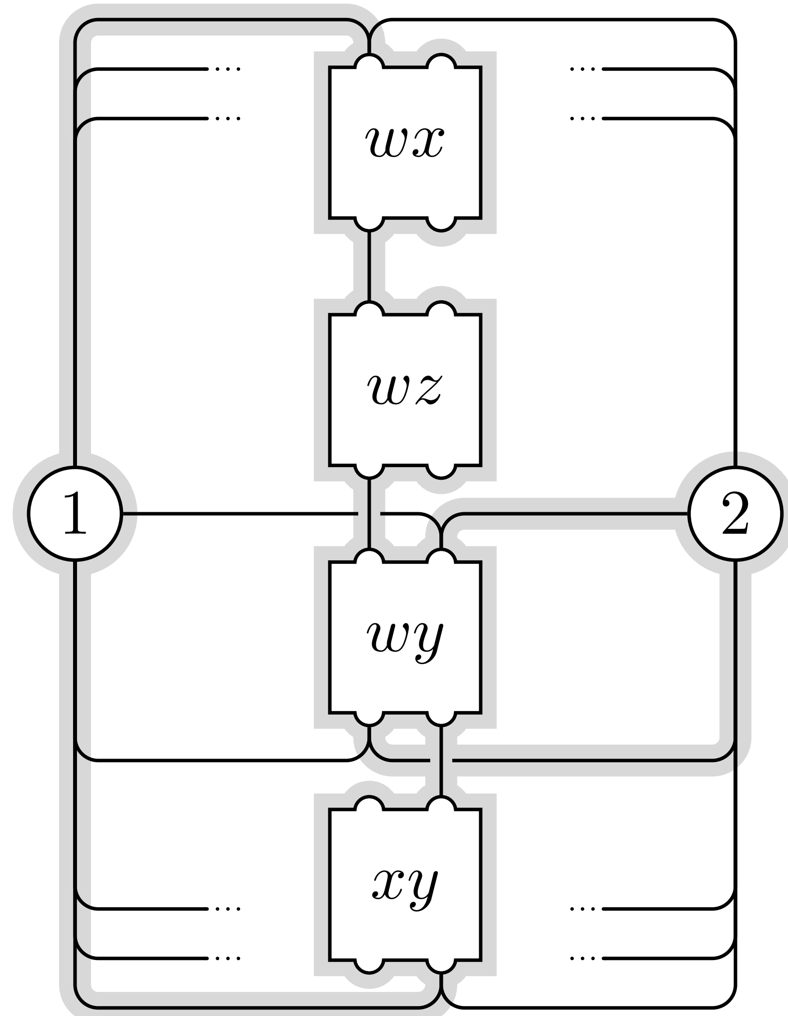
$k = 2$



Finnes et k-dekke? $\Longleftrightarrow$ Finnes en hamiltonsykel?



