Leçon 6.
Problèmes NP-complets

Denis Trystram

October 4, 2012

Motivation : caractériser la classes des problèmes les plus difficiles (au sens de la réduction de Karp) dans le but d'une classification.

1 Présentation générale

2 La classe $\mathcal{NP}$-complet

La réduction polynomiale permet de construire des classes d'équivalence. Etablir qu'un langage L appartient à $\mathcal{NP}$-complet consiste à montrer que L est "le" langage le plus difficile de $\mathcal{NP}$, au sens où il est un élément maximum de $\mathcal{NP}$ pour la réduction polynomiale.

Définition 1 *Un langage L appartient à $\mathcal{NP}$-complet ssi il appartient à $\mathcal{NP}$ et si tout langage de $\mathcal{NP}$ se réduit polynomialement à lui :*

$$L \in \mathcal{NP} \ \text{ et } \ \forall L' \in \mathcal{NP}, \quad L' \propto L$$

Il est facile de vérifier que $\mathcal{NP}$-complet est une classe d'équivalence pour la réduction polynomiale, puisque par définition si deux langages L_1 et L_2 appartiennent à $\mathcal{NP}$-complet, on a $L_1 \propto L_2$ et $L_2 \propto L_1$. Evidemment, notre définition n'indique pas s'il existe ou non des langages dans cette classe... En effet la relation $\propto$ étant simplement un pré-ordre et non un ordre total, *a priori* il peut exister deux langages maximaux pour la réduction polynomiale, qui soient par ailleurs incomparables. Dans une telle configuration, $\propto$ n'admettrait pas d'élément maximum ($\mathcal{NP}$-complet $= \emptyset$) mais uniquement des éléments maximaux.

Par contre si $\mathcal{NP}$-complet n'est pas vide, cette classe contient les langages les plus "difficiles" de $\mathcal{NP}$, au sens où si l'on peut décider n'importe lequel de ces langages en temps polynomial, alors tout langage de $\mathcal{NP}$ peut être décidé en temps polynomial.

Propriété 1 *Si un langage de $\mathcal{NP}$-complet peut être décidé par un algorithme polynomial, alors tous les langages de $\mathcal{NP}$ sont décidables en temps polynomial :*

$$\mathcal{P} \cap \mathcal{NP}\text{-}complet \neq \emptyset \Rightarrow \mathcal{P} = \mathcal{NP}$$

Si la question de savoir si $\mathcal{NP}$-complet est non vide a été résolue par le célèbre théorème de Cook que nous présentons à la section suivante, la question de savoir si $\mathcal{NP}$-complet et $\mathcal{P}$ sont disjoints est toujours d'actualité. Ladner [**?**] a montré que soit $\mathcal{P} = \mathcal{NP}$, soit il existe une infinité de classes d'équivalences distinctes sur $\mathcal{NP}$, comprenant entre autres $\mathcal{P}$ et $\mathcal{NP}$-complet. De plus, dans ce cas on peut montrer que ces classes sont denses. Cela signifie que pour tout couple de problèmes P_1 et P_2 dans $\mathcal{NP}$ non équivalents et tels que $P_1 \propto P_2$ alors, il existe un autre problème $P_{1,2}$ de $\mathcal{NP}$ tel que $P_1 \propto P_{1,2} \propto P_2$ non équivalents à P_1 et P_2. Ladner a également montré qu'il existait des problèmes de $\mathcal{NP}$ non comparables entre eux. Enfin, même si $\mathcal{NP}$ admet un plus grand et un plus petit élément (respectivement $\mathcal{P}$ et $\mathcal{NP}$-complet), il n'a pas une structure de Treillis.

Nous pouvons donner une première structure grossière pour $\mathcal{NP}$ comme représentée dans la figure **??** ci-dessous.

2.1 Frontière entre $\mathcal{P}$ et $\mathcal{NP}$

Problématique : l'étude de la complexité de certains problèmes permettent de conclure qu'ils sont polynomiaux, d'autres qu'ils sont NP-complets. Pour ces derniers, en l'état des connaissances aujourd'hui, il est inenvisageable de trouver des algorithmes efficaces pour les résoudre : motivation pour étudier des problèmes à la frontière : relaxer ces problèmes pour obtenir des sous-problèmes polynomiaux...
On prend l'exemple d'un problème d'ordonnancement général : le problème $P|prec, pj = 1|Cmax$ est polynomial pour deux machines et il est NP-complet pour m arbitraire. Sa complexité est inconnue pour $m = 3$.

On rappelle le problème de l'ordonnancement à 2 machines :

O2M
Instance. n tâches indépendantes de durées p_j (entières) pour $1 \leq j \leq n$, D un entier.
Question. Existe-t-il un ordonnancement réalisable de durée inférieure ou égale à D ?

On réduit facilement ce problème à partir de 2-Partition.
On peut également introduire un élément de comparaison supplémentaire avec le problème du Sac-à-dos. Il est possible de montrer formellement l'équivalence entre ces trois problèmes (Partition, Sac-à-dos et O2M). En particulier, comme 2-Partition était NP-complet, les deux autres se sont également...

3 Ce qu'il faut retenir

Montrer qu'un probléme est NP-completse fait en deux temps : appartenance à NPpuis trouver une réduction