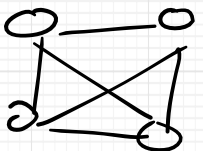


$\Pi_1 =$ "Soit G un graphe non orienté et $k \in \mathbb{N}$, est-ce que G possède une clique de taille k ?"

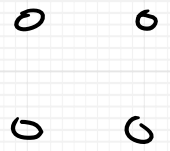
Pb de la Clique MAX

Clique :  graphe complet

$\Pi_2 =$ "Soit G^H un graphe non orienté et $k \in \mathbb{N}$, est-ce que G^H possède un stable de taille k ?"

Pb du stable MAX

On a juste changé les noms de G et k pour éviter d'avoir les mêmes noms dans les deux problèmes.

Stable :  graphe sans arête

Mq $\Pi_1 \leq \Pi_2$

$R(\text{graphe complet}, 3) = (\text{graphe sans arête}, 3)$

Soit R l'algo qui transforme un graphe G en son complémentaire H et conserve la valeur k pour l'entier l

$$(u, v) \in G \Leftrightarrow (u, v) \notin H$$

Mq R est une réduction polynomiale de Karp de Π_1 vers Π_2

1) R est polynomial

car la complexité de R est $O(m^2)$ où m est le nb de nœuds

pe

2) R prend en entrée des instances de Π_1 et en sortie des instances de Π_2
par définition de R

3) $\forall I \in \mathcal{L}_1^1 \Leftrightarrow R(I) \in \mathcal{L}_2^2$

def de $\mathcal{L}_1^1$ Soit $I \in \mathcal{L}_1^1$ $\Leftrightarrow \exists (c_1, c_2, \dots, c_k)$ nœuds du graphe G qui forment une clique $\Leftrightarrow \forall i, j \leq k \quad (c_i, c_j) \in G$ def de Clique	def de Complémentaire $\Leftrightarrow \forall i, j \leq k \quad (c_i, c_j) \notin H$ def de stable $\Leftrightarrow \exists (c_1, \dots, c_k)$ nœuds de H qui forment un stable def de $\mathcal{L}_2^2$ $\Leftrightarrow R(I) \in \mathcal{L}_2^2$	car on a inséré les arêtes et non arêtes de G
--	---	---

Sachant que Π_1 est NP-complet
alors Π_1 est NP-difficile

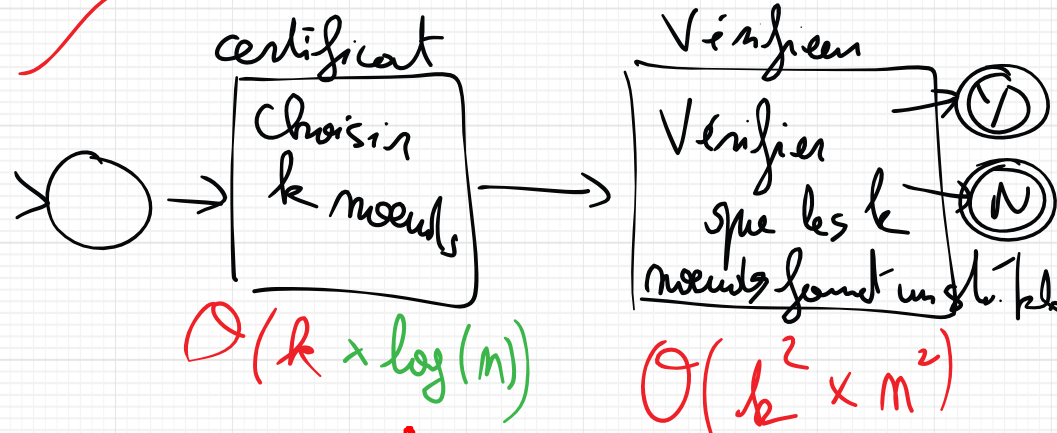
or $\Pi_1 \leq \Pi_2$

donc Π_2 est NP-difficile

(on peut prouver que 3SAT $\leq$ Clique MAX et que Clique MAX est NP)

Π_2 est NP-Complet.

