

## Cours

- Problème de décision, d'optimisation, de maximisation/minimisation. Comment transformer un problème d'optimisation en un problème de décision.
- Taille d'une instance.
- Définition d'une classe de complexité. Définition de P, NP, NP-Dur, NP-Complet
- Réduction polynomiale
- $k$ -SAT, SAT, certificat

## Question de Cours

- Comment transformer un problème de maximisation en un problème de décision ?
- Montrer que 3-SAT est NP-Complet

## Max 2-SAT

On considère le problème d'optimisation suivant:

**Entrée:**  $\varphi$  une formule en FNC

**Sortie:** Une valuation  $\varphi$  qui maximise le nombre de clauses satisfaites

1. Donner la version problème de décision
2. Montrer que si  $k$  est le nombre de clauses le problème devient polynomial
3. Montrer que la version décisionnel du problème est NP-Complet

## SUBSET-SUM

On considère le problème SUBSET-SUM suivant:

- **Entrée**  $n \in \mathbb{N}$  et  $a_1, a_2, \dots, a_n, S \in \mathbb{N}$
  - **Sortie:** Est-ce qu'il existe  $I \subseteq \llbracket 1; n \rrbracket$  tel que  $\sum_{i \in I} a_i = S$
1. Pour les suites suivante de  $(a_n)_n$  et les valeurs de  $S$  suivantes, indiquer si le problème est satisfiable ou non:
    - $(a_n)_n = (1, 2, 4, 8, 16), S = 7$
    - $(a_n)_n = (31, 24, 2, 43, 12, 12, 18), S = 29$
    - $(a_n)_n = (1001, 1010, 101, 100, 11), S = 1111$
  2. Montrer que le problème SOMME est dans la classe NP
  3. Donner un algorithme qui en  $O(nS)$  résoud le problème

On cherche à prouver que le problème précédent est NP-Complet. Pour cela on considère une généralisation du problème sur des  $k$ -uplets intitulé SUBSET-SUM-VECT:

- **Entrée**  $k, n \in \mathbb{N}$  et  $a_1, a_2, \dots, a_n, S \in \mathbb{N}^k$
  - **Sortie:** Est-ce qu'il existe  $I \subseteq \llbracket 1; n \rrbracket$  tel que  $\sum_{i \in I} a_i = S$
4. Montrer que on peut réduire le problème **SOMME-VECT** au problème **SOMME**
  5. Montrer que **SOMME-VECT** est NP-Complet à partir de 3-SAT.
  6. Est-ce que avec la question 3 on a démontré que  $P=NP$  ? Justifier.

## INDEPENDANT-SET et CLIQUE

On considère les deux problèmes CLIQUE-MAX et INDEPENDANT-SET suivant:

CLIQUE:

- **Entrée:** Un graphe  $G = (V, E)$  non orienté
- **Sortie:** Le plus grand  $S \subseteq V$  en cardinal tel que  $V^2 \subseteq E$

INDEPENDANT-SET:

- **Entrée:** Un graphe  $G = (V, E)$  non orienté et un  $k \in \mathbb{N}$
  - **Sortie:** Est-ce qu'il existe  $S \subseteq V$  tel que  $S^2 \cap E = \emptyset$  avec  $|S| \geq k$  ?
1. Donner la version problème de décision de CLIQUE. Montrer que elle est NP.
  2. Donner une réduction polynomiale de INDEPENDANT-SET dans la version problème de décision de CLIQUE.
  3. En réduisant depuis 3-SAT, et en considérant un graphe avec  $3k$  noeuds avec  $k$  le nombre de clauses, montrer que CLIQUE est NP-Complet.

## MIN-COLOR-PATH

On considère le problème MIN-COLOR-PATH suivant:

- **Entrée:**  $G = (V, E)$  un graphe,  $s, t \in V$  deux sommets et  $c : V \rightarrow \mathbb{N}$  une coloration
- **Sortie:** Un chemin  $s = x_1, \dots, x_n$  de  $s$  à  $t$  qui minimise  $\text{Card}(\{c(x_1), \dots, c(x_n)\})$

1. Donner la version problème de décision, et montrer qu'elle est NP.
2. Donner un algorithme pour résoudre le problème si  $G$  est un cycle.
3. Montrer que le problème de décision associé à MIN-COLOR-PATH est NP-Complet.

On suppose maintenant que l'on rajoute la condition  $|\text{Im}(c)| \leq k$  avec  $k \in \mathbb{N}$  fixé.

4. Montrer maintenant que le problème est P.

On considère l'algorithme qui prend des couleurs uniformément au hasard dans  $\text{Im}(c)$  jusqu'à ce qu'il existe un chemin de  $s$  à  $t$

5. Quel est ce type d'algorithme ? Proposer des exemples arbitrairement grand ou cet algorithme renvoie une solution d'espérance en  $O(n)$  au lieu de  $O(1)$ .

