Calculabilité et Complexité: CM7

Florian Bridoux

Université de Caen

2021-2022

Table des matières

1. Réduction polynomiales
2. Complétude
3. Le problème SAT
4. Autres problèmes NP-complets
5. coNp-complets

Table des matières

1. Réduction polynomiales
2. Complétude
3. Le problème SAT
4. Autres problèmes NP-complets
5. coNp-complets

Réduction polynomiales

Définition (réductions many-one polynomiales)

Une réduction many-one en temps polynomial d'un problème A (sur l'alphabet Σ_A) vers un problème B (sur l'alphabet Σ_B) est une fonction $f : \Sigma_A^* \to \Sigma_B^*$ *calculable en temps polynomial* telle que :

$$\forall x \in \Sigma_A^*, x \in A \Leftrightarrow f(x) \in B.$$

Si une telle fonction f existe alors on dit que B se réduit à B et on note $A \leq_m^P B$.

Réduction polynomiales

Proposition:

Supponsons que $A \leq_m^P B$ et que $B \in \mathscr{C}$ avec $\mathscr{C} \in \{P, NP, coNP, EXP, NEXP, coNEXP\}$. Alors $A \in \mathscr{C}$.

Idée de la démonstration: Pour décider si $x \in A$, il suffit de calculer $f(x)$ (ce qui se fait en temps polynomial de manière déterministe), puis de lancer M_B qui reconnaît B sur $f(x)$.

Remarque:

Pour prouver que A est dans $\mathscr{C}$ (est "facile"), il suffit de donner un algorithme qui reconnaît A dans la bonne classe de complexité. On utilisera surtout les réductions dans l'autre sens: pour prouver que B est "difficile".

Réduction polynomiales

Proposition:

La relation $\leq_m^P$ est réflexive et transitive.

- Réflexivité: Soit A un langage, alors $A \leq_m^P A$ via l'identité.
- Transitivité: soit A, B et C des langages tels que $A \leq_m^P B$ via f et $B \leq_m^P C$ via g. Alors $A \leq_m^P C$ via $g \circ f$.

Définition:

Si deux langages A et B vérifient $A \leq_m^P B$ et $B \leq_m^P A$ alors on notera $A \equiv_m^P B$ et on dira que A et B sont équivalents pour les réductions many-one polynomiales.

Réduction polynomiales

CLIQUE:
Entrée: un graphe non orienté G et un entier k ;
Question: G a une clique de taille k ?

ENSEMBLE_INDÉPENDANT:
Entrée: un graphe non orienté G et un entier k ;
Question: G a un ensemble indépendant de taille k (tous non reliés deux à deux)?

Exercice: Prouver que CLIQUE $\equiv_m^P$ ENSEMBLE_INDÉPENDANT.

Table des matières

1. Réduction polynomiales
2. Complétude
3. Le problème SAT
4. Autres problèmes NP-complets
5. coNp-complets

Difficulté et complétude

Définition (difficulté et complétude):

Soit $\leq$ une notion de réduction entre problèmes. Soit A un problème et $\mathscr{C}$ une classe de complexité.

- A est dit $\mathscr{C}$-difficile (ou $\mathscr{C}$-dur) pour les réductions $\leq$ si pour tout $B \in \mathscr{C}$, on a $B \leq A$.
- A est dit $\mathscr{C}$-complet pour les réductions $\leq$ s'il est $\mathscr{C}$-difficile pour les réductions $\leq$ et si $A \in \mathscr{C}$.

Cette notion dépend du type de réductions $\leq$. Par défaut, si l'on ne précise pas le type de réductions, il s'agira des réductions many-one en temps polynomial $\leq_m^P$.