or Π_2 est NP

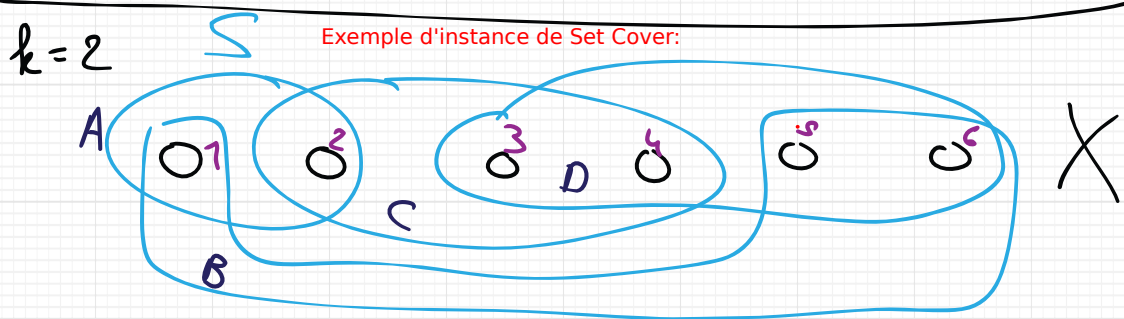


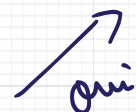
Pour chaque couple (u, v) de noeud choisi (k^2)
on regarde toutes les arêtes de G (au plus n^2)
qu'on compare à (u, v)
On devrait rajouter un $\log(n)$ pour dire qu'on compare des mots
de taille $\log(n)$

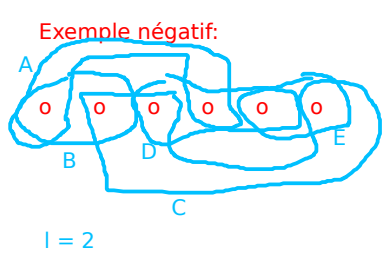
Toujours cette histoire qu'on doit gérer des
identifiants de noeuds de taille $\log(n)$

$\Pi_3 =$ "Soit X un ensemble
Set Cover) et $S \subseteq 2^X = \mathcal{P}(X)$
 (l'ensemble des
 sous-ensembles de X)
 et $k \in \mathbb{N}$

$\exists ? \mathcal{C} / \mathcal{C} \subset S ; \bigcup_{C \in \mathcal{C}} C = X,$
 $|\mathcal{C}| \leq k$



en me gardant que 2 ensembles , puis je  oui
 recouvrir X ?



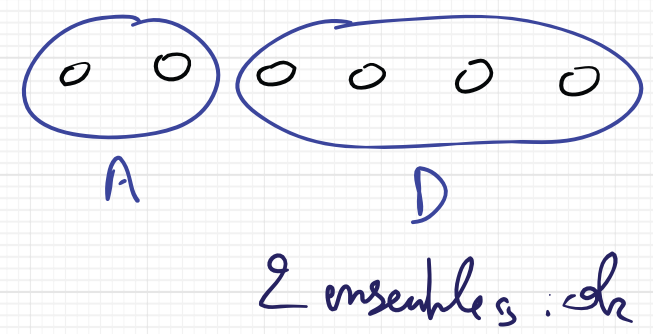
Pour tout couple d'ensemble, X n'est pas couvert.

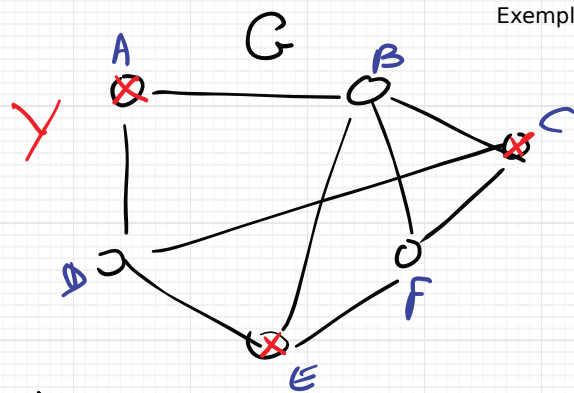
$\Pi_4 =$ "Soit G un graphe non orienté
Pb de Steiner) et Y des noeuds de G
 et $l \in \mathbb{N}$

$\exists ?$ un arbre incluant dans G qui
 relie Y avec moins de l arêtes?"

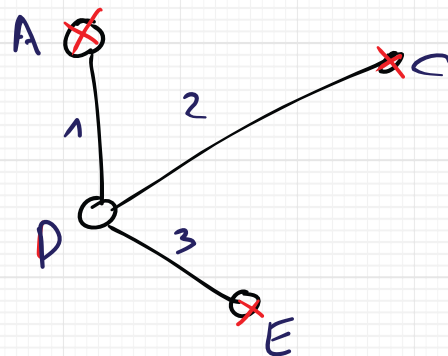
A priori, un arbre n'est pas nécessaire. Dire "un sous-graphe de G qui relie ..." suffirait.
 En effet, si j'ai un sous-graphe G' qui relie Y , je peux supprimer des arêtes pour trouver un

arbre couvrant de G' . Donc si G' existe alors il existe nécessairement un arbre avec au plus
 l arêtes qui relie Y . Inversement, si un arbre est un sous graphe, donc si un arbre T existe, alors
 un sousgraphe G' existe. Donc les deux problèmes sont équivalents.