$k = 2$

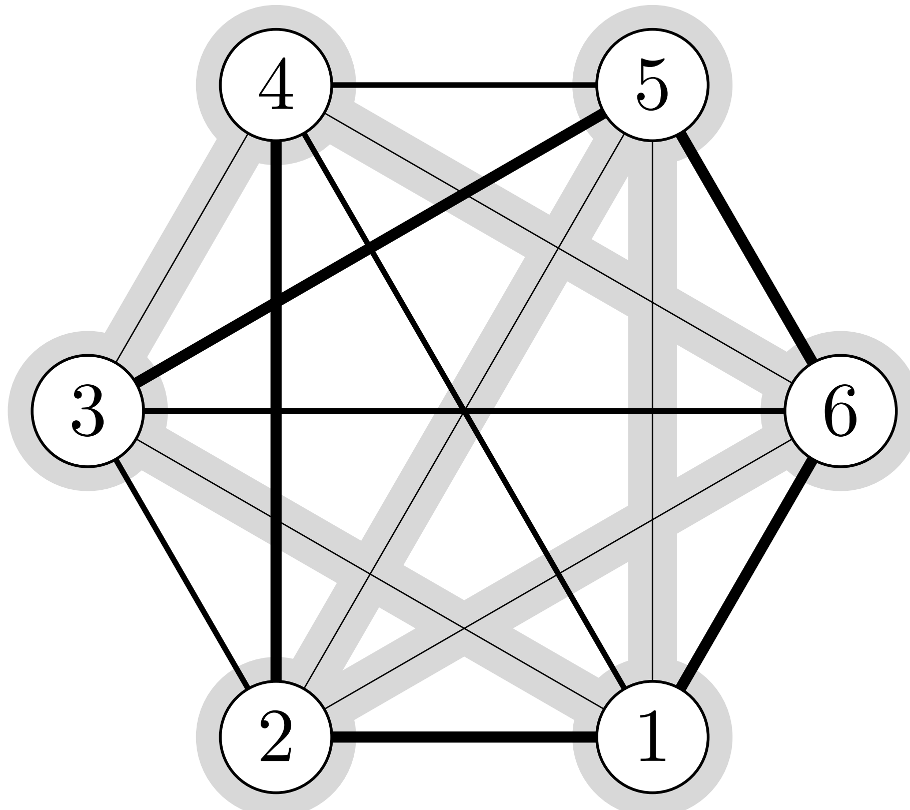


Finnes et k-dekke? $\Longleftrightarrow$ Finnes en hamiltonsykel?

Reduksjon videre til lengste enkle vei så vi på forrige gang

7:8

TSP



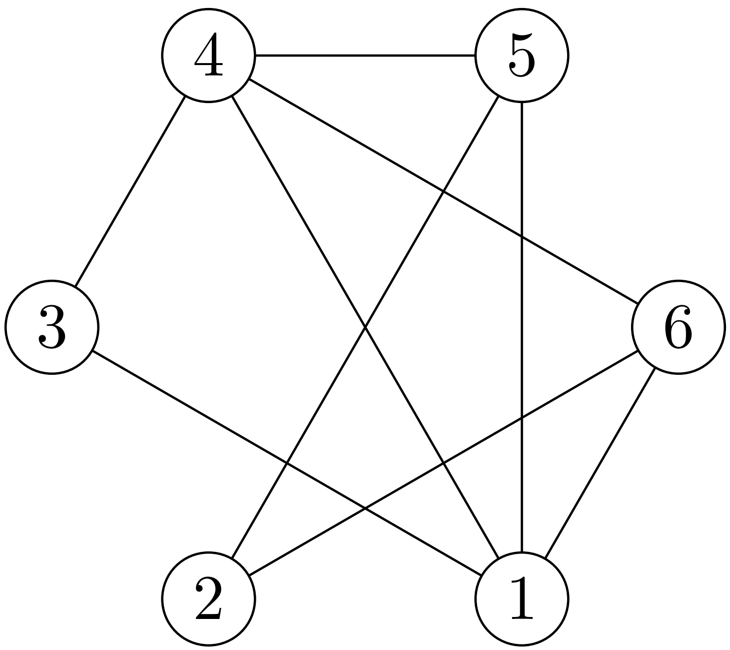
TSP

Instans: Komplett graf med vekter i $\mathbb{N}$ og et tall $k \in \mathbb{N}$

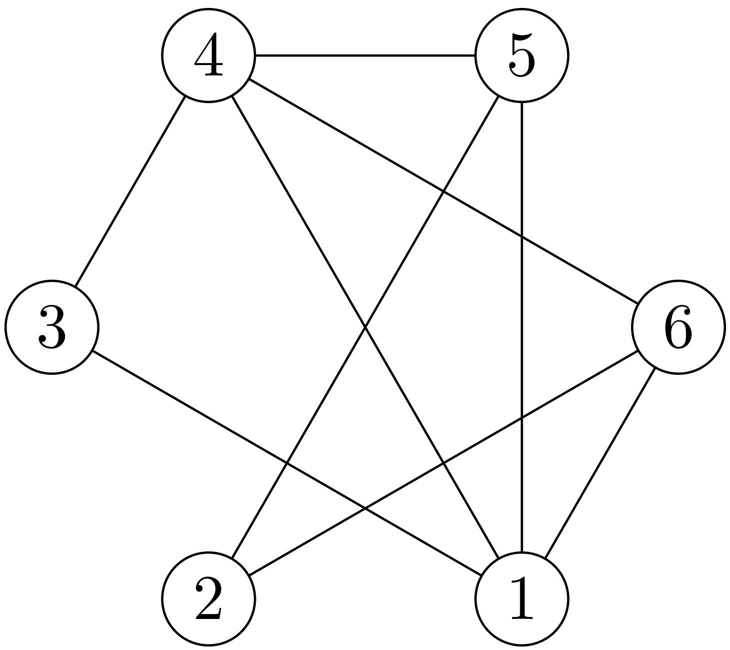
Spørsmål: Finnes det en rundtur med kostnad $\leqslant k$?

