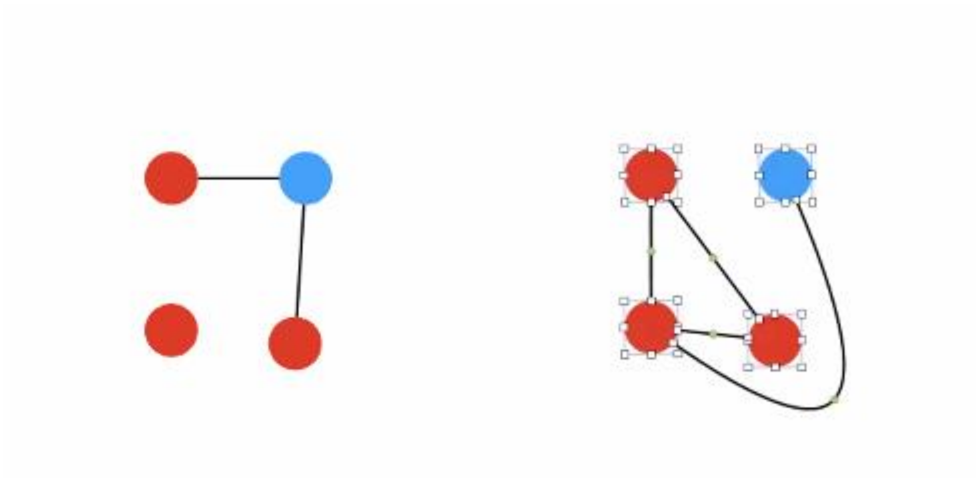
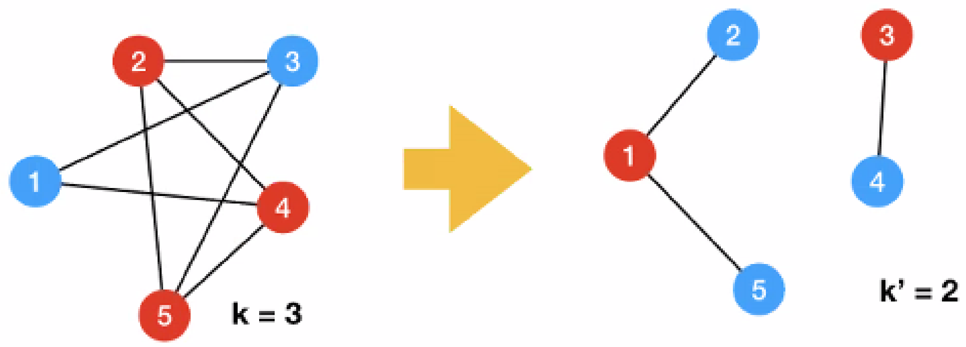
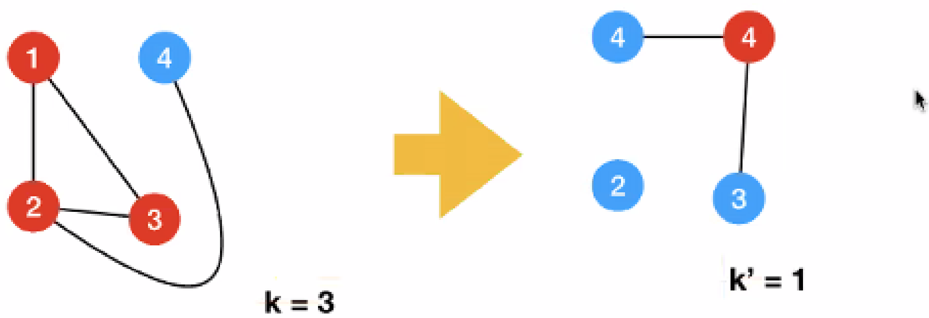


```
Si G = (S,A) a une clique X ⊆ S de taille k

pour tout x,y ∈ X avec x ≠ y on a {x,y} ∈ A

donc dans le complementaire {x,y} ∉ A
donc X est un ensemble indépendent dans le
complementaire G' de taille k


Vice-versa, si G' a un ensemble indépendent X ⊆ S de
taille k, alors pour tout x,y avec x ≠ y,
on a {x,y} ∉ A

donc, dans le complémentaire G on a {x,y} ∈ A
donc X c'est bien une clique, et la taille est
toujours k.
```