Difficulté et complétude

Remarque:

- La condition est très forte : il faut que tous les problèmes de $\mathscr{C}$ se réduisent à A. À priori, rien ne dit qu'il existe des problèmes $\mathscr{C}$-difficiles ni (a fortiori) $\mathscr{C}$-complets. Cependant, il existe de nombreux problèmes naturels complets pour les classes vues jusqu'à présent, notamment NP.
- Attention, si un langage A est hors de $\mathscr{C}$, cela n'implique pas qu'il est C-difficile ! Ce sera cela-dit le cas de tous les problèmes que nous verrons pendant ce cours.

Difficulté et complétude

Montrons que cette notion est inutile pour la classe P.

Proposition:

Tout problème B non trivial (c'est-à-dire $B \neq \emptyset$ et $B \neq \Sigma^*$) est P-difficile pour les réductions many-one en temps polynomial.

Proof.

Soit B non trivial, $x^0 \notin B$ et $x^1 \in B$. Soit $A \in P$: montrons que $A \leq_m^P B$. La réduction f de A vers B est alors définie comme suit :

$$f(x) = \begin{cases} x^1 \text{ si } x \in A \\ x^0 \text{ sinon} \end{cases}$$

Puisque $A \in P$, cette fonction f est calculable en temps polynomial. Par ailleurs, si $x \in A$ alors $f(x) = x^1 \in B$ et si $x \notin A$ alors $f(x) = x^0 \notin B$: ainsi, $x \in A$ ssi $f(x) \in B$, donc f est une réduction de A à B. □

Difficulté et complétude

Explication

Ce résultat vient du fait que la réduction est trop puissante pour la classe P : pour avoir une pertinence, il faudra donc réduire la puissance des réductions. Pour la classe P on utilise généralement la réduction many-one en espace logarithmique. Il s'agit d'un phénomène général : plus la classe est petite, plus il faut utiliser des réductions faibles pour comparer les problèmes de la classe.

Table des matières

1. Réduction polynomiales
2. Complétude
3. Le problème SAT
4. Autres problèmes NP-complets
5. coNp-complets

Difficulté et complétude

Définition: formule CNF (conjunctive normal form)

Une formule normale conjonctive (ou CNF) sur un ensemble de *variables* $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ est une formule de la forme:

$$\mu = \mu_1 \wedge \mu_2 \wedge \cdots \wedge \mu_m.$$

Chaque formule μ_j (appelé *clause*) est elle même une disjonction de *littéraux*:

$$\mu_j = \mu_{j,1} \vee \mu_{j,2} \vee \cdots \vee \mu_{j,m_j}.$$

Chaque littéral $\mu_{j,k}$ correspond lui même à une variable λ_i et peut-être de polarité positive (on a alors $\mu_{j,k} = \lambda_i$) ou négative (on a alors $\overline{\lambda_i}$)

Difficulté et complétude

Définition:

Une *valuation* est une fonction $v : \lambda \to \{0, 1\}$.
Une formule CNF μ est satisfiable s'il existe une valuation de ses variables qui la rend vraie.

Exemples:

- La formule CNF $\mu = \lambda_1 \wedge \overline{\lambda_1}$ n'est pas satisfiable.
- Pour la formule CNF $\mu = (\lambda_1 \vee \lambda_2) \wedge (\overline{\lambda_2} \vee \overline{\lambda_2})$ est satisfiable:
  - les valuations v et v' avec $v(\lambda_1) = v(\lambda_2) = 1$ et $v'(\lambda_1) = v'(\lambda_2) = 0$ ne satisfont pas la formule.
  - les valuations u et u' avec $u(\lambda_1) = 1, u(\lambda_2) = 0$ et $u'(\lambda_1) = 0 u'(\lambda_2) = 1$ satisfont la formule.

Difficulté et complétude

Problème SAT:
Entrée: Une formule CNF μ.
Question: μ est-elle satisfiable?

Théorème de Cook

Le problème SAT est NP-complet.

Ce théorème a été démontré en 1971 par Stephen Cook.

