Etude des symétries dans les modèles finis

Gilles Audemard — Belaid Benhamou

Laboratoire d'Informatique de Marseille
Centre de Mathématiques et d'Informatique
39, Rue Joliot Curie - 13453 Marseille cedex 13
{audemard,benhamou}@lim.univ-mrs.fr

RÉSUMÉ. *La génération de modèles et de contre modèles finis pour des théories de la logique du premier ordre est une notion complémentaire à la démonstration automatique. Les méthodes de génération de modèles finis exploitent certaines symétries pour élaguer l'espace de recherche. Les systèmes* FALCON, SEM *et* FMSET, *utilisent l'heuristique* LNH*(Least Number Heuristic) de Jian* ZHANG *qui consiste a conserver les symétries existant entre les individus des domaines de la formule de départ. Bien qu'intéressantes, ces symétries triviales disparaissent sitôt qu'on avance dans la recherche de la solution. Les individus utilisés dans l'interprétation partielle deviennent non symétriques au sens* LNH. *Or, d'autres symétries existent abondamment sur ces éléments. Nous étudions, dans ce papier un cadre plus général de la symétrie pour les modèles finis, nous montrons que la symétrie* LNH *est un cas particulier de ce cadre. Nous Donnons quelques résultats sur les symétries et proposons une méthode de détection dynamique de ces dernières. Nous expliquons ensuite comment utiliser ces symétries en les combinant avec celle de* LNH *pour améliorer l'efficacité des générateurs de modèles finis. Enfin, Une étude expérimentale et comparative de la méthode* SEM *avec et sans l'apport des symétries est faite sur des problèmes intéressants en mathématiques.*

ABSTRACT. *Finite modele search for first order logic theories is a complementary alternative to automated deduction. Systems like Falcon, SEM and FMSET use the LNH(Least Number Heuristic) heuristic to eliminate some trivial symmetries. Such symmetries are worthful, but their exploitation is limited to the first levels of the model search tree, since they desappear as soon as the first cells have been instanciated. We study in this paper a more general notion of symmetry than LNH and provide some exploitations which we combine with it to optimize the symmetry use and increase the model search efficiency. The method SEM with and without symmetry is experimented on several interesting mathematic problems to show the advantage of symmetry.*

MOTS-CLÉS : *Modèles Finis, Symétries*

KEYWORDS: *Finite models, Symmetries*

1. Introduction

La recherche de modèles finis pour des théories exprimées en logique du premier ordre est un cas particulier de modèles de satisfaction de contraintes. Un modèle fini d'une théorie de la logique du premier ordre est une interprétation des variables, des symboles fonctionnels et des prédicats sur les domaines finis d'individus. Un générateur de modèles finis est un moyen automatique de recherche de ces interprétations. L'existence de modèles confirme la cohérence de la théorie. L'existence de contre modèles peut servir pour réfuter une conjecture. Dans ce sens, la génération de modèles finis joue un rôle complémentaire en démonstration automatique.

La génération de modèles finis d'une théorie, ne peut être efficace sans l'élimination des symétries et la considération des interprétations isomorphes (symétriques). Il existe plusieurs approches d'élimination des symétries. Une façon de raisonner est de poser des contraintes additionnelles exprimant les symétries dans la formalisation du problème. Cette approche statique a été utilisée par CRAWFORD et al. dans [CRA 96] pour la logique propositionnelle et par FUJITA et al.[FUJ 93] pour la génération de modèles finis pour les problèmes de quasigroupes. L'inconvénient de cette approche est qu'elle n'utilise que les symétries de départ. Pour y remédier d'autres approches d'utilisation dynamique des symétries sont apparues.