**Clique ≤ Node Cover**



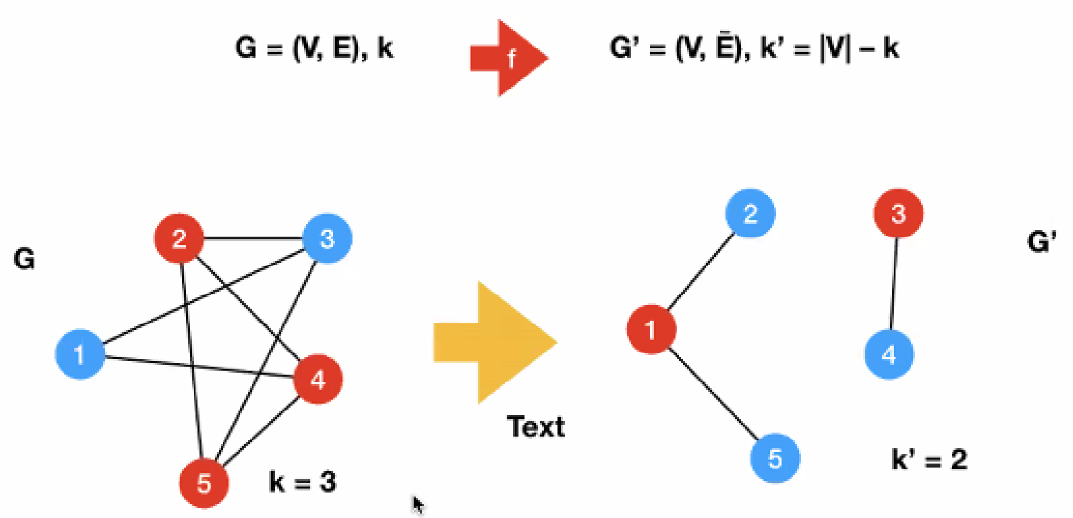

$$G = (V, E), k \Rightarrow G' = (V, \bar{E}), k' = |V| - k$$

```
G a une clique de taille k ssi
G' a une node cover de taille k'

1) G a une clique de taille k →
   G' a une node cover de taille k'

appelons X ⊆ V la clique
X est une zone vide (ou ensemble indépendant) dans G'
si {u,v} ∈ E' alors ils ne sont pas tous les deux
dans X, c-à-d, soit u ∉ X, soit v ∉ X, soit les deux

chaque arête a une extremité qui n'est pas dans X
elle est dans V − X

du coup, V − X est une node cover
|V−X| = |V| − k = k'
```

Les 2 disent la même chose, mais à droite c'est plus propre

```
2) G' a une node cover de taille k' →
   G a une clique de taille k

soit Y ⊆ V une node cover de taille k' dans G'

pour toute arête {u,v} ∈ E', soit u ∈ Y, soit v ∈ Y
(soit les deux), ce n'est jamais le cas que u, v ∉ Y

dans le graphe complementaire de G', c-à-d G

soit u, v ∉ Y, soit x, y ∈ V − Y
est-ce que {u,v} ∈ E' ? no ! {u,v} ∉ E'

donc, dans G, on a {u,v} ∈ E
```

```
2) G' a une node cover de taille k' →
   G a une clique de taille k

soit Y ⊆ V une node cover de taille k' dans G'

on veut montrer que V − Y est une clique dans G

soit u,v ∈ V − Y

alors {u,v} ∉ E', mais alors {u,v} ∈ E\

du coup V − Y est
```




**G' a une node cover de taille k' →**
**G a une clique de taille k = |V| − k'**

Non déterministe

```
algo node-cover(G, k):
    n := |V|
    X = tableau de booléens de taille n
    pour i := 1 à n faire
        X[i] := deviner(0, 1)
    taille := 0
    pour i := 1 à n faire
        taille := taille + X[i]
    si taille ≠ k alors
        rejeter
    pour tout {u,v} ∈ E faire
        si non X[u] et non[v] alors
            rejeter
    accepter
```

Déterministe

```
algo verificateur-node-cover(G, k, X):
    n := |V|
    taille := 0
    pour i := 1 à n faire
        taille := taille + X[i]
    si taille ≠ k alors
        rejeter
    pour tout {u,v} ∈ E faire
        si non X[u] et non[v] alors
            rejeter
    accepter
```

---

```
algo verificateur-set-packing(S₁,…,Sₗ):
   pour i := 1 à l faire
      pour j := i+1 à l faire
         si Sᵢ ∩ Sⱼ ≠ ∅ alors
            rejeter
   accepter
```

Ins = insemble, ind = indépendant

réduction ENS-IND ≤ SET-PACKING:

$G = (V,E)$ a ins. ind. $\{u,v,w\}$ de taille $k$

$\{S_v : v \in V\}$
$S_v = \{\{u,v\} \in E\}$

S_v, S_u, S_w tous disjoints, y en a $\ell = k$

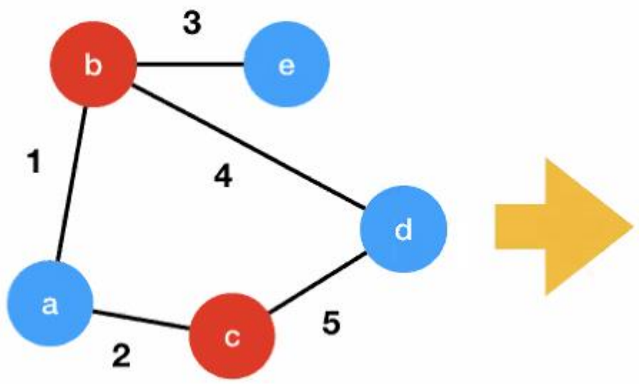
(sinon $S\ v \cap S\ u = \{\{u,v\}\}$)
la réduction est

$$G = (V,E) \overset{f}{\mapsto} \{S_v : v \in V\} \text{ où } S_v = \{\{u,v\} \in E\}$$

