

Formation à la Recherche

Nom :**Prénom :****Groupe :**

Questionnaire autour de l'article "Knowledge Compilation and Theory Approximation", de H.Kautz et B.Selman (JACM 1996).

Durée : 45mn. Document autorisé : l'article en question.

1. Indiquer brièvement ci-dessous quel est le problème général auquel s'attaquent les auteurs.

Il s'agit de concevoir des algorithmes d'inférence aussi efficaces que possible dans des langages de représentation aussi généraux que possible.

2. Quelle est la stratégie adoptée par les auteurs pour résoudre ce problème ?

Il s'agit "d'encadrer" (au sens de l'inclusion des modèles) un ensemble de formules par deux ensembles de formules pour lesquels on connaît des algorithmes d'inférence efficaces.

3. Sur quel exemple les auteurs se basent-ils pour développer leur méthode ?

Les auteurs se basent sur la logique propositionnelle.

4. Parmi les formules propositionnelles suivantes, indiquer celle(s) qui est (sont) en Forme Normale Conjonctive (CNF).

(a) $(a \wedge b \wedge \neg b) \vee (b \wedge \neg c) \vee (\neg a \wedge a \wedge \neg c)$

Pas CNF (mais DNF)

(b) $(a \vee b \vee \neg b) \wedge (b \vee \neg c) \wedge (\neg a \vee a \vee \neg c)$

CNF - pas Horn à cause de la première clause

(c) $((a \wedge \neg b \wedge d) \vee (b \wedge \neg c)) \wedge (\neg a \vee a \vee \neg c)$

Ni CNF - Ni DNF - le dernier composant est une tautologie

(d) $(\neg a \vee b \vee \neg c) \wedge (\neg b \vee c) \wedge (a \vee b)$

CNF - Pas Horn (à cause de la 3ème clause)

(e) $(\neg a \vee b) \wedge (\neg b \vee \neg a) \wedge (a \vee b)(a \vee \neg b)$

CNF - Horn

(f) $(\neg a \vee b \vee c) \wedge (d \vee \neg c) \wedge (\neg b \vee c \vee d)$

CNF - Pas Horn (la 1ère clause ne l'est pas)

5. Parmi les formules de la question précédente, indiquez celles qui sont satisfaisables et celles qui ne le sont pas (et prouvez vos assertions).

(a) $(a \wedge b \wedge \neg b) \vee (b \wedge \neg c) \vee (\neg a \wedge a \wedge \neg c)$

Satisfiable - ex : $I = [b, \neg c]$

(b) $(a \vee b \vee \neg b) \wedge (b \vee \neg c) \wedge (\neg a \vee a \vee \neg c)$

Satisfiable - ex : $I = [b]$ ou $I = [\neg c]$ - les clauses 1 et 3 sont des tautologies

(c) $((a \wedge \neg b \wedge d) \vee (b \wedge \neg c)) \wedge (\neg a \vee a \vee \neg c)$

Satisfiable - ex : $I = [a, \neg b, d]$ ou $I = [b, \neg c]$

(d) $(\neg a \vee b \vee \neg c) \wedge (\neg b \vee c) \wedge (a \vee b)$

Satisfiable - ex : $I = [b, c]$

(e) $(\neg a \vee b) \wedge (\neg b \vee \neg a) \wedge (a \vee b)(a \vee \neg b)$

Insatisfiable - quelle que soit la façon d'affecter a et b il y a toujours une des 4 clauses qui est falsifiée

(f) $(\neg a \vee b \vee c) \wedge (d \vee \neg c) \wedge (\neg b \vee c \vee d)$

Satisfiable - ex : $I = [c, d]$

6. Qu'a prouvé Cook en 1971 ?

Cook a démontré que le problème SAT est NP-Complet

7. Parmi les formules propositionnelles de la question 4, indiquer celle(s) qui est (sont) une (des) Théorie(s) de Horn.

Seule la formule $(\neg a \vee b) \wedge (\neg b \vee \neg a) \wedge (a \vee b)(a \vee \neg b)$ est une théorie de Horn

8. Donner un GLB de la dernière formule de la question 4.

Il y a 4 renforcements de Horn, fonction des littéraux positifs qui sont éliminés dans les clause qui ne sont pas de Horn :

- $(\neg a \vee b) \wedge (d \vee \neg c) \wedge (\neg b \vee c)$
- $(\neg a \vee b) \wedge (d \vee \neg c) \wedge (\neg b \vee d)$
- $(\neg a \vee c) \wedge (d \vee \neg c) \wedge (\neg b \vee c)$
- $(\neg a \vee c) \wedge (d \vee \neg c) \wedge (\neg b \vee d)$

9. Expliquez en quelques phrases l'algorithme présenté en figure 2, page 7.

Le principe est assez simple et consiste à déterminer si une formule est conséquence d'une théorie Σ en raisonnant sur ses approximations supérieures et inférieures.

Si la formule f est une conséquence Σ_{lub} alors elle est aussi une conséquence de Σ . Inversement, si f n'est pas une conséquence de Σ_{glb} , alors ce n'est pas une conséquence de Σ non plus. L'intérêt étant que ces deux tests peuvent s'exécuter en temps polynomial.

Si aucun des deux cas n'est vérifié... on ne peut se prononcer il faut utiliser une autre approche qu'en passant par les approximations pour déterminer si $\Sigma \models f$.

10. Que cherchent à montrer, respectivement, les sections 2.4 et 2.5 ? *La section 2.4 illustre les gains induits par cette approche en étudiant le raisonnement sur des théories 3-CNF connues comme particulièrement difficiles pour le problème SAT. Ils comparent de façon expérimentale les temps nécessaire pour répondre sans et avec les approximations, sur des requêtes aléatoires.*

La section 2.5 étudie la taille des approximations générées par les deux algorithmes d'approximation. Si les bornes inférieures sont toujours plus petites que la théorie initiale, les bornes supérieures peuvent dans certains cas être exponentiellement plus grande que la théorie initiale. Dans certains cas cette taille peut être réduite en introduisant des variables auxiliaires, mais ce n'est pas toujours possible. L'approche reste malgré tout intéressante car on n'est pas obligé de raisonner sur les meilleures approximations possibles.

11. Quel est le but de la section 3 ? *La section 3 montre dans quelle mesure les principes introduits ici pour la logique propositionnelle peuvent être étendus à des formalismes plus riches, en généralisant les notions de renforcement et d'affaiblissement afin de pouvoir se ramener à des approximations par des théories sur lesquelles le raisonnement est connu pour être facile (i.e. Polynomial).*

12. Quel est l'apport distinctif de cet article par rapport aux travaux antérieurs ?
- *Le fait de déplacer l'effort de calcul dans l'étape de compilation, afin de pouvoir répondre efficacement aux requêtes par la suite.*
 - *Le caractère anytime des algorithmes d'approximation*
 - *Les généralisations possible de l'approche*

13. Qu'est-ce que P ? Qu'est-ce que NP ? Qu'est-ce qu'un problème NP-complet ?

- (a) *P est la classe de complexité des problèmes pouvant être résolu en temps polynomial par rapport à la taille des données d'entrée, par une machine déterministe*
- (b) *NP est la classe de complexité des problèmes non-déterministes polynomiaux, i.e. pouvant être résolu en temps polynomial par rapport à la taille des données d'entrée, par une machine non déterministe*
- (c) *Un problème P est dit NP – Complet est un problème au moins aussi difficile que tous les problèmes de NP, i.e. tout problème P' de NP peut se réduire au problème P par une transformation dont le coût est polynomial.*