Intégration d'un raisonnement sur les symétries dans une procédure DPLL*

Chu Min Li et Bernard Jurkowiak
LaRIA, Université de Picardie Jules Verne
5, rue du Moulin Neuf
80000 Amiens (France)
email: `{cli,jurkowia}@laria.u-picardie.fr`

Paul W.Purdom Jr.
Computer Science Dept., Indiana University
101 Lindly Hall
Bloomingon, In. 47405 (USA)
email: `pwp@cs.indiana.edu`

1 Introduction

De nombreux problèmes structurés SAT possèdent des symétries véritablement intéressantes. La détection de ces symétries peut apporter une amélioration considérable pour la résolution dès lors qu'elle évite la résolution de sous-formules déductibles par application de symétries sur des sous-formules déjà produites.
La détection et le traitement de ces symétries ont fait l'objet de nombreux travaux au sein de la communauté SAT. Crawford, Ginsberg, Luks et Roy [3] suggèrent l'ajout d'axiomes dans la formule CNF, Benhamou et Sais [1] proposent de détecter les symétries syntaxiques dans une formule CNF et les exploitent pour réduire l'espace de recherche tandis que Brown, Finkelstein et Purdom [2] présentent l'intégration d'un détecteur appliqué à chaque branchement de la procédure DPLL en utilisant une technique issue de la théorie des groupes.
Notre approche consiste à mettre en place une architecture générique capable de briser les symétries dans une procédure DPLL. Comme dans [2], nous exploitons les symétries à chaque noeud de l'arbre de résolution. Cependant, notre raisonnement exploite uniquement et pleinement la logique propositionnelle.

2 Une approche générique

Soit $\mathcal{F}$ le codage CNF d'un problème $\mathcal{P}$. Nous introduisons une procédure de "brisure de symétrie" au sein de la procédure DPLL. La procédure DPLL "étendu" obtenu ainsi, travail à la fois sur $\mathcal{F}$ et sur la représentation sémantique de $\mathcal{P}$. De cette manière, elle réalise une interaction entre ces 2 représentations.
Ainsi les branchements successifs et les propagations de contraintes produisent une affectation partielle des variables booléennes correspondantes aux objets partielles de la

*Ce travail est financé par le CNRS (SUB/2001/0111/DR16)

représentation du problème $\mathcal{P}$. La procédure de détection de symétries consiste à anticiper sur les permutations possibles des affectations booléennes de $\mathcal{F}$ en tenant compte de la représentation sémantique de P.

3 Résultats expérimentaux

Nous illustrons notre approche avec 2 exemples. Soient $MatNL$ la multiplication de 2 matrices $N \times N$ en utilisant L produits et $PolNL$ la multiplication de 2 polynôme de degré-borne N avec L produits.
Ces 2 problèmes codés en SAT, permettent de déterminer s'il est possible de faire la multiplication de 2 matrices (resp. 2 polynômes) avec L produits. La symétrie à détecter se situe au niveau des L produits. En effet, il existe $L!$ permutations possibles et équivalentes des produits. Ainsi, la détection de symétrie consiste à déterminer un ordre sur ces produits autrement dit sur la représentation des objets du problème initial afin de briser tous travaux inutile par le solveur SAT employé. Plus concrètement, nous imposons $V_1 > V_2 > ... > V_L$ où V_l représente un produit l i.e. l'affectation booléenne des variables intervenant dans V_l.
Nous avons implémenté cette procédure dans EqSatz [4] et nous avons nommé la procédure obtenue "sEqSatz". Les expérimentations ont été effectuées sur un P4 à 1,8 GHz avec 256 Mo de RAM. " ?" signale que zChaff renvoie "Out of memory".

Pb	#var	#clause	sat	Satz	EqSatz	Relsat	zChaff	Walksat	sEqSatz
$Mat25$	588	1968	non	1511s	657s	19720s	12s	-	34s
$Mat26$	744	2464	non	>36000s	>36000s	>36000s	?	-	1462s
$Mat27$	900	2960	oui	>36000s	>36000s	>36000s	<1s	$\diamond$	15419s

TAB. 1 – *Temps d'exécutions de Satz, EqSatz, Relsat, zChaff, Walksat et sEqSatz*

Pb	#var	#clause	sat	Satz	EqSatz	Relsat	zChaff	Walksat	*sEqSatz*
$Pol23$	69	192	oui	<1s	<1s	<1s	<1s	-	<1s
$Pol35$	415	1350	non	<1s	1s	6s	<1s	-	<1s
$Pol36$	525	1692	oui	<1s	<1s	8s	<1s	1s	<1s
$Pol48$	1704	5760	non	>7200s	>7200s	>7200s	?	-	<1s
$Pol49$	1959	6592	oui	>7200s	>7200s	>7200s	<1s	$\diamond$	<1s
$Pol512$	5253	18000	non	>7200s	>7200s	>7200s	?	-	1298s
$Pol513$	5747	19650	oui	>7200s	>7200s	>7200s	?	$\diamond$	28s

TAB. 2 – *Temps d'exécutions de Satz, EqSatz, Relsat, zChaff, Walksat et sEqSatz*

Références

[1] B. Benhamou and L. Sais. Tractability through symmetries in prositional calculus. In *Journal of automated reasoning,*, pages 89–102, 1994.

[2] C.A. Brown, L. Finkelstein, and P.W. Purdom. Backtrack searching in the presence of symmetry. In *Applied algebra, algebraic algorithms and error correcting codes, 6th international conference*, pages 99–110, 1988.

[3] J.M. Crawford, M. Ginsberg, E. Luks, and A. Roy. Symmetry-Breaking Predicates for Search Problems. In *proceedings Fifth International Conference on Principles of Knowledge Representation and Reasoning*, 1996.

[4] C.M. Li. Integrating equivalency reasoning into davis-putnam procedure. In *proceeding of AAAI'2000*, pages 291–296, 2000.