Introduction aux classes de complexité classique

HUIN Nicolas

COATI (INRIA/I3S)

10 mars 2015

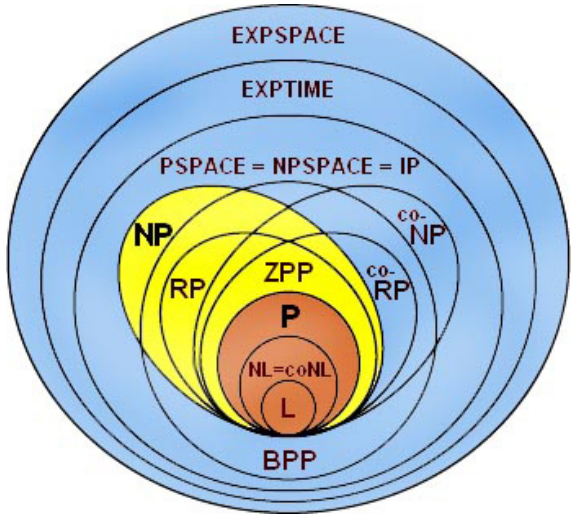
1. La théorie de la complexité

2. Hiérarchie des classes
   - Problème de décision
     - P & NP
     - Complexité paramétrée
   - Problème d'optimisation (NPO & APX)

Qu'est ce que la théorie de la complexité ?

Classer les différents problèmes de calcul selon

- le type de problème,
- les ressources utilisées,
- et le modèle de calcul.



Types de problème

Plusieurs types de problèmes existent tels que

- Problème de décision (Oui/Non)
  - Est-ce qu'un nombre est premier ?
  - Est-ce qu'un graphe est connexe ?
- Problème d'optimisation (Maximisation et Minimisation)
- Problème de fonctions

Types de problème

Plusieurs types de problèmes existent tels que

- Problème de décision (Oui/Non)
- Problème d'optimisation (Maximisation et Minimisation)
  - Le flot maximum d'un graphe.
  - La coupe minimum d'un graphe.
- Problème de fonctions

Types de problème

Plusieurs types de problèmes existent tels que

- Problème de décision (Oui/Non)
- Problème d'optimisation (Maximisation et Minimisation)
- Problème de fonctions
  - Fonction d'Ackermann.
  - Calcul d'un polynôme.

Ressources

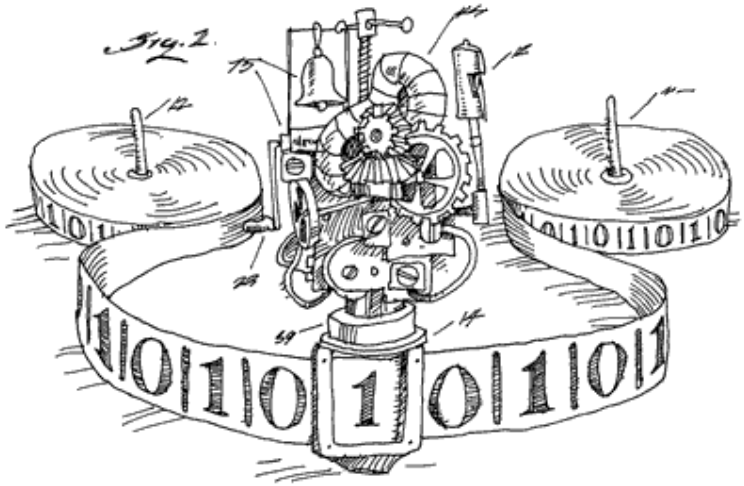
Plusieurs types de ressources sont prises en compte dans le calcul de la complexité des algorithmes tels que

- le **temps**
- la **mémoire**
- le nombre de processeurs, le nombre de portes dans un circuit, etc...

Modèle de calcul

Définition (Machine de Turing)

Une machine de Turing est une machine abstraite qui manipule les symboles inscrits sur un ruban de taille infinie selon un ensemble de règles.



Modèle de calcul

Définition (Machine de Turing)

Une machine de Turing est une machine abstraite qui manipule les symboles inscrits sur un ruban de taille infinie selon un ensemble de règles.

Il existe plusieurs variantes de la machine de Turing :

- non déterministe
- probabiliste
- à plusieurs rubans, etc...