Notamment, La méthode FMC [PEL 98] énumère des interprétations complètes et utilise les symétries pour éviter certaines interprétations redondantes. Des générateurs de modèles finis comme SEM [ZHA 95], FMSET [BEN 99] utilisent l'approche LNH qui exploitent certaines symétries existant entre les individus des domaines pendant l'énumération. Pour exprimer les problèmes, ces méthodes utilisent la forme clausale de la logique du premier ordre où toutes les variables sont quantifiées universellement. De ce fait, au départ tous les individus des domaines des variables sont symétriques. Si à un moment donné de la résolution les premiers individus $\{0 \ldots mdn\}$[1] d'un domaine $D_n = \{0 \ldots n-1\}$[2] (avec $0 \leq mdn \leq n-1$), ont été utilisés dans la construction du modèle alors, seuls les individus de la partie $\{mdn+1 \ldots n-1\}$ resteront symétriques au sens LNH. Pour garder longtemps cette symétrie l'heuristique LNH choisit en priorité les individus qui ont été déjà utilisés pour instancier un nouveau terme.

Bien qu'intéressantes, ces symétries de départ (partie $\{mdn \ldots n-1\}$) disparaissent en avançant dans la recherche de la solution. Du fait des propagations, on arrive rapidement à utiliser tous les individus du domaine et l'ensemble des individus symétriques au sens LNH se vide. Plus précisément, La partie symétrique $\{mdn + 1 \ldots n-1\}$ diminue et devient vide quand la frontière mdn atteint la valeur $n-1$. Inversement la partie non symétrique $\{0 \ldots mdn\}$ augmente et finit par contenir tous les éléments de D_n. Ainsi les symétries LNH de la partie $\{mdn + 1 \ldots n-1\}$ dispa-

1. Le nombre mdn (maximum designated number) est la frontière qui sépare les individus de la partie symétrique de ceux de la partie non symétrique du domaine
2. Sans perdre de généralité, un domaine à n individus peut être considéré comme l'ensemble des n premiers entiers

raissent au fur et à mesure, mais il reste beaucoup d'autres symétries non exploitées dans la partie $\{0 \ldots mdn\}$.

Dans ce travail, nous nous sommes intéressé à la détection et à l'exploitation des symétries existant entre les individus de la partie $\{0 \ldots mdn\}$. Ces dernières seront combinées avec les symétries LNH de la partie $\{mdn \ldots n-1\}$ pour assurer une meilleure utilisation.

Cet article est organisé comme suit : Dans la section 2 nous donnons une brève description des théories multi types que nous utilisons pour représenter la connaissance. Dans la section 3 nous étudions le principe de symétrie pour les modèles finis, nous montrons que la symétrie LNH est un cas particulier de notre cadre d'étude. Dans la quatrième section nous montrons comment exploiter ces symétries dans la méthode génération de modèles finis SEM. La section 5 présente une étude expérimentale et comparative de la méthode SEM avec et sans l'avantage des symétries sur des problèmes mathématiques intéressants.

2. Les théories multi types

Dans cette section, nous définissons le langage d'entrée pour représenter les problèmes. Nous utilisons des théories multi types de la logique du premier ordre avec le prédicat d'égalité. Les formules sont écrites sous forme clausale classique skolémisée où toutes les variables sont quantifiées universellement. Formellement, une théorie multi types T est un triplet (S, F, C), où S est l'ensemble des types, F l'ensemble des symboles fonctionnels et C l'ensemble des clauses. Nous notons l'ensemble des instances terminales de C par C^t. Sans perte de généralité, les individus d'un type $s \in S$ de cardinalité n sont considérés comme étant les premiers entiers naturels $\{0 \ldots n-1\}$. Un symbole fonctionnel $f \in F$ d'arité k est spécifié par $f : s_1 \times s_2 \ldots \times s_k \mapsto s$, où $s_1 \ldots s_k, s \in S$. Les *termes* sont construits récursivement à partir des symboles fonctionnels, des variables et des constantes. Les prédicats sont considérés comme des fonctions de type booléen. Les problèmes sont alors représentés par des ensembles de clauses C de la logique du premier ordre. Une clause est une disjonction de litéraux. Un litéral est soit une équation $t_1 = t_2$ ou $t_1 \neq t_2$, où t_1 et t_2 sont des termes, soit un terme de type booléen t. Un *terme terminal* est de la forme $f(e_1, \ldots, e_k)$ où tous les éléments e_i appartiennent à des types $s_i \in S$ et $f \in F$. L'équation $f(e_1, \ldots, e_k) = e$, définit l'instanciation du terme $f(e_1, \ldots, e_k)$ à $e \in s$. Une interprétation de l'ensemble C^t des instances terminales de C consiste à donner à chaque terme terminal une valeur dans son domaine. Une interprétation peut être représentée par la table d'opération de chaque symbole fonctionnel. Une interprétation qui satisfait toutes les clauses de C^t est un modèle fini. La génération de tels modèles peut être vu comme la résolution de CSP discrets à domaines finis particuliers où les variables sont les termes terminaux et les domaines sont les individus des types associés. Si C^t est un ensemble de clauses terminales et I l'instanciation courante, on notera alors C^t_I l'ensemble de clause C^t simplifié par I.