Difficulté et complétude

Le problème SAT est dans NP car la machine de Turing non déterministe suivante le décide en temps polynomial :

- Lire l'expression booléenne μ.
- Pour toute variable λ_i apparaissant dans μ, choisir de manière non déterministe une valeur $v(\lambda_i)$ dans $\{0, 1\}$.
- Accepter si la valuation v satisfait μ et refuser sinon.

Difficulté et complétude

Preuve que SAT est NP-dur:
Soit A un problème dans NP. Il existe donc une machine de Turing non déterministe M qui le décide en temps polynomial : pour toute instance x de A, x est une instance positive de A si et seulement s'il existe une exécution acceptante de M avec x sur le ruban d'entrée de longueur polynomiale en $|x|$. L'idée est donc de construire effectivement en temps polynomial une formule booléenne $\mu^{M,x}$ qui dit intuitivement " il existe une exécution acceptante de M avec x sur le ruban d'entrée de longueur polynomiale en $|x|$ ". Ainsi, x est une instance positive de A si et seulement si $\mu^{M,x}$ est satisfiable. Ainsi, on a une réduction polynomiale de A dans SAT : SAT est donc NP-dur.

Difficulté et complétude

Le théorème de Cook est la première preuve de NP-complétude. Les autres preuves de NP-complétude se font généralement par réduction polynomiale d'un problème déjà classé comme NP-complet. En effet, pour montrer la NP-difficulté d'un problème A, il suffit maintenant de prouver que SAT $\leq_m^P A$.

Table des matières

1. Réduction polynomiales
2. Complétude
3. Le problème SAT
4. Autres problèmes NP-complets
5. coNp-complets

3-SAT

Une formule 3-CNF est une formule CNF ou chaque clause est composé de 3 littéral ou moins.
Exemple:

$$\mu = (\lambda_1 \vee \lambda_2 \vee \overline{\lambda_3}) \wedge (\overline{\lambda_1} \vee \lambda_2 \vee \overline{\lambda_3}) \wedge (\lambda_1 \vee \overline{\lambda_2} \vee \overline{\lambda_3}) \wedge (\lambda_3)$$

3-SAT

Problème 3-SAT:
Entrée: Une formule 3-CNF μ.
Question: μ est-elle satisfiable?

Théorème

Le problème 3-SAT est NP-complet.

3-SAT

- 3-SAT est bien sûr dans NP, car les 3-CNF sont des CNF particulières.
- Pour réduire de SAT vers 3-SAT il faut "couper" les clauses de 4 littéraux ou plus en plusieurs littéraux. (Il n'y a pas de magie: on va devoir utiliser plus de variables).
- Par exemple si on a une clause d'au moins $n \geq 4$ littéraux:

$$\lambda_1 \vee \lambda_2 \vee \lambda_3 \vee \cdots \vee \lambda_n$$

on peut la remplacer par les deux clauses:

$$(\lambda_1 \vee \lambda_2 \vee \lambda_c) \wedge (\lambda_3 \vee \cdots \vee \lambda_n \vee \overline{\lambda_c}).$$

- Si v est une valuation de notre nouvelle formule:
  - Soit $v(\lambda_c) = 0$ et donc on a $\lambda_1 \vee \lambda_2$ qui doit valoir vrai.
  - Soit $v(\lambda_c) = 1$ et donc on a $\lambda_3 \vee \cdots \vee \lambda_n$ qui doit valoir vrai.

  La formule est bien équivalente.
- Si $n = 4$ c'est gagné. Sinon, on refait le même découpage récursivement.