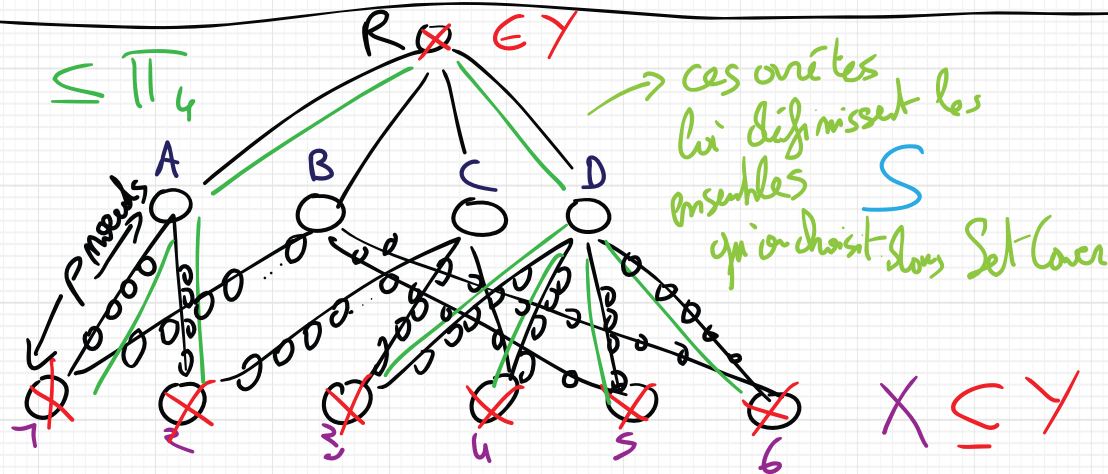
→



3 arêtes: ok

$$l = 3$$

$$m_1 \pi_3 \leq \pi_4$$



$$l = k + |X|/p$$

La valeur de p est définie plus loin dans la démonstration. Tout ce qu'on sait pour l'instant c'est que p est "suffisamment grand" pour rendre les chaînes du bas trop chère pour être toutes sélectionnées dans la solution.

En noir, on a dessiné ce que la réduction donnée au slide suivant produirait sur l'exemple de Set Cover de la slide précédente.
En vert, on a représenté un arbre couvrant les noeuds de Y .

On voit que cet arbre a 2 propriétés intuitives :

- chaque noeud de X (3e niveau) est relié à un noeud de S (2e niveau) par une chaîne de p arêtes
- chaque noeud de S qui est relié à T est un ensemble qui a été choisi dans Set Cover (d'où la remarque à droite du dessin)

$$X \subseteq Y$$

$R: \forall x \in X$ créer un nœud x dans G $O(|X|)$

$\forall s \in S$ créer un nœud s dans G $O(|S|)$

relier $\forall x \in S$ s à x avec une chaîne de P nœuds $O(P \cdot |S|)$

→ Valeur? on verra (P = k + 1 of slide 8)

Ajouter un nœud r et le relier à tout nœud $s \in S$ $O(|S|)$

$$l = k + |X| \cdot P \quad O(\log(k) + \log(|X|) + \log(P))$$

$$Y = X \cup \{r\} \quad O(X)$$

→ R polynomial (si P est pas trop grand)

Par définition de R , les entrées de R sont des instances de Π_3 et les sorties sont des instances de Π_4

$$\text{Mq } I \in \mathcal{L}_Y^3 \Leftrightarrow R(I) \in \mathcal{L}_Y^4$$

$$\Rightarrow \text{Si } I \in \mathcal{L}_Y^3$$

$$\exists \mathcal{C} \subseteq S \quad |\mathcal{C}| \leq k$$

$\mathcal{C}$ coupe X

Construisons l'arbre T suivant

- $\forall s \in \mathcal{C}$ relie r et s ↗ $|\mathcal{C}|$ arêtes
 - $\forall x \in X, \exists s \in \mathcal{C}$ et $x \in S$ (def de $\mathcal{C}$) ↗ $|X| \cdot p$ arêtes
- on relie x à s (avec la chaîne de p nœuds)
- (l'arbre vert sur le dessin slide 5)