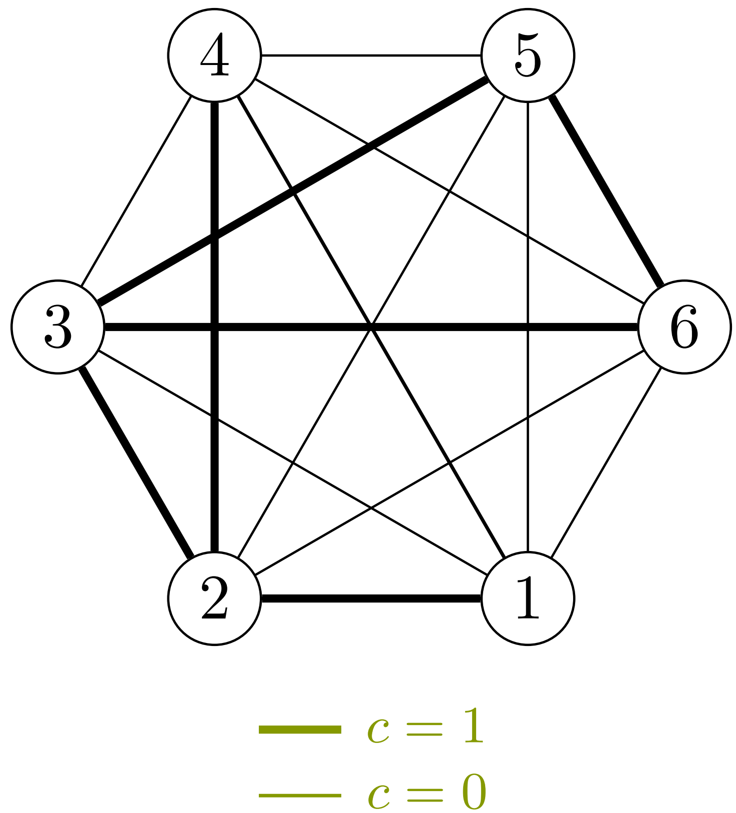
› Billigste hamiltonsykel!