## DOMINATING-SET

Soit  $G = (V, E)$ , on dit que  $S \subseteq V$  est un *ensemble dominant* si tous les sommets de  $V$  sont soit dans  $S$ , soit voisin d'un sommet de  $S$ . On note  $\gamma(G)$  le cardinal minimum d'un ensemble dominant de  $G$ .

On considère le problème DOMINATING-SET suivant:

- **Entrée:**  $G = (V, E)$  un graphe
- **Sortie:**  $\gamma(G)$

1. Donner la version problème de décision de DOMINATING-SET. Montrer qu'elle est NP.
2. Montrer que  $\gamma(G) \leq |V|/2$
3. Montrer que la version problème de décision de DOMINATING-SET est NP-Complet.
4. Montrer que si l'on restreint l'entrée du problème à des arbres, alors le problème est dans P.
5. Pour  $G$  un graphe, on note  $\Delta_G$  son degré maximal. Donner une  $\Delta_G$ -approximation de DOMINATING-SET.
6. Donner similairement une  $\log(|V|)$ -approximation. Est-ce qu'une des approximations fait les deux ?

# Langages NP-Complets sur $\Sigma = \{a\}$ <sup>1</sup>

Un langage  $L$  est dit unaire si c'est un langage sur un alphabet à une lettre, donc si  $L \subseteq \{a\}^*$ .  
 $L$  est dit *NP-Difficile* si le problème de tester si  $x \in L$  est NP-Difficile. Dans ce problème, la taille de l'entrée est  $|x|$ .

1. On note  $\text{SAT}^+$  l'ensemble des entrées positives au problème SAT et S. Montrer que  $L$  est NP-Difficile ssi il existe une fonction  $f$  de complexité polynomiale telle que

$$\forall \varphi, \varphi \in \text{SAT}^+ \iff f(\varphi) \in L$$

On cherche à montrer le théorème suivant

**Théorème de Beraman:** S'il existe un langage unaire NP-Dur alors  $P = NP$

Soit  $\varphi(x_1, x_2, \dots, x_n)$  une instance de SAT. On suppose que  $L$  est un langage unaire NP-Difficile.

2. On note  $\varphi_{\top}(x_2, \dots, x_n) := \varphi(\top, x_2, \dots, x_n)$  et  $\varphi_{\perp}(x_2, \dots, x_n) := \varphi(\perp, x_2, \dots, x_n)$ . Montrer que

$$\varphi \in \text{SAT}^+ \iff \varphi_{\top} \in \text{SAT}^+ \vee \varphi_{\perp} \in \text{SAT}^+$$

On note  $|\varphi|$  la taille d'une formule  $\varphi$ , et on suppose que  $\forall i, |\top|, |\perp| < |x_i|$

3. Soit  $\varphi$  une formule à  $n$  variables. Donner un algorithme qui, pour tout  $i \leq n$ , construit un ensemble de formules  $\Phi_i$  tel que toute formule de  $\Phi_i$  contient  $n - i$  variables et  $\varphi \in \text{SAT}^+ \iff \exists \varphi \in \Phi_i, \varphi \in \text{SAT}^+$
4. Montrer qu'il existe un polynôme  $p$  tel que  $\sum_{\varphi \in \Phi_i} |\varphi| \leq p(|\varphi|)$
5. En déduire que  $P = NP$ , et donc le théorème de Beraman.

---

<sup>1</sup>Mallory Marin, doc de colles

## Intersection d'automates<sup>2</sup>

On considère le problème NONEMPTY-REGULAR-INTERSECTION suivant:

**Entrée:**  $A_1, \dots, A_n$  une liste d'automate sur  $\Sigma = \{a, b\}$

**Sortie** Est-ce que

$$\bigcap_{i \leq n} L(A_i) \neq \emptyset$$

1. Montrer que NONEMPTY-REGULAR-INTERSECTION est NP
2. Donner une instance du problème tel que le plus petit mot dans l'intersection est de longueur exponentielle en la taille de l'entrée.
3. Montrer que si  $\Sigma = \{a\}$  alors le problème est polynomial.
4. Montrer que NONEMPTY-REGULAR-INTERSECTION est NP-Complet à partir de 3-SAT

---

<sup>2</sup>Algo 1 ENS Lyon TD 8

## Sheffer

On définit l'opérateur de sheffer  $X \uparrow Y := \neg(X \wedge Y)$  et l'opérateur  $X \oplus Y := (X \vee Y) \wedge (\neg X \vee \neg Y)$ . Indiquer si les problèmes suivant sont Np-Complet ou Polynomial:

- **Entrée:** Une formule  $\varphi$  avec que des  $\oplus$  **Sortie:** Si  $\varphi$  est satisfiable.
- **Entrée:** Une formule  $\varphi$  avec que des  $\wedge$  et  $\neg$  **Sortie:** Si  $\varphi$  est satisfiable.
- **Entrée:** Une formule  $\varphi$  avec que des  $\uparrow$  **Sortie:** Si  $\varphi$  est satisfiable.

Plus dur, essayer de trouver une réduction de SAT (pas 3-SAT) aux formules avec  $\uparrow$ .