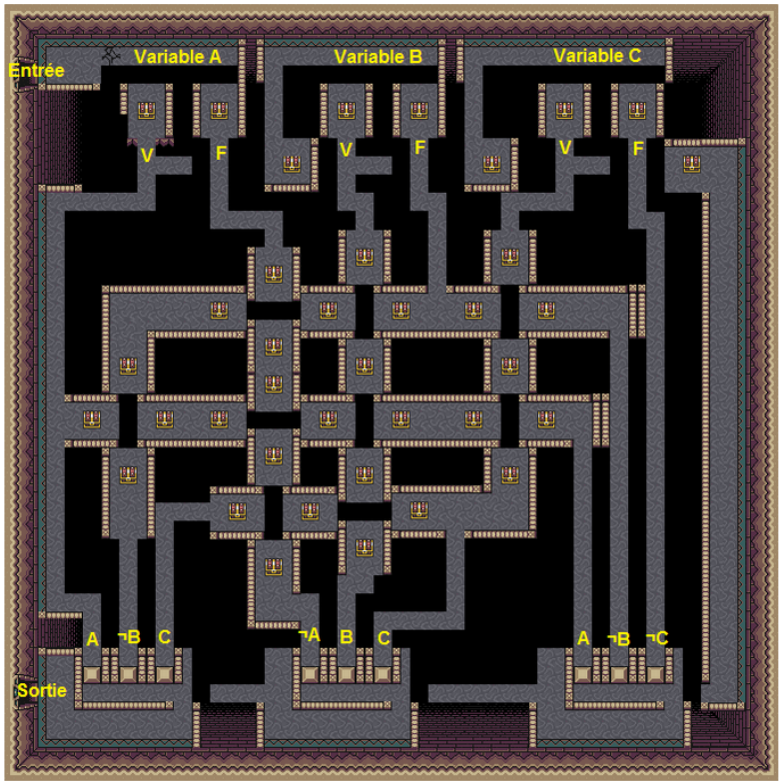
2-SAT

Remarque

On verra en TD que le problème 2-SAT n'est pas NP-dur et est en fait dans P (donc pas NP-complet à moins que $P = NP$).

NP-difficulté de jeux célèbres

Étant donné un niveau dans un jeu type mario, zelda, pokemon, etc... on peut se demander s'il est possible de gagner le niveau. En faisant une réduction depuis 3-SAT, on peut prouver que ces jeux sont en faite très difficile: Ils sont NP-difficiles.



Lien vers le billet de blog

Table des matières

1. Réduction polynomiales
2. Complétude
3. Le problème SAT
4. Autres problèmes NP-complets
5. coNp-complets

coNP-complets

Théorème

L est NP-complet ssi son complémentaire $\overline{L}$ est coNP-complet.

Preuve: (sens gauche $\Rightarrow$ droite) Si L est dans NP alors $\overline{L}$ est dans coNP par définition. Soit $\overline{A}$ un problème dans coNP. Donc son complémentaire A est dans NP et $A \leq_m^P L$ via une fonction f. On a $x \in A$ iff $f(x) \in L$. Donc $x \in \overline{A}$ iff $f(x) \in \overline{L}$. Donc $\overline{A} \leq_m^P \overline{L}$ via la fonction f.

DNF

Définition: formule DNF (disjunctive normal form)

Une formule normale disjonctive (ou DNF) sur un ensemble de *variables* $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ est une formule de la forme:

$$\mu = \mu_1 \vee \mu_2 \vee \cdots \vee \mu_m.$$

Chaque formule μ_j (appelé *clause*) est elle-même une disjonction de *littéraux*:

$$\mu_j = \mu_{j,1} \wedge \mu_{j,2} \wedge \cdots \wedge \mu_{j,m_j}.$$

Chaque littéral $\mu_{j,k}$ correspond lui-même à une variable λ_i et peut-être de polarité positive (on a alors $\mu_{j,k} = \lambda_i$) ou négative (on a alors $\overline{\lambda_i}$)

coNp-complets

Une tautologie est une formule qui est toujours vraie. Exemple:

- Les jeunes gens sont jeunes.
- Si c'est trop tard, c'est trop tard.
- $\lambda_i \vee \overline{\lambda_i}$.

Une formule μ sur des variables λ.

Problème TAUTOLOGIE:
Entrée: Une formule DNF μ.
Question: μ est-elle une tautologie?

Théorème

Le problème TAUTOLOGIE est NP-complet.

Théorème

Le problème 3-TAUTOLOGIE est également NP-complet.