2. Hiérarchie des classes
   - Problème de décision
     - P & NP
     - Complexité paramétrée
   - Problème d'optimisation (NPO & APX)

Continuum Espace-Temps

$DTIME(f(n))$ & $DSPACE(f(n))$

Classes contenant les problèmes de décision utilisant $O(f(n))$ ressources sur une machine de Turing déterministe.

$NTIME(f(n))$ & $NSPACE(f(n))$

Classes contenant les problèmes de décision utilisant $O(f(n))$ ressources sur une machine de Turing non déterministe.

Polynomial Time (P)

Définition

Ensemble de problèmes de décision pouvant être résolus en temps polynomial sur une machine de Turing déterministe.

$$P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$$

Si un problème est dans P, on peut le résoudre "facilement".

Exemple

Savoir si un nombre est premier est dans P.

Non Deterministe Polynomial Time (NP)

Définition

Ensemble de problèmes de décision pouvant être résolus en temps polynomial sur une machine de Turing non déterministe.

$$NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$$

On peut vérifier une solution du problème en temps polynomial.

Exemple

Voyageur de commerce (TSP), Couverture de taille k d'un graphe

Réduction

Définition (Réduction de Karp)

Soient Π et Π', deux problèmes d'une classe C. Une réduction f consiste alors à transformer toutes instances i de Π en une instance $f(i)$ de Π' tel que $i \in \Pi \Leftrightarrow f(i) \in \Pi'$.

Une réduction est un pré-ordre : réflexive et transitive.

Le type de la réduction dépend de la complexité de la fonction de transformation.

Problème difficile et complet

Définition (C-Difficile)

Soit C une classe de complexité, un problème Π est **C-difficile** si tout problème de C peut être réduit à Π.

Définition (C-Complet)

Un problème C-difficile appartenant à C est **C-Complet**.

Exemple

NP et P sont fermées par réduction en temps polynomial.

Réduction est un pré-ordre $\rightarrow$ Réduction à un seul problème complet.

3 Color & 3 SAT

Définition (SAT)

Soit une formule booléenne, existe-t-il une affectation des variables de telle sorte que la formule soit vraie ?

L'un des premiers problèmes montré NP-Complet [Cook & Levin, 1971]

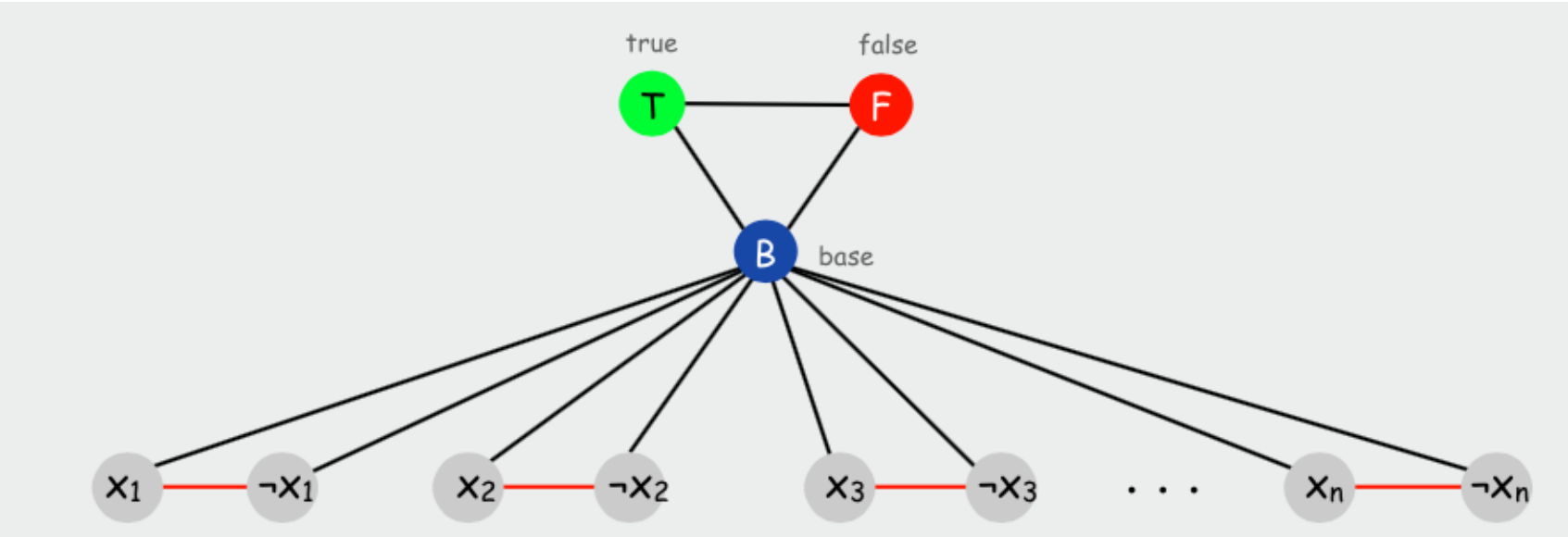
3-SAT est une variante où les clauses conjonctives sont de taille 3