- Billigste hamiltonsykel!
- Trenger bare gjøre originalgrafen veldig billig

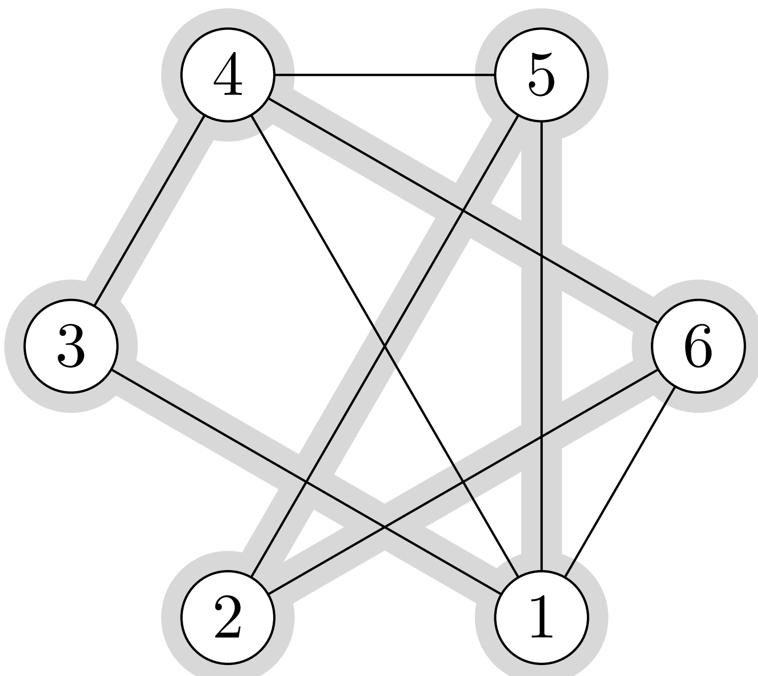


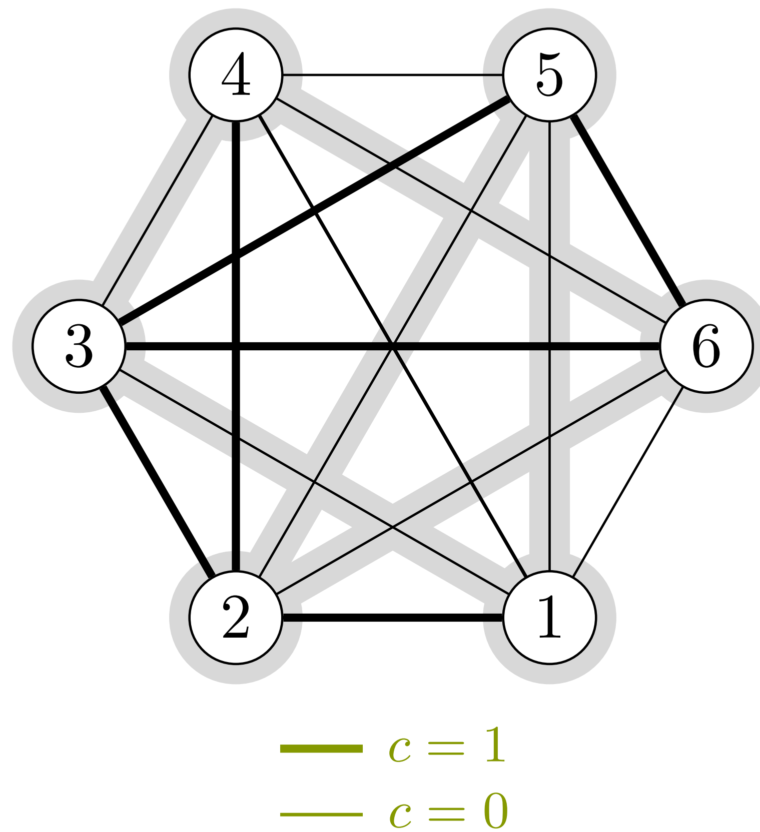
Finnes en hamiltonsykel? $\iff$ Finnes tur m/kostnad 0?





Finnes en hamiltonsykel? $\iff$ Finnes tur m/kostnad 0?

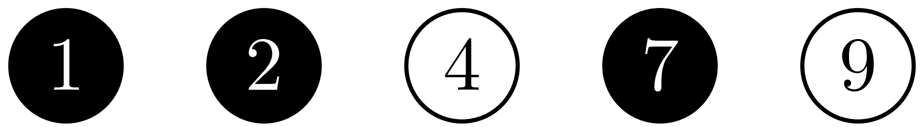




Finnes en hamiltonsykel? $\Longleftrightarrow$ Finnes tur m/kostnad 0?

8:8

SUBSET-SUM



SUBSET-SUM

Instans: Mengde positive heltall S og positivt heltall t

Spørsmål: Finnes en delmengde av S med sum t?