Il faut maintenant prouver que T est bien une solution qui répond positivement à la question du pb de Steiner: Il relie Y ; Il a moins de l arêtes

$$\begin{aligned} |T| &= |\mathcal{C}| + |X| \cdot p \\ &\leq k + |X| \cdot p \\ &\leq l \end{aligned}$$

on a un arbre avec moins de l arêtes.

$$\forall y \in Y$$

$$\text{Soit } y = r$$

sinon $y \in X$, donc $\exists s \in \mathcal{C}$ ↖ s

ty y est relié à s dans T

et $\forall s \in \mathcal{C}$ s est relié à r dans T

donc T relie les nœuds de Y à r , donc relie Y .

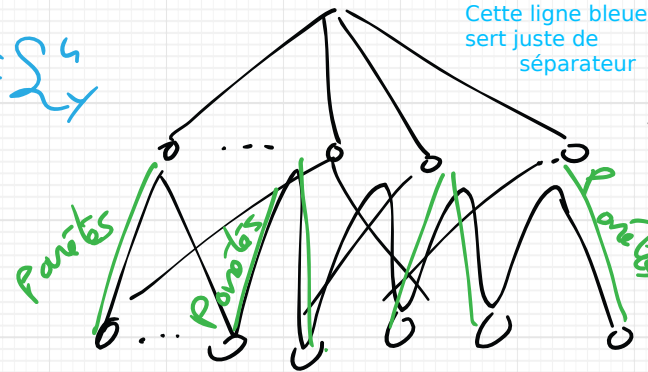
("donc T relie tous les nœuds de Y à r ; donc T relie tous les nœuds de Y entre eux")

$\Leftrightarrow \exists T \subseteq G \text{ tq}$
 $|T| \leq l$
 $T \text{ relie } Y$

$\exists T \subseteq G \text{ tq}$
 $|T| \leq l$
 $T \text{ relie } Y$

$\rightarrow T \text{ relie } Y,$
 alors $\forall x \in X$

$\exists$ une chaîne de T qui relie x à r
 cette chaîne a nécessairement p noeuds
 distincts pour chaque $x \rightarrow$ au moins
 $p \cdot |X|$ arêtes de ce type



T ne peut contenir
 $l = (|X| + 1)p$ arêtes

1 chaîne de T
 par x
 $\rightarrow p \cdot |X|$ arêtes

Ces k
 arêtes
 ne peuvent
 être une
 chaîne de T
 de p
 noeuds reliant
 un noeud x à un
 noeud s .

$$\text{or } |T| \leq l = |X| \cdot p + k$$

donc T a au plus k arêtes
 qui ne sont pas ces chaînes

C est "l'ensemble des noeuds
 de S reliés à r dans T "

$$\text{Soit } C = \{s \in S / (s, r) \in T\}$$

$|C| \leq k \rightarrow$ Conclut qu'il ne reste que k arêtes
 au plus pour relier r
 à lui-même
 ligne de G

$\exists$ exactement
 $|X|$ chaînes de T
 reliant la ligne
 X à la ligne
 S du
 graphe

Dans cette partie, on dispose de T et il faut construire C ; il faut bien se rendre compte qu'en disposant de T , on dispose d'une structure puissante ("qui relie Y ") et qu'il faut mettre à profit cette puissance pour construire C .

On ne peut pas supposer de base que T ressemble à ce qui a été dessiné slide 6. On voudrait que T ressemble à ça pour pouvoir définir une solution pour Set Cover facilement mais rien ne le garanti. On sait juste que T relie Y et que T a moins de l arêtes.

La première étape consiste donc à montrer que, en respectant ces deux propriétés, T ressemble bien à ce qu'on a imaginé; c'est tout l'objectif de ce slide, montrer que chaque noeud x du 3e niveau est relié avec exactement une chaîne de p arêtes à un noeud s du 2e niveau. (propriété rose à droite).

Une fois qu'on sait ça, on peut démontrer (au slide suivant) que le seul moyen de T de relier Y en moins de l arêtes est de décrire une solution pour Set Cover.