Définition (3-COLOR)

Soit un graphe $G = (V, E)$, une 3-coloration de G assigne une couleur $c(v) \in \{c_1, c_2, c_3\}, \forall v \in V$ tel que $\forall (u, v) \in E, c(u) \neq c(v)$

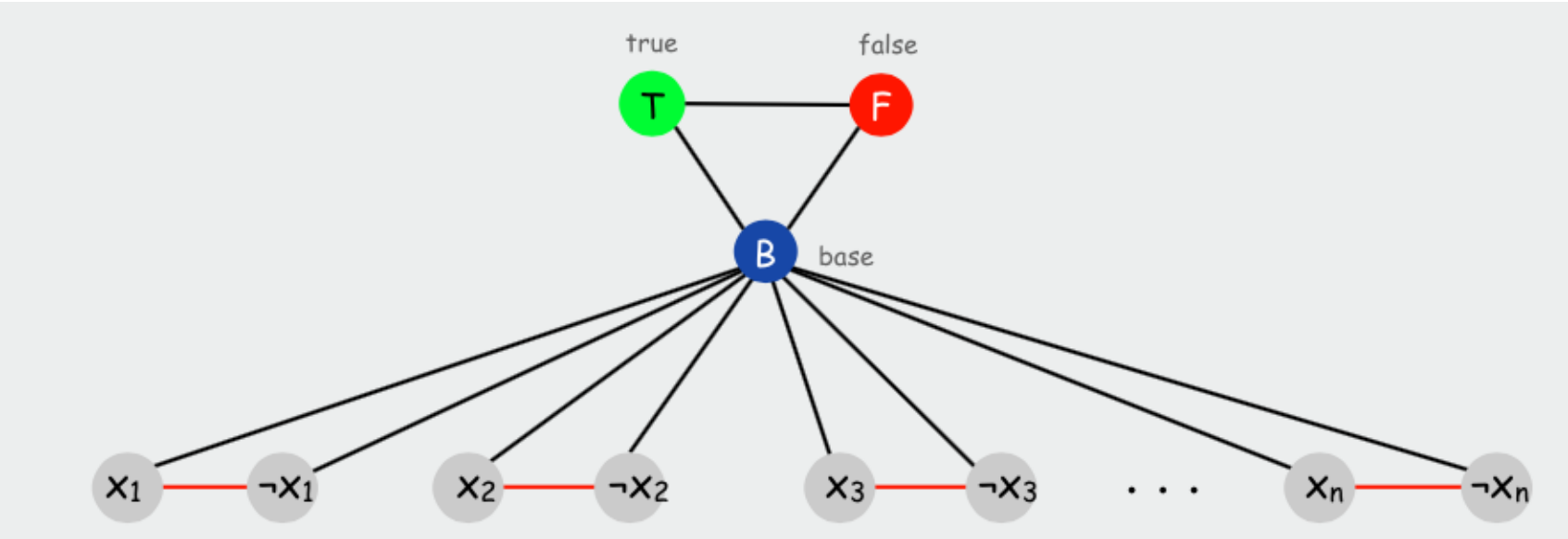
NP-Complet (en réduisant 3-SAT à 3-Color)

