Complexité - Nicolas Ollinger

Alexandre Masson

15 Janvier 2013

Table des matières

1 non-déterminisme

$P = \cup_{k>0} DTIME(n^k)$ la classe des problèmes qu'on peut résoudre en temps polynomial.
$NP = \cup_{k>0} NTIME(n^k)$ la classe des problèmes vérifiables en temps polynomial

Rétrospective DFA : automate déterminisme , NFA non déterminisme.
ex : $(a+b)^*baba$ mot finissant par baba. Tout NFA non déterminisme, se code en DFA, mais au prix d'une complexité plus importante en nombre d'état.

Définition une MTND (MT non déterministe), est une MT avec comme table de transition :

$$T \subseteq (Qx\Gamma^k)^{input} x (Qx\Gamma^k x\{<-,v,->\})^{output}$$

on redéfinie la relation de transition :

$c \vdash c'$ si T contient une transition applicable sur c qui produit c'

Une exécution d'une MTND partant d'une entrée u est une suite de transition valides de la configuration initiale associée à u à une configuration d'arrêt.

$$c_0 \vdash c_1 \vdash ... \vdash c_n$$

Une MTND M accepte un mot $u \in \Sigma^*$ s'il existe une exécution acceptée pour u.

	MT	MTND
Rejet	l'exécution doit s'arrêter sur q_{non}	toutes les exécutions doivent s'arrêter sur q_{non}
Acceptation	l'exécution s'arrête sur q_{oui}	un seul état à oui renvoie oui

remarque A un ralentissement linéaire près, on peut supposer que les choix non déterministes sont au plus binaires : pour tout configuration c :
- ou bien c n'as pas d'image (arrêt)
- ou bien $c \vdash c'$ unique.
- ou bien $c \vdash c' \& c \vdash c''$

exemple le 3-color, entrée : un graphe G=(V,E), sortie : G est il 3-colorable
3Color est reconnu par une MTND dont les exécutions s'arrêtent en temps polynomial en la taille de l'instance.
la machine fonctionne en deux temps.
- Choix non déterministe d'une couleur par sommet (O(|v|)
- Vérification déterministe que le coloriage choisit est valide : si oui accepte, sinon rejette.

Définition NTIME(f(n)) est la classe des langages reconnus par une MTND dont toutes les exécutions sont de longueur $\leq$ f(n) sur les entrées de taille n.

proposition Si $f(n) \leq g(n)$ alors $NTIME(f(n)) \subseteq NTIME(g(n))$
$DTIME(f(n)) \subseteq NTIME(f(n))$
$NTIME(f(n)) \subseteq DTIME(2^{O(f(n))})$
Idée : Il suffit de simuler les choix de la MTND par du backtracking

def $NP = \cup_{k>0} NTIME(n^k)$
$NEXP = \cup_{k>0} NTIME(2^{n^k})$
Question : Est ce que P NP, **rq :** $P \subseteq NP$
prop NP ets clos par union et intersection.
def $co\Im = \{!L | L \in \Im\}$ ou $\Im$ et une classe de langages.

Un problème B est un **certificateur** pour un problème A si $\forall x \in \Sigma^*, x \in A <=> \exists y \in \Sigma^* < x, y > \in B$ on dit que y est un certificat pour x.

thm Np la classe des langages qui possèdent un certificateur dans P avec certificats de longueur polynomiale en la longueur de l'instance.
ex 3Color ∈ NP , cf exemple antérieur. certificateur pour 3COLOR.
sur l'entrée <G,C> avec colorisation des sommets vérifier que les arêtes ont des couleurs distinctes à chaque extrémité. Ce certificat est dans P(car il ne fais que parcourir les arêtes pour vérifier une propriété). la taille du certificat est polynomiale.