Ma $\mathcal{C}$ couvre X

autrement dit :

soit $x \in X$, ma $\exists s \in \mathcal{C} / x \in s$

$\Leftrightarrow$ ma $\exists s$ relié à $x / x \in s$

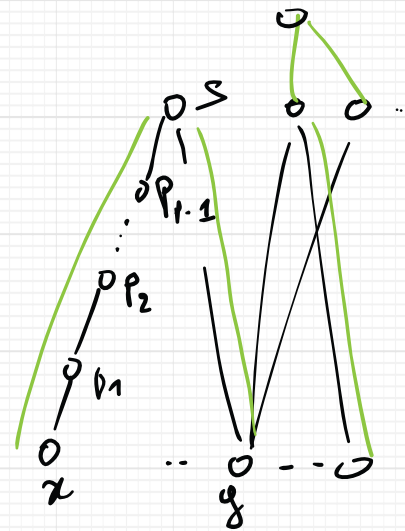
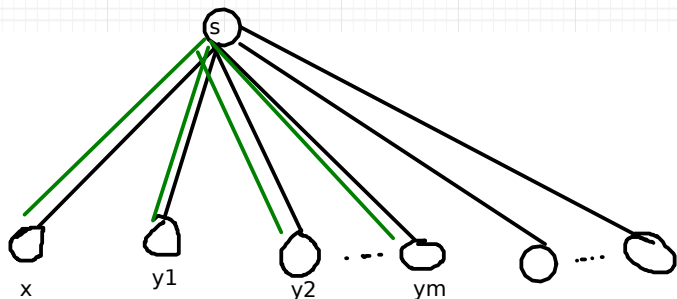
Soit $(p_1 p_2 \dots p_{p-1})$ la chaîne ^{de T} _{unique} reliant x à un noeud s de G

(celle chaîne existe et est unique) Ma s est relié à x dans T

$S: (n, s) \notin T$

puisque n est relié à x dans T , $\exists$ une chaîne qui relie s à n . Si cette chaîne n'est pas (s, n) alors $\exists$ une autre chaîne qui relie s à un autre noeud $y \in X$ dans T

pour être tout à fait exact, il existe éventuellement plus qu'une autre chaîne qui relie s à plusieurs noeuds de X dans T ; le raisonnement est exactement le même quand on considère plusieurs noeuds.



donc $\mathcal{C}$ couvre X
 $\downarrow$
 donc $\exists \mathcal{C} \subseteq S$
 "tel que"
 $\forall x |x| \leq k$
 $\mathcal{C}$ "couvre" X
 donc $I \in \mathcal{L}^3$

cette chaîne est donc l'*unique* chaîne de T reliant y à la 2^e ligne de G .

$\rightarrow y$ n'est pas relié à n , impossible. Car y est relié à n par def de T

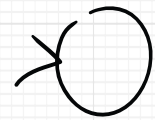
La chaîne qui relie y à r passe nécessairement par un noeud de S , ici c'est nécessairement s car y n'est relié qu'à un unique noeud de S . Donc x et y sont reliés entre eux via s et ne sont donc pas reliés à r .
 Or T relie x et y à r , donc il y a contradiction.

$$\text{donc } \Pi_3 \leq \Pi_4$$

Sachant que Set Cover est NP-Complet
 Π_3 $\rightarrow$ Π NP-Complet

donc Π_4 est NP-difficile

or $\Pi_4 \in NP$



certificat $O(l \log(m))$
 Trouver un ensemble
 "Trouver"
 T de l arêtes

Vérifieur $O(m+m)$

(probablement que des log devraient apparaître dans cette complexité)

Vérifier que T relie Y
 $\rightarrow$ créer le graphe restreint où T
 $\rightarrow$ faire un parcours du graphe
 pour vérifier que c'est connexe
 et que $Y \subseteq T$