3 Color & 3 SAT



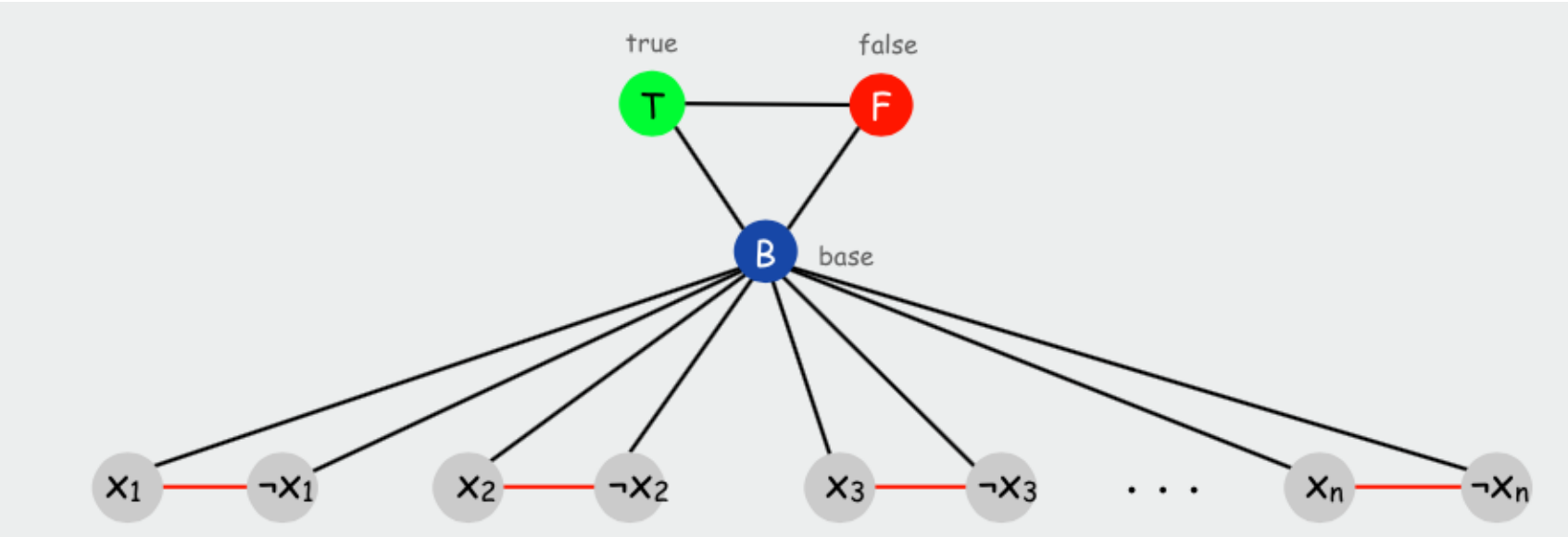
- Un sommet pour un littéral et son complément.
- Un triangle pour chaque couleur.
- Un littéral et son opposé ne peuvent être de la même couleur.

3 Color & 3 SAT



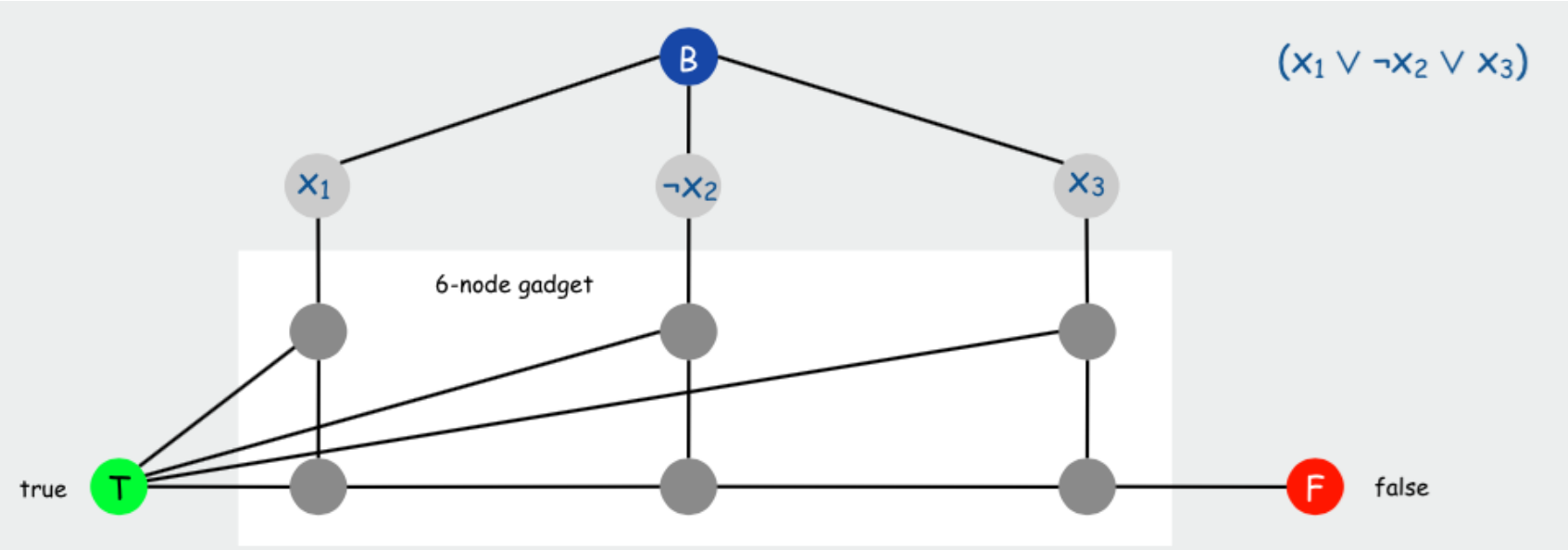
- Un sommet pour un littéral et son complément.
- Un triangle pour chaque couleur.
- Un littéral et son opposé ne peuvent être de la même couleur.

