*Département Informatique : Deuxième Année*

---

**Décidabilité T.D. N° 3**

20 décembre 2010

**La complexité**

# 1 Exemple d'un problème de la classe P : Le cycle eulérien

Dans cet exercice nous nous intéressons au problème suivant : étant donné un graphe non orienté G, G possède-t-il un cycle eulérien ? Nous rappelons qu'un cycle est dit eulérien s ?il passe par toutes les arêtes du graphe non orienté :

- Démontrer le théorème d'Euler : un graphe non orienté admet un cycle eulérien si et seulement si : il est connexe et tous ses sommets sont de degrés pairs.
- Déduire que ce problème est dans la classe P.

**Corrigé** **La condition nécessaire** : En suivant le cycle eulérien de graphe : on passe par chaque sommet un nombre pair de fois car : chaque fois qu'on arrive à un sommet on repart via une autre arête ; chaque fois, par des pairs d'arêtes différentes, etc. Jusqu'à l'arrivée au point de départ.

**La condition suffisante** : Pour m=1 la propriété est vraie (un sommet et une boucle). Supposons la propriété est vraie pour un graphe ayant un nombre d'arêtes strictement inférieur à m.

Soit un graphe de degré $> m$. Le degré minimale est de 2 (car le graphe est connexe). Soit C un cycle de G. Soit E(C) L'ensemble d'arêtes de C. G-E(C) les composantes connexes du graphe restant. Soient $H_1, .., H_k$ les composantes de G-E(C). Chaque $H_i$ Etant un graphe eulérien (par hypothèse de récurrence) Soit $C_i$ le cycle eulérien de $H_i$, Il existe au moins un sommet commun $x_i$ entre C et $C_i$ (puisque le graphe G est connexe) On peut définir un cycle eulérien dans le graphe en intercalant $C_i$ dans C au sommet $x_i$ pour chaque i dans $1..k$.

**fonction** Euler(G) : cycle
  construire-Cycle(C,G)

**si** $E(G) \setminus E(C) <> \Phi$ **alors**
determiner-Composantes $(H_1, H_2, .., H_k)$
$i \leftarrow 1$
**tantque** $i <= k$ **faire**
$C_i \leftarrow Euler(H_i)$
**fin tantque**
intercaler $(C_1, C_2, .., C_k)$
**fin si**
**RETOURNER** C

Selon ce théorème , il suffit d'examiner le degré de chaque sommet pour décider si le graphe est eulérien : O(n). n étant le nombre de sommets du graphe.

# 2 Exemple d'un problème de la classe NP-complet : le problème SAT et ses variantes

## 2.1 Le problème SAT

Soit la forme conjonctive normale suivante : $F = (p \vee q \vee r) \wedge (\neg q \vee \neg p \vee t \vee s) \wedge (\neg t \vee \neg s)$.

1. Proposer un algorithme exhaustif qui répondra à la question suivante : est-ce que F est satisfiable ? Quel est le coût de cet algorithme ?
2. Proposer un algorithme non déterministe répondant à la même question. Quel est le coût de cet algorithme ?

**Corrigé** L'algorithme exhaustif génère toutes les interprétations et vérifie la valeur de vérité de chaque interprétation : $O(2^n)$ avec n=5.

L'algorithme non déterministe génère une interprétation et calcule sa valeur de vérité : O(n) avec n=5.

## 2.2 Deux variante du problème SAT

Nous souhaitons maintenant démontrer que le problème 3-SAT (dont nous présentons deux versions légèrement différentes) est NP- complet. Dans sa première version le problème 3-SAT est celui de la satisfiabilité des formules F ayant des clauses qui peuvent être composées d'un littéral, de deux littéraux ou de trois littéraux. Dans sa deuxième version que nous appelons 3-SAT exacte, chaque clause a exactement 3 littéraux.

1. Démontrer que le problème 3-SAT est NP-complet.
2. En conclure que le problème 3-SAT exacte est NP-complet.

**Corrigé**

1. Nous démontrons qu'il existe une transformation polynomiale du problème SAT vers 3-SAT. Pour cela, nous transformons un clause donnée appartenant au problème SAT ainsi : $C_i = L_1 \vee L_2 \vee .. \vee L_n$, nous regroupons les littéraux deux à deux et nous obtenons : $C_i = (Y_1 \vee .. \vee Y_{\frac{n}{2}}) \wedge (Y_1 \leftrightarrow (L_1 \vee L_2) \wedge ... Y_i \leftrightarrow (L_{2i-1} \vee L_{2i}) \wedge ...)$
$Y_1 \leftrightarrow (L_1 \vee L_2) = (Y_1 \rightarrow (L_1 \vee L_2)) \wedge ((L_1 \vee L_2) \rightarrow Y_1) = (\neg Y_1 \vee L_1 \vee L_2) \wedge (\neg L_1 \vee Y_1) \wedge (\neg L_1 \vee Y_1)$

   Pour chaque littéral $Y_i$ trois clauses ont été ajoutées une contenant 3 littéraux et deux contenant deux littéraux. Nous continuons le regroupement des littéraux jusqu'à ce que nous arrivons à une FNC contenant une clause de 2 ou trois littéraux...

2. Nous démontrons qu'il existe une transformation polynomiale du problème 3-sat vers le problèmes 3-sat-exact :
   (a) Pour les clauses de la forme $C_i = L_1 \vee L_2 \vee L_3$, rien à faire.
   (b) Pour les clauses de la forme $C_i = L_1 \vee L_2$, en ajoutant un littéral et sa négation : $C_i = (L_1 \vee L_2) \vee (p \wedge \neg p)$, $C_i = (L_1 \vee L_2 \vee p) \wedge (L_1 \vee L_2 \vee \neg p)$.
   (c) Pour les clauses de la forme $C_i = L_1$, en ajoutant un littéral et sa négation nous obtenons : $C_i = (L_1 \vee p) \wedge (L_1 \vee \neg p)$. En ajoutant encore un littéral et sa négation : $C_i = (L_1 \vee p \vee (q \wedge \neg q)) \wedge (L_1 \vee \neg p \vee (q \wedge \neg q))$. Nous obtenons :

   $$C_i = (L_1 \vee p \vee q) \wedge (L_1 \vee p \vee \neg q) \wedge (L_1 \vee \neg p \vee q) \wedge (L_1 \vee \neg p \vee \neg q)$$

   .

# 3 Le problème de la clique

Etant donné un graphe G(S,A), G possède-t-il une clique ? c'est-à- dire un sous-graphe complet.

Démontrer que ce problème est NP-complet.