Exemple 1
Soit la théorie suivante $T = (\{D_n\}, \{h\}, C)$ ou $D_n = \{0...n-1\}$ et C contient les deux clauses suivantes :
- $\forall x, h(x,x) = x$
- $\forall x \forall y, h(h(x,y),x) = y$

Lorsque $n = 4$ cette théorie possède plusieurs modèles dont un que nous nommons I est donné par la table de la fonction h suivante :

h	0	1	2	3
0	0	2	3	1
1	3	1	0	2
2	1	3	2	0
3	2	0	1	3

Ceci correspond aux affectations : $h(0,0) = 0$, $h(0,1) = 2$, etc...

3. Permutations et Symétries

Tout d'abord nous rappelons quelques notions de base sur les permutations, ensuite nous étudions la notion de symétrie sur les individus des domaines pour les modèles finis.

Définition 1 *Soit E un ensemble fini, une permutation σ de E est une bijection de E dans E. On note $Perm(E)$ l'ensemble des permutations de E.*

Le couple $(Perm(E), \circ)$ forme le groupe des permutations de E. La composition de deux permutations et l'inverse d'une permutation sont des permutations. Si σ est une permutation, alors σ^k exprime k compositions de σ. L'ordre d'une permutation σ est le plus petit entier n tel que $\sigma^n = Id_E$. L'ensemble des remplacements $\sigma(a_1) = a_2$, $\sigma(a_2) = a_3 \ldots \sigma(a_n) = a_1$, forme un cycle de permutation sur les individus $\{a_1, \ldots a_n\}$ de E. On note un tel cycle par le tuple $(a_1, a_2, \ldots, a_n)$, son ordre est n. Toute permutation peut être vue comme une composition de cycles.

Définition 2 *Une permutation σ sur l'ensemble des types $S = \{s_1, \ldots, s_n\}$ est définie par le tuple de permutation $\sigma = (\sigma_1, \ldots, \sigma_n)$, où chaque $\sigma_i \in Perm(s_i)$.*

Définition 3 *Si le symbole fonctionnel $f \in F$ est spécifié par $f : s_{i_1} \times \ldots \times s_{i_k} \mapsto s_j$ et que $f(e_{i_1}, \ldots, e_{i_k})$ est un terme terminal alors $\sigma(f(e_{i_1}, \ldots, e_{i_k})) = f(\sigma_{i_1}(e_{i_1}), \ldots \sigma_{i_k}(e_{i_k}))$.*

Ainsi, la permutation σ peut se généraliser aux termes, aux litéraux et à l'ensemble des clauses terminales C^t de la théorie T. Nous définissons maintenant la symétrie pour les modèles finis.

Définition 4 *Soient $T = (S, F, C)$ une théorie, I l'interprétation courante et σ une permutation définie sur l'ensemble des types S. La permutation σ est une symétrie de $T \cup I$, si $\sigma(I) = I$ et $\sigma(C^t_I) = C^t_I$.*

La définition 4 donne de nouvelles conditions de la symétrie comparativement au cas propositionnel [BEN 94]. Pour vérifier les conditions de symétrie en un noeud donné de l'arbre de recherche, la permutation σ doit laisser invariant l'ensemble de clauses représentant l'interprétation partielle I et celles du système C^t définie en ce noeud. L'interprétation courante est en effet une structure mathématique qui regroupe des équations (clauses unitaires) de la forme : $f(e_1, \ldots, e_k) = e_{k+1}$ où $e_i \in s_i \in S$ qui sont en général sensibles à toute permutation des individus e_i. Ceci est différent du cas propositionnel où l'interprétation partielle n'est pas concernée par la permutation.