3 Color & 3 SAT

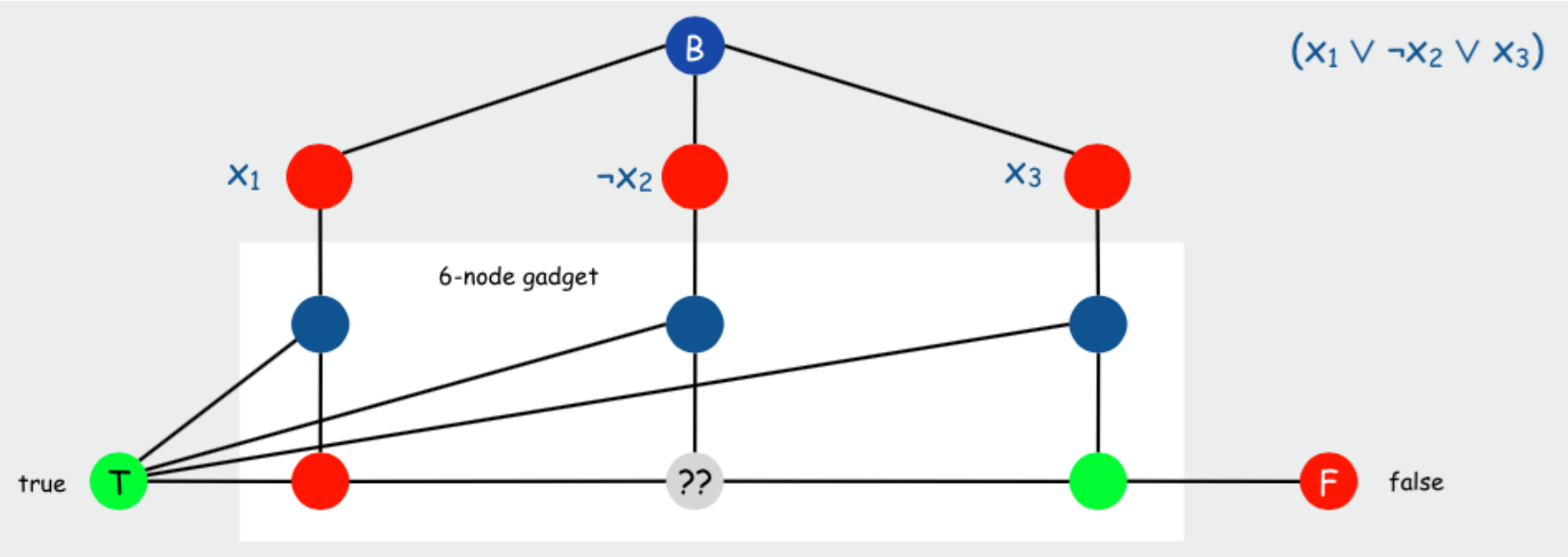


- Un sommet pour un littéral et son complément.
- Un triangle pour chaque couleur.
- Un littéral et son opposé ne peuvent être de la même couleur.