Démonstration

i (dans ce sens, on connait A de P , et B le certificateur, on essaie de construire un chemin qui accepte x dnas la machine M, ce chemin s'appelle y , si l'on suis le chemin y pour arrive en qoui ave x en entrée, alros y est certificat et le certif existe , donc B est certificateur.)
Soit A un langage qui possède un certificateur B ∈ P avec certificat de longueur ≤ p(n) sur les instances de longueur n ou p est polynomiale en n.
montrons que A ets dans NP en construisant une MTND M qui reconnait A comme suit :

- choisir grace au non déterminisme un mot y de longueur ≤ p(|x|)
- exécuter une MT qui reconnait B sur l'entrée <x,y>
- accepter si (2) accepte

M s'exécute bien en temps polynomial en |x| et M accepte x ssi $\exists y : |y| \leq p(n)$ et <x,y> ∈ B, si x∈ A.

ii (dans ce sens, on sais que le Y sera un certificat, puisqu'on chosiit un chemin acceptant de x dans l'exécution de M, donc si B repond ok , vu que le Y ets ok , cela veux dir que B est bon)
Soit $A \in NP$ On va construire un certificat $B \in P$ pour A avec certificats polynomiaux.
Soit M un MTND qui reconnais A.
On utilise les exécutions acceptantes de M comme certificats et B se content de

vérifier l'exécution. Sur l'entrée <x,y>B vérifie que Y est une exécution acceptable pour M sur l'entrée x. B est bien un certificateur dans P avec certificat polynomiaux pour A.
Np est la classe des problèmes vérifiables en temps polynomial.

3Color : E : graphe G, S : est il 3Colorable, certif : coloration, Verif : pas d'arete monochrome.

Hamilton : E/S idem : certif : une permutation des sommets, i.e le cycle, verif : les aretes existent bien.

CLIQUE(sous graphe complet du graphe) Entrée : un graphe et un entier k : certif : la clique, verfi : les aretes existe =elles

SAT (clause satisfiables) entrée : formule Φ en CNF, sortie : Φ est elle satisfaisable ?
variable : x,y,z, littéraux : var ou neg.var, clause : disjonction de littéraux, formules : conjonction de clauses, **certificat :** une affectation qui satisfait Φ, vérif : évaluer Φ **3SAT** SAT où Φ ont des clause d'ua moins 3 littéraux.

COMPOSITE entrée : entier n en binaire, question : existe il p et q > 1 tels que n=pq ?, certif : la paire <p,q> en binaire. vérif : multiplier p par q et comparer à n.

2 19 Mars 2013

2.1 NP-Completude

Au menu : théorème de Cook-Levin SAT$\in$ P <=> P=NP
remarque sens <= trivial car SAT $\in$ NP.
Définition $A \leq_p B$ si $\exists f : \Sigma^* -> \Sigma^*$ totale calculable en temps polynomiale telle que :$\forall x \in \Sigma^*, X \in A <=> f(x) \in B$

prop si $A \leq_p B$ et B $\in$ P alors A $\in$ P. Que B soit dans P implique qu'il existe un algo poly qui résout B,

def on dit que A est NP-difficile si quelquesoit le problème que je prend dans NP , ce problème est plus simple que A.($\forall B \in NP, B \leq_p A$
A est NPComplet si
A est NP-difficile **ET** A$\in$ NP.(**NE PAS L'OUBLIER**)

Remarque : si A est NP-complet, et A $\in$ P, alors P=NP. En effet soit B $\in$ NP, comme A est NP-difficile, d'après la def, $B \leq_p A$ et d'après la proposition

ci dessus, B∈P . Donc $NP \subseteq P$, i.e P=NP.

thm Cook-Levin SAT est NP-complet. On sais que SAT ∈ NP, on va montrer que $\forall A \in NP, A \leq_p SAT$
soit $A \in NP$.soit M une MTND totale qui reconnaît A en temps polynomial p(n).
On peux supposer sans perte de généralité que M à un seul ruban, à choix binaire, à ruban semi infini, avec arrêt sur la case 0, exactement au temps p(n). A l'instant 0 , le ruban contient le mot de taille N, si le temps d'arrêt est p(n), alors la ruban ne peux etre de taille plus grand que p(n) , car on parcours une case par étape , donc pour un mot donnée, on av obtenir des q_{oui} ou des q_{non} , $x \in A$ ssi ∃ un digne-espace temps de M sur l'entrée X, avec q_{oui} en (0,p(n)).

Idée : Pour montrer que $A \leq_p SAT$ on construit f, fonction de réduction , qui a un mot $x \in \Sigma^*$ associe une formule CNF $\Phi_x^{M,p}$ qui code les diagrammes espace-temps de M sur l'entrée x de sorte que $\Phi_x^{M,p} \in SAT$ ssi le diagramme possède des choix non-déterministes acceptant , c'est a dire avec q_{oui} en (0,p(x)).