Définition 5 *Soit $T = (S, F, C)$ une théorie. Deux individus e_1 et e_2 d'un type s sont symétriques s'il existe une symétrie σ de T telle que $\sigma(e_1) = e_2$.*

REMARQUE. — Les éléments d'un cycle de symétrie sont symétriques deux à deux.

Définition 6 *Soit $T = (S, F, C)$ une théorie. Deux interprétations I et J sont symétriques si il existe une symétrie σ de T telle que $\sigma(I) = J$.*

Il est évident que si I est un modèle de la théorie T alors J le sera aussi. Dans ce cas I et J forment deux modèles symétriques de T. Nous nous intéressons maintenant au problème de recherche des symétries.

3.1. *Détection des symétries*

Les permutations vérifiant les conditions de la symétrie (définition 4) sont construites à partir de compositions de cycles de permutations. L'algorithme de détection des symétries comporte deux étapes distinctes : La première consiste à partitionner chaque type $s \in S$ de la théorie en classes d'équivalence où les individus doivent vérifier certaines conditions nécessaires de la symétrie que nous discutons dans la proposition 1. Deux éléments peuvent être symétriques si et seulement si ils appartiennent à la même classe d'équivalence.

3.1.1. *Les conditions nécessaires*

Pour être symétriques, les individus des différents types doivent satisfaire les conditions nécessaires de la symétrie.

Définition 7 *Si I est une interprétation d'une théorie $T = (S, F, C)$ et $f \in I \cap F$ une fonction d'arité n, alors $I_{f_i}(a)$ est le nombre d'occurences dans I de l'individu a comme i^{eme} argument de la fonction f. De même, $I_{f_{val}}(a)$ est le nombre d'occurences dans I de l'élément a comme valeur de la fonction f.*

Dans l'exemple 1, nous avons I_{h_1}(0)=4, I_{h_2}(0)=4, et $I_{h_{val}}$(0)=4. En fait, dans cet exemple nous avons I_{h_i}(a)=4, $\forall i \in$ {1,2}, $\forall a \in$ {0,1,2,3} et $I_{h_{val}}$(a)=4, $\forall a \in$ {0,1,2,3}.

Proposition 1 (Conditions Nécessaires) *Soient une théorie $T = (S, F, C)$ ayant I comme interprétation partielle courante, a et b deux éléments d'un type $s \in S$. Pour que a et b soient symétriques dans $I \cup T$ (appartiennent à un même cycle de symétrie), ils doivent satisfaire les conditions suivantes :*
- $\forall f \in I \cap F, \quad \forall i \in \{1, \ldots, \textit{arité}(f)\}, \quad I_{f_i}(a) = I_{f_i}(b)$
- $\forall f \in I \cap F, \quad I_{f_{val}}(a) = I_{f_{val}}(b)$.

Preuve Montrons la deuxième condition, la preuve est identique pour la première. Soient I l'interprétation courante de la théorie T, σ une symétrie de $I \cup T$, et deux individus a et b d'un type $s \in S$, telle que $\sigma(a) = b$. Supposons qu'il existe une fonction $f \in I \cap F$ telle que $= I_{f_{val}}(a) \neq I_{f_{val}}(b)$. On aura donc, $\sigma(I)_{f_{val}}(b) \neq I_{f_{val}}(b)$, car le nombre d'occurences de b dans I comme valeur de f est différent de celui de a. Ceci qui implique que $\sigma(I) \neq I$ d'où la contradiction car σ est une symétrie de $I \cup T$.

Ces conditions nécessaires forment une étape très importante de l'algorithme de détection des symétries. Elles permettent de réduire considérablement l'espace de recherche des permutations. Elles partitionnent les individus en classes d'équivalence formant ainsi les candidats potentiels sur lesquels s'effectue la recherche de la symétrie.

3.1.2. *Calcul de la symétrie : vérification de la condition suffisante*

La deuxième étape est celle qui construit la permutation (la symétrie) en utilisant les classes d'équivalence préalablement définies.