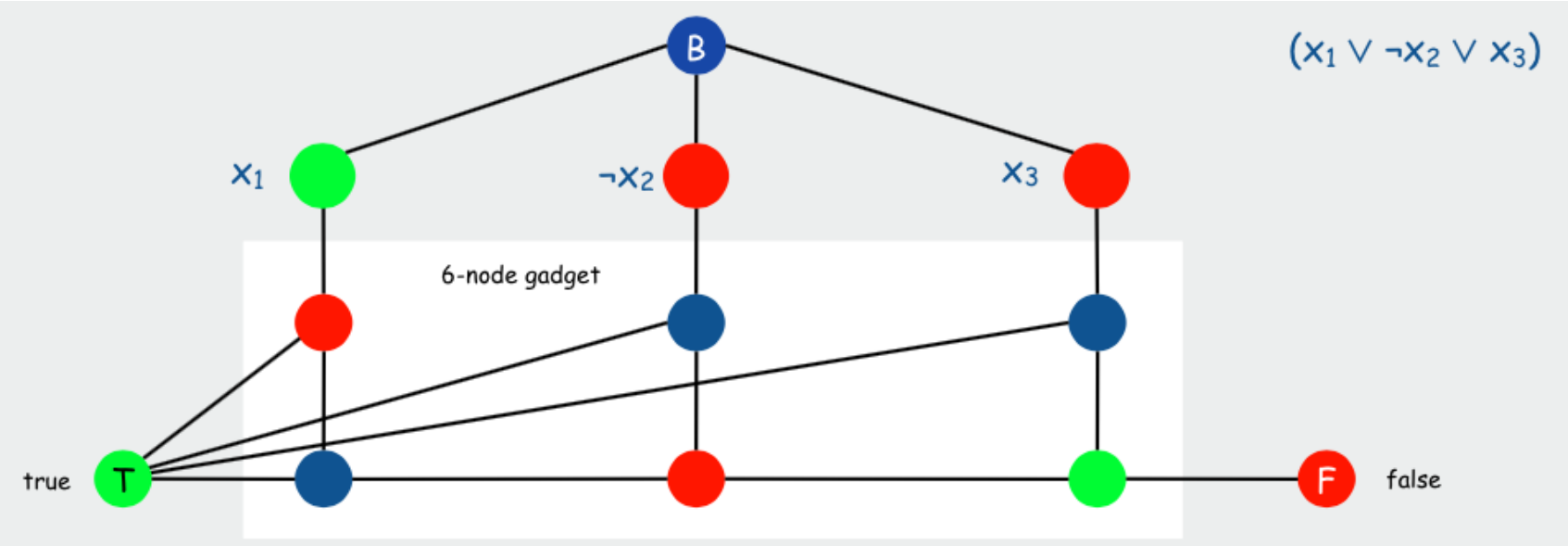
Une clause



Tout le monde est faux.



Un littéral est vrai.



Temps pseudo-polynomial

Définition

Polynomial par rapport à la valeur numérique (unaire) de l'entrée mais exponentiel par rapport à sa taille (binaire).

Pour tester la primalité d'un nombre x, on peut naïvement vérifier si $\exists i \in \{1, ..., \sqrt{n}\}$ qui divise n.

Exemple

Pour $x \approx 10\ 000\ 000\ 000$, il faut 100 000 divisions. $|x| \approx 33 bits$
Pour $x \approx 1\ 000\ 000\ 000\ 000$, il faut 1 000 000 divisions. $|x| \approx 39 bits$

Fortement & faiblement NP

Définition

Un problème NP-Difficile (NP-Complet) pouvant être résolu en temps pseudo-polynomial est un problème **faiblement** NP-Difficile (NP-Complet).

Ce n'est pas le cas des problèmes **fortement** NP-Difficiles (NP-Complets).

Exemple

Le problème du sac à dos est **faiblement** NP-Complet.
Le problème de la mise en boîte est **fortement** NP-Complet.

NP-Intermédiaire

Définition

Classe des problèmes dans NP qui sont ni dans P, ni dans NP-Complet. Elle est hypothétique.

Exemple

L'isomorphisme de graphe, et la factorisation d'entier.

Si NP-Intermédiaire est vide, alors P = NP.

2. Hiérarchie des classes
   - Problème de décision
     - P & NP
     - Complexité paramétrée
   - Problème d'optimisation (NPO & APX)

Fixed-parameter tractable

Définition

Un problème FPT en paramètre k un est problème de décision pouvant être résolus en temps $f(k)n^{O(1)}$ où le paramètre k ne dépend pas de la taille du problème.

Exemple

Couverture de taille k

Nucléarisation

Définition (Nucléarisation)

Soit une instance x avec un paramètre k d'un problème Π.
y est un noyau de x ssi :

$(x, k) \in \Pi \Leftrightarrow (y, k') \in \Pi$ et,

la taille de y est bornée par une fonction sur k

$k' < k$

En temps polynomial.

