

Calculabilité et Logique

TD12 – Algorithmes de démonstration automatique

17 décembre 2008

Exercice 1 – *HORN-SAT*

Décrire une simplification de l'algorithme de DPLL qui décide le problème suivant HORN-SAT en temps polynomial :

ENTRÉE : un ensemble fini, S , de clauses de Horn (c'est-à-dire clauses contenant au plus un littéral positif) ;

QUESTION : S est-il satisfiable ?

Donc HORN-SAT $\in \mathbf{P}$.

Exercice 2 – *3-SAT-NON-TRIV*

On considère le langage 3-SAT-NON-TRIV :

ENTRÉE : un ensemble non vide S de 3-clauses, dont aucune n'est une tautologie, aucune n'est une clause unitaire (réduite à un littéral), aucune n'est vide ($\square$), et aucune n'est pure.

QUESTION : S est-elle satisfiable ?

Montrer que 3-SAT-NON-TRIV est $\mathbf{NP}$ -complet.

Exercice 3 – *2SAT*

Montrer que le problème 2-SAT :

ENTRÉE : une liste finie, S , de 2-clauses ;

QUESTION : S est-elle satisfiable ?

est dans $\mathbf{P}$. On rappelle qu'une 2-clause est une clause contenant au plus deux littéraux.

Exercice 4 – *HORN-REN-SAT*

Soit S un ensemble de clauses, avec $\text{FV}(S) = \{A_0, A_1, \dots, A_{n-1}\}$. Un *renommage* R de S est un sous-ensemble de $\text{FV}(S)$. L'application $R[S]$ du renommage R à un littéral, une clause, ou un ensemble de clauses S est défini comme l'objet obtenu en changeant le signe de toutes les occurrences des atomes de R , et en laissant inchangé le signe des autres. Par exemple, si $R = \{A_1\}$ et $S = \{+A_1 \vee +A_2, -A_1 \vee -A_0\}$, on a $R[S] = \{-A_1 \vee +A_2, +A_1 \vee -A_0\}$. Disons qu'un ensemble de clauses S est *Horn-renommable* si et seulement s'il existe un renommage R tel que $R[S]$ soit un ensemble de clauses de Horn.

On considère le langage HORN-REN-SAT :

ENTRÉE : un ensemble de clauses Horn-renommable S ;

QUESTION : S est-elle satisfiable ?

Montrer que HORN-REN-SAT $\in \mathbf{P}$. Attention : on rappelle que l'on doit décider non seulement si S est satisfiable, mais aussi si S est de la forme requise en ENTRÉE.

Exercice 5 – SAT-2-OCC

On a vu que le problème 3-SAT-3-OCC était NP-complet. Montrer que le problème analogue SAT-2-OCC est, lui, dans P :

ENTRÉE : une liste finie, S , de clauses, où chaque variable propositionnelle apparaît au plus 2 fois ;

QUESTION : S est-elle satisfiable ?

Exercice 6 – Construction de BDD

Rappel : Si Φ et Φ' sont des BDD, les fonctions $\text{BDDneg}(\Phi)$ et $\text{BDDor}(\Phi, \Phi')$ calculant respectivement les BDD $\neg\Phi$ et $\Phi \vee \Phi'$, sont définies comme suit :

BDDneg :

- $\text{BDDneg}(\mathbf{1}) = \mathbf{0}$, $\text{BDDneg}(\mathbf{0}) = \mathbf{1}$,
- $\text{BDDneg}(A \longrightarrow \Phi_+; \Phi_-) = \text{BDDmake}(A, \text{BDDneg}(\Phi_+), \text{BDDneg}(\Phi_-))$.

BDDor :

- $\text{BDDor}(\mathbf{1}, \Phi) = \mathbf{1}$, $\text{BDDor}(\mathbf{0}, \Phi) = \Phi$;
- $\text{BDDor}(\Phi, \mathbf{1}) = \mathbf{1}$, $\text{BDDor}(\Phi, \mathbf{0}) = \Phi$;
- si $\Phi = A \longrightarrow \Phi_+; \Phi_-$, et $\Phi' = A' \longrightarrow \Phi'_+; \Phi'_-$, alors :
 - si $A < A'$, alors
 $\text{BDDor}(\Phi, \Phi') = \text{BDDmake}(A, \text{BDDor}(\Phi_+, \Phi'_+), \text{BDDor}(\Phi_-, \Phi'_-))$;
 - si $A > A'$, alors
 $\text{BDDor}(\Phi, \Phi') = \text{BDDmake}(A', \text{BDDor}(\Phi, \Phi'_+), \text{BDDor}(\Phi, \Phi'_-))$;
 - si $A = A'$, alors
 $\text{BDDor}(\Phi, \Phi') = \text{BDDmake}(A, \text{BDDor}(\Phi_+, \Phi'_+), \text{BDDor}(\Phi_-, \Phi'_-))$.

1. Définir de manière analogue la fonction BDDand .
2. Montrer que les opérations BDDor , BDDneg , BDDand , sont toutes en temps polynomial (sur une machine à accès direct, et en supposant que les recherches dans les diverses tables de hachage sont en temps constant).

Exercice 7 – Satisfiabilité à l'aide des BDD

Pour convertir une formule propositionnelle F en un BDD $BDD(F)$ on procède de bas en haut et on calcule : $BDD(A) = A \longrightarrow \mathbf{1}; \mathbf{0}$, $BDD(\top) = \mathbf{1}$, $BDD(\perp) = \mathbf{0}$, $BDD(F_1 \vee F_2) = \text{BDDor}(BDD(F_1), BDD(F_2))$, $BDD(\neg F_1) = \text{BDDneg}(BDD(F_1))$, $BDD(F_1 \wedge F_2) = \text{BDDand}(BDD(F_1), BDD(F_2))$, $BDD(F_1 \Rightarrow F_2) = \text{BDDor}(\text{BDDneg}(BDD(F_1)), BDD(F_2))$.

1. Montrer que $\models F$ si et seulement si $BDD(F)$ égale $\mathbf{1}$ (au sens de l'égalité des adresses). Comment peut-on tester la satisfiabilité de F à l'aide de $BDD(F)$?
2. Par l'exercice 6, toutes les opérations BDDor , BDDneg , BDDand sont en temps polynomial, de plus on peut tester la satisfiabilité de F en faisant un calcul simple sur $BDD(F)$, et certainement en temps polynomial. Or FORM-SAT est NP-complet. Peut-on en déduire $\mathbf{P} = \mathbf{NP}$? Sinon, où est l'erreur ?