On va découper $\Phi_x^{M,p}$ en plusieurs parties :
$\Phi_{sim}^{M,p(|x|)}(calculdelaMT) \wedge \Phi_{input}^{M,P,x}(entrée) \wedge \Phi_{accept}^{M,P(|x|)}(sortie)$

Pour chaque case du diagramme, elle sont codées par les variables booléennes, $x_1^{ij}, ..., x_k^{ij}$, en binaire.
les codes valides sont vérifiés par une formule CNF $\Phi_{val}(x_1^{ij}, ..., x_k^{ij})$
$\Phi_{sim}^{M,p(|x|)} = \wedge_{i=0}^{N} \wedge_{j=0}^{N} \Phi_{val}(x_1^{ij}, ..., x_k^{ij}) \wedge \Phi_{trans}^{M,N}$
la variable C_t code le choix non-det à l'instant t. (si le contenu est loin de la tête , il n'a aucune chance de changer entre t et t+1.
$\Phi_{trans}^{M,N} = \wedge_{i=0}^{N} \wedge_{j=0}^{N} \Phi_{delta}^{M,N}(x_1^{i-1j}, ..., x_k^{i-1j}, x_1^{ij}, ..., x_k^{ij}, x_1^{i+1j}, ..., x_k^{i+1j}, x_1^{i-1j+1}, ..., x_k^{i-1j+1}, x_1^{ij+1}, ..., x_k^{ij+1}, x_1^{i+1j+1}, ..., x_k^{i+1j+1})$
où Φ_{delta}^{M} vérifie que que le rectangle 3x2 cellules communiqués est une transition valide pour M

On obtient une formule CNF $\Phi_{sim}^{M,N}$ à kN^2 variables et de taille O(N^2). On est tous convaincu de savoir produire $\Phi_{sim}^{M,N}$ en temps poly.

$\Phi_{input}^{M,P,x}$ vérifie que la ligne 0 code via la config initial
$\Phi_{input}^{M,P,x} = \Phi_{case}^{M}(q_0, x_0), x_1^{00}, ..., x_1^{00} \wedge \wedge_{i=1}^{|x|-1} \Phi_{case}^{M}(x_i, x_1^{i0}, ..., x_k^{i0}) \wedge \wedge_{i=|x|}^{p(|x|)}(B, x_1^{i0}, ..., x_k^{i0})$
formule de ligne $O(n)^i = |x|$ est calculable en temps polynomial

$\Phi_{accept}^{M,N}$ vérifie que la case (0,n] contient bien q_{oui}
la formule $\Phi_x^{M,N}$ est donc calculable en temps polynomial en |x| et est satisfiable ssi l existe des choix non déterministes pour lequels M accepte x, i.e x in A.

Si A ets NP-diificile et $A \leq_p B$ alors B ets NP difficile, car B est plus compliqué que A., pour montrer que BLOP ets np complet, on montre que blop in NP et blip $\leq_p$ BLOP , comme blop plus compliqué que blip et blip np , alors blop np
prop 3SAT et variante est NP-complet,
une formule CNF avec jsuqu'a 3 littéraux par clause
idem mais avec exactement
idem mais avec 3 littéraux distincts

démonstration
3SAT in NP : cf cours précédent
3SAT est NP-difficile, montrons $SAT \leq_p 3SAT$
Exhibons F calculable en temps ploynomial telle que
$\forall \Phi, \Phi \in SAT <=> f(\Phi) \in 3SAT$
no doit trouver un moyen de transformer les littéraux de tailel n en littéraux de taille 3 avec des conjonctions entre eux, pour respecter CNF.

F remplace chaque clause $(l_1 v l_2 v ... v l_k)$ par la formule $(l_1 v l_2 v \alpha_2) \wedge (\bar{\alpha_2} v l_3 n \alpha_3) \wedge ... \wedge (\bar{\alpha_{k-2}} v l_{k-1} v l_k)$ où $\alpha_2, ..., \alpha_{k-2}$ sont k-3 nouvelles variables, et cette formule est satisfiable ssi la clause l'est.