$$\Pi_5 = 3\text{-SAT}$$

$$\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_m$$

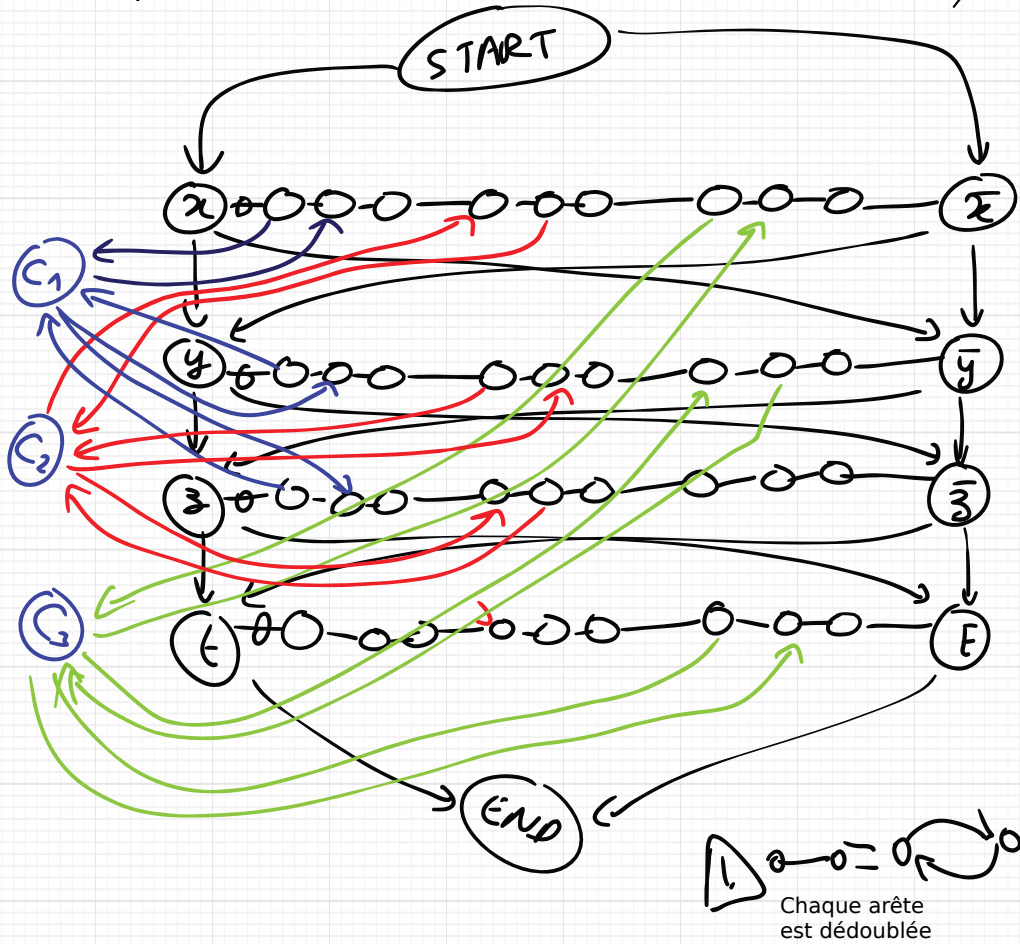
$$C_i = l_1^i \vee l_2^i \vee l_3^i$$

$$l_j^i \in \{ \bar{x}_1, x_1, \bar{x}_2, x_2, \dots, \bar{x}_m, x_m \}$$

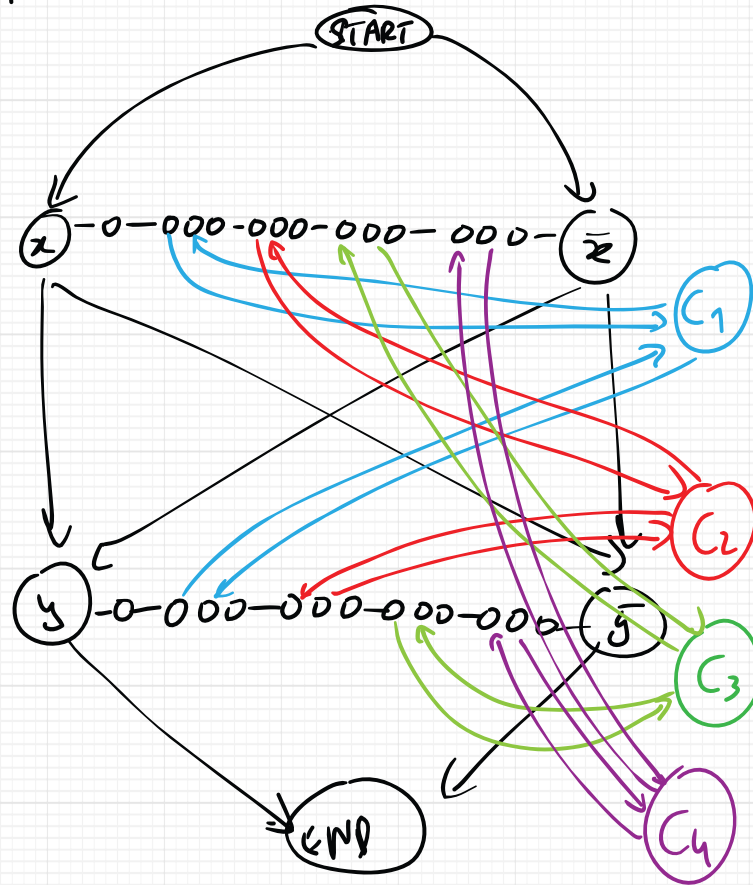
φ satisfiable ? "