On cherche à réduire la taille de l'instance

Un problème admet une nucléarisation et décidable $\Longleftrightarrow$ Un problème est FPT

Couverture

Définition (Couverture)

Soit un graphe $G = (V, E)$. Une couverture est un ensemble de sommet S tel que $\forall (u, v) \in E, u \in S \vee v \in S$.

Le problème de décision cherche l'existence d'un couverture de taille k.

Un algorithme brute force nous donne la solution en $O(2^k m^{O(1)})$

Nucléarisation du vertex cover

Une instance d'une couverture de taille k peut être réduite par ces 3 règles :

- Si un sommet est isolé, on peut le supprimer du graphe
- Si un sommet u est de degré $> k$, u fait partie de la solution. On cherche une couverture de taille $k-1$ sur le graphe privé de u.
- Si un sommet u a un unique voisin v, alors il existe une couverture minimum comprenant v et pas u.

Nucléarisation du vertex cover

Une instance d'une couverture de taille k peut être réduite par ces 3 règles :

- Si un sommet est isolé, on peut le supprimer du graphe
- Si un sommet u est de degré $> k$, u fait partie de la solution. On cherche une couverture de taille $k - 1$ sur le graphe privé de u.
- Si un sommet u a un unique voisin v, alors il existe une couverture minimum comprenant v et pas u.

Nucléarisation du vertex cover

Une instance d'une couverture de taille k peut être réduite par ces 3 règles :

- Si un sommet est isolé, on peut le supprimer du graphe
- Si un sommet u est de degré $> k$, u fait partie de la solution. On cherche une couverture de taille $k - 1$ sur le graphe privé de u.
- Si un sommet u a un unique voisin v, alors il existe une couverture minimum comprenant v et pas u.

Nucléarisation du vertex cover

Une instance d'une couverture de taille k peut être réduite par ces 3 règles :

- Si un sommet est isolé, on peut le supprimer du graphe
- Si un sommet u est de degré $> k$, u fait partie de la solution. On cherche une couverture de taille $k-1$ sur le graphe privé de u.
- Si un sommet u a un unique voisin v, alors il existe une couverture minimum comprenant v et pas u.

Si le graphe contient plus de k^2 arêtes, on ne peut pas avoir de couverture de taille k. Tous les sommets sont de degré au plus k, donc on peut couvrir au plus k^2 arêtes.

2 Hiérarchie des classes

- Problème de décision
- Problème d'optimisation (NPO & APX)

NPO

Définition (NPO)

Classe contenant les problèmes d'optimisation dont la formulation en problème de décision se trouve dans NP.

Exemple

TSP

APX

Définition (APX)

Classe de tous les problèmes NPO tel qu'il existe une r-approximation en temps polynomial pour le résoudre, c'est à dire un algorithme polynomial déterministe qui renvoie au moins OPT/r.

Exemple

Couverture minimum, Coupe maximum

$TSP \notin APX$

Définition (Cycle hamiltonien)

Soit $G = (V, E)$, un cycle hamiltonien est un cycle qui passe par tous les sommets une et une seule fois. NP-Complet.

Définition (TSP)

Soit un graphe pondéré $G = (V, E, W)$. Le problème du voyageur de commerce cherche un cycle de taille n de poids minimum.

Pour prouver que TSP n'appartient pas à APX, il suffit d'y réduire le problème du cycle hamiltonien.

$TSP \notin APX$

Définition (Cycle hamiltonien)

Soit $G = (V, E)$, un cycle hamiltonien est un cycle qui passe par tous les sommets une et une seule fois. NP-Complet.

Définition (TSP)

Soit un graphe pondéré $G = (V, E, W)$. Le problème du voyageur de commerce cherche un cycle de taille n de poids minimum.

Pour prouver que TSP n'appartient pas à APX, il suffit d'y réduire le problème du cycle hamiltonien.

$TSP \notin APX$

Définition (Cycle hamiltonien)

Soit $G = (V, E)$, un cycle hamiltonien est un cycle qui passe par tous les sommets une et une seule fois. NP-Complet.

Définition (TSP)

Soit un graphe pondéré $G = (V, E, W)$. Le problème du voyageur de commerce cherche un cycle de taille n de poids minimum.

Pour prouver que TSP n'appartient pas à APX, il suffit d'y réduire le problème du cycle hamiltonien.

$TSP \notin APX$ (Preuve)

On admet que le TSP peut être résolu par une r-approximation A.