G a ins.ind. de taille $k \leftrightarrow$ il y a $\ell$ ensembles mut.disjoints

si G a ins.ind. de taille $k$, disons $\{v_1,...,v_k\}$
alors prenon $S_{v_1}, ..., S_{v_k}$
ils sont disjoints mutuellement

vice-versa, si $S_{v_1}, ..., S_{v_k}$ sont disjoints mutuellement
-> il n'y a pas d'arête $\{u,v\}$ pour chaque $u,v \in \{v_1,...,v_k\}$
-> $\{v_1,...,v_k\}$ est ins.ind.




on veut recouvrir {1,2,3,4,5}

Sa = {1,2}
Sb = {1,3,4}
Sc = {2,5}
Sd = {4,5}
Se = {3}

b couvre {a,b}, {b,d}, {b,e}

c couvre {a,c}, {c,d}

Sb U Sc = Sa U Sb U Sc U Sd U Se
= {1,2,3,4,5}

réduction NODE-COVER ≤ SET-COVERING

$f(G=(V,E), k) = (\{S_v : v \in V\}, \ell = k)$
où $S_v = \{\{u,v\} \in E\}$

$\varphi = (x \lor y \lor z) \land (\neg x \lor \neg y \lor z)$
1 0 0 0 1 0

$x = 0, y = 0, z = 1$
$x = 1, y = 0, z = 0$ satisfait 1 littéral par clause

3SAT ≤ Exactly-1 3SAT

$\varphi = C_1 \land C_2 \land \cdots \land C_m$

m clauses, n variables

$C_i = (\ell_{i1} \lor \ell_{i2} \lor \ell_{i3})$

où chaque $\ell_{ij}$ est soit une variable,
soit la négation d'une variable

$T_i = (\neg\ell_{i1} \lor x_i \lor y_i) \land (\ell_{i2} \lor y_i \lor z_i) \land (\neg\ell_{i3} \lor z_i \lor w_i)$

$\varphi' = T_1 \land T_2 \land \cdots \land T_m$

3m clauses, n+4m variables

PARENTHESE

$(x \lor y) \to (x \lor y \lor a) \land (x \lor y \lor \neg a)$
$(x) \to (x \lor a \lor b) \land (x \lor \neg a \lor b) \land (x \lor a \lor \neg b) \land (x \lor \neg a \lor \neg b)$

faut prouver que φ est satisfaisable ssi φ' est satisfaisable par une affectation qui rend vrai exactement un littéral par clause

φ sat -> φ' sat avec un lit. par clause vrai
prenons $C_i = (\ell_{i1} \lor \ell_{i2} \lor \ell_{i3})$
supposons que $\ell_{i2} = 1$
$T_i = (\neg\ell_{i1} \lor x_i \lor y_i) \land (\ell_{i2} \lor y_i \lor z_i) \land (\neg\ell_{i3} \lor z_i \lor w_i)$
0 1 0 0 0 1 0 1 0
si $\ell_{i2} = 1$, $y_i = 0$, $z_i = 0$
si $\neg\ell_{i1} = 1$, c-à-d $\ell_{i1} = 0$, alors $x_i = 0$
si $\neg\ell_{i1} = 0$, c-à-d $\ell_{i1} = 1$, alors $x_i = 1$

si $\ell_{i2} = 0$ et $\ell_{i1} = 1$ et $\ell_{i3} = 1$
$x_i = 1$, $y_i = 0$, $z_i = 1$, $w_i = 0$

si $\ell_{i2} = 0$ et $\ell_{i1} = 1$ et $\ell_{i3} = 0$
$z_i = 0$, $w_i = 0$, $y_i = 1$, $x_i = 0$

si $\ell_{i2} = 0$ et $\ell_{i1} = 0$ et $\ell_{i3} = 1$
$z_i = 1$, $w_i = $ , $y_i = 0$, $x_i = 0$

```
A -> B même chose que ¬B -> ¬A

φ' sat avec un lit. par clause vrai -> φ sat
φ non sat -> φ' non sat avec un lit. par clause vrai
```

```
φ = C_1 ∧ C_2 ∧ ⋯ ∧ C_m

pour chaque affectation, au moins C_i est fausse
C_i = (ℓ_i1 v ℓ_i2 v ℓ_i3)

ℓ_i1 = 0, ℓ_i2 = 0, ℓ_i3 = 0

T_i = (¬ℓ_i1 v x_i v y_i) ∧ (ℓ_i2 v y_i v z_i) ∧ (¬ℓ_i3 v z_i v w_i)
         1                    0                     1

y_i = 0
z_i = 0
```

```
(x v y v z) ∧ (¬x v ¬y v z)
 0   1   0      1   0    0
x=0, y=1, z=0

/
|  x + y + z = 1              0 + 1 + 0 = 1
|  −x − y + z = −1            0 − 1 + 0 = −1
\
```