Π_6 : Chemin hamiltonien

"Soit G un graphe orienté, et u et v 2 nœuds de G , est-ce qu'il existe un chemin de u à v qui passe une unique fois par chaque nœud ? "



$$q_n = (x \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$$



Description formelle de R :

- m : nombre de clauses C_j
- n : nombre de variables x_i
- créer deux noeuds START et END
- pour chaque variable x_i , créer deux noeuds x_i et $\overline{x_i}$ et les relier avec un chemin bi-orienté de taille $3m + 1$ noeuds, (bi-orienté = dans les 2 sens).
- On nomme dans l'ordre x_i^1 jusqu'à x_i^{3m+1} les noeuds de ce chemin
- Pour chaque clause C_j , créer un noeud C_j
- si le littéral x_i est dans la j-ième clause C_j alors relier x_i^{3j+1} à C_j et C_j à x_i^{3j+2}
- si le littéral $\overline{x_i}$ est dans la j-ième clause C_j alors relier x_i^{3j+2} à C_j et C_j à x_i^{3j+1}
- relier les noeuds x_i et $\overline{x_i}$ aux noeuds x_{i+1} et $\overline{x_{i+1}}$
- relier START à x_1 et $\overline{x_1}$
- relier x_n et $\overline{x_n}$ à END

R se fait en temps polynomial car on crée un graphe de taille polynomiale avec $O(3m * n)$ noeuds. Il ne peut donc y avoir plus de $O((3m * n)^2)$ arêtes. L'instance étant uniquement constituée de G, R s'arrête bien en temps polynomial.

Pourquoi si la formule Phi est satisfiable alors G est hamiltonien?

Si Phi est satisfiable, il existe une affectation des variables qui rend Phi vraie.
Soit $a_1, a_2, \dots, a_n$ cette affectation. a_i est la valeur VRAI ou FAUX de la variable x_i .

On peut alors parcourir G ainsi :

- on part de START
- pour tout i de 1 à n :
 - == Si a_i est VRAI, aller en x_i et parcourir le chemin de gauche à droite, sinon aller en $\overline{x_i}$ et le parcourir de droite à gauche
 - == Si a_i est VRAI et que x_i est le premier littéral VRAI de C_j , aller de x_i^{3j+1} à C_j et C_j à x_i^{3j+2}
 - == Si a_i est FAUX et que $\overline{x_i}$ est le premier littéral VRAI de C_j , aller de x_i^{3j+2} à C_j et C_j à x_i^{3j+1}
- aller en END.

Puisque Phi est satisfiable, pour chaque clause C_j , il existe un des littéraux de C_j qui est vrai, donc on visite bien tous les noeuds C_j .
On visite également naturellement tous les noeuds x_i^k , START et END, donc le chemin est hamiltonien.

Si maintenant G est un graphe hamiltonien, montrons que Phi est satisfiable.

L'idée de la preuve est la suivante

- PROPRIÉTÉ 1 : On montre d'abord que, dans un chemin hamiltonien, pour tout i , tous les arcs qui relient x_i^{3j+2} à x_i^{3j+3} et x_i^{3j+3} à x_i^{3j+4} sont tous dans le même sens (soit vers la gauche ou tous vers la droite).
- On déduit de la propriété 1 que dans un chemin hamiltonien, pour tout i , soit le chemin contient (x_i, x_i^1) et $(\overline{x_i}^{3m+1}, x_i)$ soit (x_i^1, x_i) et (x_i, x_i^{3m+1})
- On construit l'affectation suivante : a_i est VRAI dans le premier cas et FAUX dans le second cas.
- PROPRIÉTÉ 2 : On peut ensuite se servir de la propriété 1 pour montrer que si on arrive au noeud C_j depuis le chemin reliant x_i et $\overline{x_i}$ alors on est obligé d'y retourner (on ne peut pas visiter la moitié d'un chemin de x_i à $\overline{x_i}$ puis aller sur un chemin de x_j à $\overline{x_j}$ en passant par un noeud C_i)
- Ceci permet de démontrer que
 - == Si x_i est un littéral de C_j , alors C_j n'est visité depuis le noeud x_i^{3j+1} que si a_i est VRAI, sinon on viendrait de la droite et on ne pourrait pas revenir (propriété 2 en défaut)
 - == Si $\overline{x_i}$ est un littéral de C_j , alors C_j n'est visité depuis le noeud x_i^{3j+2} que si a_i est FAUX, sinon on viendrait de la droite et on ne pourrait pas revenir (propriété 2 en défaut)
- Dans les deux cas, si C_j est visité, C_j est vrai
- Puisque le chemin est hamiltonien, chaque clause est visitée, donc chaque clause est vraie, donc Phi est satisfiable.