› Harmløse antagelser til reduksjonen:

- Harmløse antagelser til reduksjonen:
  - Ingen disjunksjon inneholder både x og $\neg x$

- Harmløse antagelser til reduksjonen:
  - Ingen disjunksjon inneholder både x og $\neg x$
  - Hver x forekommer i minst én disjunksjon

$$
\begin{aligned}
\phi = {} & (\ \ x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\
& (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\
& (\neg x_1 \vee \neg x_2 \vee \ \ x_3) \wedge \\
& (\ \ x_1 \vee \ \ x_2 \vee \ \ x_3)
\end{aligned}
$$

Kan ϕ være sann? $\iff$ Har S en delsum t?

$$S = \{\quad\}$$

$$t =$$

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

Kan ϕ være sann? $\iff$ Har S en delsum t?

$$
\begin{aligned}
\phi = {} & (\ \ x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\
& (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\
& (\neg x_1 \vee \neg x_2 \vee \ \ x_3) \wedge \\
& (\ \ x_1 \vee \ \ x_2 \vee \ \ x_3)
\end{aligned}
$$

1 2 3

$S = \{$

$\}$

$t =$

Vi innfører et siffer (en kolonne) per variabel

$$
\begin{aligned}
\phi = {} & (\ x_1 \lor \neg x_2 \lor \neg x_3) \land \\
& (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land \\
& (\neg x_1 \lor \neg x_2 \lor \ x_3) \land \\
& (\ x_1 \lor \ x_2 \lor \ x_3)
\end{aligned}
$$

$$
\begin{array}{rcccl}
 & 1 & 2 & 3 & \\
S = \{ & 1 & 0 & 0 & \\
 & 1 & 0 & 0 & \neg \\
 & 0 & 1 & 0 & \\
 & 0 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & \\
 & 0 & 0 & 1 & \neg \\
 & & & & \} \\
t = & & & &
\end{array}
$$

Ett tall per variabel og dens negasjon

$$\begin{aligned}\phi = (\ x_1 \lor \neg x_2 \lor \neg x_3) &\land \\ (\neg x_1 \lor \neg x_2 \lor \neg x_3) &\land \\ (\neg x_1 \lor \neg x_2 \lor x_3) &\land \\ (\ x_1 \lor x_2 \lor x_3) &\end{aligned}$$

$$\begin{array}{lccc l}
 & 1 & 2 & 3 & \\
S = \{ & 1 & 0 & 0 & \\
 & 1 & 0 & 0 & \neg \\
 & 0 & 1 & 0 & \\
 & 0 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & \\
 & 0 & 0 & 1 & \neg \\
 & & & & \} \\
t = & 1 & 1 & 1 &
\end{array}$$

Delsummen tvinger frem nøyaktig én sannhetsverdi per variabel

$$
\begin{aligned}
\phi = (\ & x_1 \lor \neg x_2 \lor \neg x_3) \land && (4)\\
(& \neg x_1 \lor \neg x_2 \lor \neg x_3) \land && (5)\\
(& \neg x_1 \lor \neg x_2 \lor x_3) \land && (6)\\
(& x_1 \lor x_2 \lor x_3) && (7)
\end{aligned}
$$

$$
\begin{array}{lccccccccc}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S = \{ & 1 & 0 & 0 & & & & & \\
 & 1 & 0 & 0 & & & & & \neg \\
 & 0 & 1 & 0 & & & & & \\
 & 0 & 1 & 0 & & & & & \neg \\
 & 0 & 0 & 1 & & & & & \\
 & 0 & 0 & 1 & & & & & \neg \\
 & & & & & & & & \} \\
t = & 1 & 1 & 1 & & & & &
\end{array}
$$

Vi innfører et siffer per disjunksjon

$$
\begin{aligned}
\phi = {} & (\ x_1 \lor \neg x_2 \lor \neg x_3) \land && (4)\\
& (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land && (5)\\
& (\neg x_1 \lor \neg x_2 \lor x_3) \land && (6)\\
& (\ x_1 \lor x_2 \lor x_3) && (7)
\end{aligned}
$$

$$
\begin{array}{lccccccccc}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & & & & & & & & \} \\
t = & 1 & 1 & 1 & & & & &
\end{array}
$$

For hver disjunksjon Er variabelen eller negasjonen med?