- Pour toute instance $G = (V, E)$ du problème du cycle hamiltonien, on construit un graphe complet pondéré $G' = (V, E', W)$.

$$\forall (u, v) \in E', W_{uv} = \begin{cases} 1 \text{ si } (u, v) \in E \\ 1 + nr \text{ sinon} \end{cases}$$

- S'il existe un cycle hamiltonien, A renvoie un cycle de taille **n** puisque la plus petite étape d'approximation supérieur est d'au moins de taille $n - 1 + (1 + nr) = n(1 + r)$, supérieur au ratio r.
- Sinon, il renvoie un cycle de taille $n(1 + r)$.

Impossible (si $P \neq NP$)

Le problème du cycle hamiltonien serait alors résoluble en temps polynomial.

$TSP \notin APX$ (Preuve)

On admet que le TSP peut être résolu par une r-approximation A.

- Pour toute instance $G = (V, E)$ du problème du cycle hamiltonien, on construit un graphe complet pondéré $G' = (V, E', W)$.

$$\forall (u, v) \in E', W_{uv} = \begin{cases} 1 \text{ si } (u, v) \in E \\ 1 + nr \text{ sinon} \end{cases}$$

- S'il existe un cycle hamiltonien, A renvoie un cycle de taille **n** puisque la plus petite étape d'approximation supérieur est d'au moins de taille $n - 1 + (1 + nr) = n(1 + r)$, supérieur au ratio r.
- Sinon, il renvoie un cycle de taille $n(1 + r)$.

Impossible (si $P \neq NP$)

Le problème du cycle hamiltonien serait alors résoluble en temps polynomial.

$TSP \notin APX$ (Preuve)

On admet que le TSP peut être résolu par une r-approximation A.

- Pour toute instance $G = (V, E)$ du problème du cycle hamiltonien, on construit un graphe complet pondéré $G' = (V, E', W)$.

$$\forall (u, v) \in E', W_{uv} = \begin{cases} 1 \text{ si } (u, v) \in E \\ 1 + nr \text{ sinon} \end{cases}$$

- S'il existe un cycle hamiltonien, A renvoie un cycle de taille **n** puisque la plus petite étape d'approximation supérieur est d'au moins de taille $n - 1 + (1 + nr) = n(1 + r)$, supérieur au ratio r.
- Sinon, il renvoie un cycle de taille $n(1 + r)$.

Impossible (si $P \neq NP$)

Le problème du cycle hamiltonien serait alors résoluble en temps polynomial.

$TSP \notin APX$ (Preuve)

On admet que le TSP peut être résolu par une r-approximation A.

- Pour toute instance $G = (V, E)$ du problème du cycle hamiltonien, on construit un graphe complet pondéré $G' = (V, E', W)$.

$$\forall (u, v) \in E', W_{uv} = \begin{cases} 1 \text{ si } (u, v) \in E \\ 1 + nr \text{ sinon} \end{cases}$$

- S'il existe un cycle hamiltonien, A renvoie un cycle de taille **n** puisque la plus petite étape d'approximation supérieur est d'au moins de taille $n - 1 + (1 + nr) = n(1 + r)$, supérieur au ratio r.
- Sinon, il renvoie un cycle de taille $n(1 + r)$.

Impossible (si $P \neq NP$)

Le problème du cycle hamiltonien serait alors résoluble en temps polynomial.

PTAS & FPTAS

Définition (PTAS)

Un PTAS est un algorithme qui pour une instance x et un paramètre $\epsilon > 0$ renvoie une solution en temps polynomial qui est $> (1 - \epsilon)OPT$.
Complexité en $O(n^{f(1/\epsilon)})$.

Exemple

TSP Euclidien

Définition (FPTAS)

Plus restrictif que PTAS. Complexité polynomiale en n et $1/\epsilon$

Exemple

Problème du sac à dos.

PTAS & FPTAS

Définition (PTAS)

Un PTAS est un algorithme qui pour une instance x et un paramètre $\epsilon > 0$ renvoie une solution en temps polynomial qui est $> (1 - \epsilon)OPT$.
Complexité en $O(n^{f(1/\epsilon)})$.

Exemple

TSP Euclidien

Définition (FPTAS)

Plus restrictif que PTAS. Complexité polynomiale en n et $1/\epsilon$

Exemple

Problème du sac à dos.

D'autres classes

- PSPACE. P dans l'espace (Frédéric Giroire)
- PCP. Probabilistically checkable proof. (Nicolas Nisse)
- Classe de comptage. (Stéphane Pérennes)
- IP. Preuve interactive. (Guillaume Ducoffe)

Merci !