$\Pi_2 =$ "Soit p un polynome,
 x un mot de $\{0,1\}$ et

M une machine de Turing non
 déterministe, est-ce que
 $M(x)$ peut se terminer en
 temps $P(|x|)$?"

$\hookrightarrow \Pi_2 \in NP$, il suffit comme certificat
 d'avoir les choix de M et de voir
 si, avec ces choix, $\tau(n, x, c) \leq P(|x|)$

$\Pi_3 = 3 SAT$

$\rightarrow \Pi_3$ est NP difficile.

Mq, si $\Pi \in NP$, alors $\Pi \leq \Pi_3$

$\hookrightarrow$ Si $\Pi \in NP$, $\exists$ P et N_Π ^{polynôme} N_Π résout
 Π en temps $O(P(|I|))$ (où I est une instance
 de Π)

J'insiste sur les indices " P_i " pour p et M .

Chaque problème de NP peut être résolu par une machine
 en temps polynomial (définition de NP)

Cette machine et ce polynome (celui qui limite le temps
 polynomial) sont spécifique au problème P_i . Deux problèmes
 sont résolus par 2 machines distinctes avec des polynomes
 différents.

Le "Il existe" ci-dessus signifie que, pour prouver que P_i est
 dans NP, on a été capable de trouver M et p . Donc c'est plus
 qu'un "Il existe" mathématique, c'est vraiment "quelqu'un
 nous a donné une machine et un polynome associé à P_i , que
 je peux utiliser dans un algorithme, comme par exemple
 une réduction polynomiale".

Dans ce slide, on montre que P_i est NP-Complet (et la suivante) en utilisant la définition originale de NP-Complet :
 - si pour tout P_i de NP, $P_i \leq P_{i_7}$ et si P_{i_7} est dans NP alors P_{i_7} est NP-Complet.

Soit I une instance de Π

R : remplace I , P_π et N_π $\rightarrow$ modifiée en supprimant les états terminaux de refus.

On doit supprimer ces états pour créer une dissymétrie entre les réponses OUI et NON :
OUI, la machine s'arrête en temps $p(x)$
NON, la machine ne s'arrête pas.

$\Rightarrow$ Si $I \in \mathcal{L}_\pi^Y$ alors N_π s'arrête par définition en temps $O(P_\pi(I))$
donc $(I, P_\pi, N_\pi) \in \mathcal{L}_\pi^Y$

$\Leftarrow$ Sinon N_π ne s'arrête pas (plus d'état de refus) $\rightarrow (I, P_\pi, N_\pi) \in \mathcal{L}_\pi^N$

donc $\Pi \leq \Pi_\pi$

$$M_q \quad \Pi_7 \leq \Pi$$

$$\bigvee_{t=0}^{P(x)} \bigvee_{i=-P(x)}^{P(x)} \left(R_A^t \wedge (H_i^t \wedge T_{iB}^t) \right) \Rightarrow H_{i+1}^{t+1} \wedge R_C^t$$

Détail, ici on a pas dit que chaque case de la bande devait conserver le même symbole entre les instants t et $t+1$.

- 1) Utiliser des variables pour décrire la configuration de la machine M à l'instant t

Exemple $T_{i,1}^t ; T_{i,0}^t ; T_{i,B}^t$

$\hookrightarrow$ $\begin{array}{|c|} \hline i \\ \hline 11 \\ \hline \end{array} \quad \begin{array}{|c|} \hline i \\ \hline 10 \\ \hline \end{array} \quad \begin{array}{|c|} \hline i \\ \hline 11 \\ \hline \end{array}$

H_i^t

$\begin{array}{|c|} \hline i \\ \hline 1 \square \\ \hline \end{array}$

R_s^t
Registre de l'état sur l'état s

- 2) S'assurer que le nb de variables est polynomial

- 3) Relier les configurations à l'instant t et à l'instant $t+1$



Pour chaque transition, il faut écrire une formule

- 4) Instant 0

Un détail important est le point 2.

Le nombre de variables est polynomial car la machine doit s'arrêter avant l'instant $p(x)$.

Donc

- $t \leq p(x)$

- la tête de lecture ne peut écrire plus loin que la case $p(x)$ et la case $-p(x)$

Donc le nombre de variables est de l'ordre de $O(p(x)^2 |S|)$ où S est l'ensemble des états du graphe de la machine