$$
\begin{aligned}
\phi = (\ & x_1 \vee \neg x_2 \vee \neg x_3) \wedge && (4)\\
(&\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge && (5)\\
(&\neg x_1 \vee \neg x_2 \vee x_3) \wedge && (6)\\
(\ & x_1 \vee x_2 \vee x_3) && (7)
\end{aligned}
$$

$$
\begin{array}{rccccccccl}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & & & & & & & & \}
\end{array}
$$

$$t = 1\quad 1\quad 1$$

Vi vil tvinge frem <u>minst én</u> sann literal per disjunksjon

$$
\begin{aligned}
\phi = (\ \ x_1 \lor \neg x_2 \lor \neg x_3) &\land \quad (4)\\
(\neg x_1 \lor \neg x_2 \lor \neg x_3) &\land \quad (5)\\
(\neg x_1 \lor \neg x_2 \lor \ \ x_3) &\land \quad (6)\\
(\ \ x_1 \lor \ \ x_2 \lor \ \ x_3) & \quad\ \ \ (7)
\end{aligned}
$$

$$
\begin{array}{llllllllll}
 & & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S & = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & & & & & & & & & \} \\
t & = & 1 & 1 & 1 & & & & &
\end{array}
$$

Sum per disjunksjonskolonne skal altså være 1, 2 eller 3

$$
\begin{aligned}
\phi = (\ \ x_1 &\vee \neg x_2 \vee \neg x_3) \wedge && (4)\\
(\neg x_1 &\vee \neg x_2 \vee \neg x_3) \wedge && (5)\\
(\neg x_1 &\vee \neg x_2 \vee \ \ x_3) \wedge && (6)\\
(\ \ x_1 &\vee \ \ x_2 \vee \ \ x_3) && (7)
\end{aligned}
$$

$$
\begin{array}{lcccccccl}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \} \\
t = & 1 & 1 & 1 & & & & &
\end{array}
$$

Vi innfører slakkverdier, som fyller ut det som mangler

$$
\begin{aligned}
\phi = (\ \ x_1 &\lor \neg x_2 \lor \neg x_3) \land && (4)\\
(\neg x_1 &\lor \neg x_2 \lor \neg x_3) \land && (5)\\
(\neg x_1 &\lor \neg x_2 \lor \ \ x_3) \land && (6)\\
(\ \ x_1 &\lor \ \ x_2 \lor \ \ x_3) && (7)
\end{aligned}
$$

$$
\begin{array}{rccccccccl}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \} \\
t = & 1 & 1 & 1 & 4 & 4 & 4 & 4 &
\end{array}
$$

Men: Nå kan vi få slakk på opptil 3; må kreve at summen er 4

$$
\begin{aligned}
\phi = (\ & x_1 \vee \neg x_2 \vee \neg x_3) \wedge && (4)\\
(&\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge && (5)\\
(&\neg x_1 \vee \neg x_2 \vee x_3) \wedge && (6)\\
(& x_1 \vee x_2 \vee x_3) && (7)
\end{aligned}
$$

$$
\begin{array}{rccccccccl}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
\mathrm{S} = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \} \\
t = & 1 & 1 & 1 & 4 & 4 & 4 & 4 &
\end{array}
$$

Kan ϕ være sann? $\iff$ Har S en delsum t?

$$\phi = (x_1 \lor \neg x_2 \lor \neg x_3) \land \quad (4)$$
$$(\neg x_1 \lor \neg x_2 \lor \neg x_3) \land \quad (5)$$
$$(\neg x_1 \lor \neg x_2 \lor x_3) \land \quad (6)$$
$$(x_1 \lor x_2 \lor x_3) \quad (7)$$

$$x_1 = 1, \; x_2 = 0$$

$$\begin{array}{rccccccccl}
 & 1 & 2 & 3 & (4) & (5) & (6) & (7) & \\
S = \{ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & \neg \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \neg \\
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \\
 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \neg \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \} \\
t = & 1 & 1 & 1 & 4 & 4 & 4 & 4 &
\end{array}$$

Kan ϕ være sann? $\iff$ Har S en delsum t?

Lett å redusere videre til det binære ryggsekkproblemet

Bare la vekt være lik verdi

1. CIRCUIT-SAT
2. SAT
3. 3-CNF-SAT
4. CLIQUE
5. VERTEX-COVER
6. HAM-CYCLE
7. TSP
8. SUBSET-SUM