Une symétrie en calcul propositionnel [BEN 94] est une permutation qui doit laisser invariant l'ensemble de clauses courant. Cette condition rend un peu lourd la détection quand le cardinal de l'ensemble de clauses est important. Dans les modèles finis nous simplifions le calcul. En effet, si l'ensemble d'axiomes de départ ne contient aucune constante, il suffit de chercher une permutation laissant invariant l'interprétation courante I. Nous montrons que dans ce cas l'invariance de l'ensemble de clauses terminales C^t_I correspondant est garantie. Ceci nous dispense donc d'une lourde tâche qui est la vérification de son invariance. Formellement, on a la proposition suivante :

Proposition 2 *Soient une théorie $T = (S, F, C)$, I l'interprétation partielle courante et σ une permutation de $Perm(S)$. Si C ne contient aucun individu e d'un type $s \in S$ et si $\sigma(I) = I$ alors $\sigma(C^t_I) = C^t_I$.*

Preuve Si C ne contient aucun élément $e \in s$ alors, toute permutation $\sigma \in Perm(S)$ vérifie $\sigma(C^t) = C^t$. Donc $\sigma(C^t_I) = \sigma(C^t)_{\sigma(I)} = C^t_{\sigma(I)} = C^t_I$.

Ainsi, nous obtenons un résultat important pour la détection des symétries. La proposition 2 simplifie la condition suffisante de la symétrie de la définition 4. En conséquence l'efficacité de détection sera meilleure.

REMARQUE. — Dans le cas où des individus apparaissent dans un sous-ensemble de clauses $C' \subset C$ de la théorie $T = (S, F, C)$, la permutation σ doit laisser invariant ${C'}^t$ en plus de l'invariance de I, c'est à dire $\sigma(I) = I$ et $\sigma({C'}^t) = {C'}^t$.

Proposition 3 *Soient une théorie $T = (S, F, C)$, I l'interprétation partielle courante et σ une permutation au sens* LNH*. Si C ne contient aucune constante alors $\sigma(I) = I$.*

Preuve La permutation LNH σ forme un cycle complet des individus de la partie $\{mdn + 1, \ldots, n - 1\}$ du domaine. L'interprétation I est formée d'individus de la partie $\{0, \ldots, mdn\}$ et donc est indépendante de la permutation σ, et $\sigma(I) = I$.

La symétrie LNH est donc un cas particulier de notre cadre d'étude. D'autres extensions de l'heuristique LNH se reconnaissent aussi dans ce cadre. Par exemple dans [AUD 00], les auteurs proposent une extension de LNH en commencant la génération des modèles par l'assignation d'une fonction unaire et bijective.

La figure 1 donne une description du schèma algorithmique de la recherche des symétries. C'est une procédure de type backtrack qui dans le pire des cas peut avoir une complexité exponentielle. Mais dans la pratique, les symétries, quand elles existent, sont calculées avec un petit nombre de backtracks et n'affectent pas trop le temps de recherche des modèles. En théorie, la complexité du problème de recherche des symétries est équivalente à celle de recherche d'isomorphismes de graphe (cf. CRAWFORD [CRA 92]) qui demeure un problème ouvert.

Dans la figure 1, $Cl(e, s)$ désigne la classe d'équivalence de l'individu e du type $s \in S$ définie par les conditions nécessaires de la symétrie (proposition 1). La fonction *Partitionner(s,I)* crée les classes d'équivalence du type $s \in S$ correspondant aux conditions nécessaires de la symétrie. La fonction *Mdn(s)* retourne la valeur du MDN associé au type $s \in S$. Cette valeur definie la frontière supérieure du sous-domaine $\{0 \ldots mdn\}$ de s dans lequel nous cherchons les symétries. Les deux fonctions précédentes sont très élémentaires, leurs codes respectifs ne sont pas donnés dans cet article.

4. Exploitation des symétries

Dans cette section, nous donnons des propriétés permettant d'utiliser les symétries détectées afin d'améliorer l'efficacité des générateurs de modèles finis.

Proposition 4 *Soient $T = (S, F, C)$ une théorie, I l'interprétation courante invariante par $\sigma \in Perm(S)$ et t un terme terminal non instancié. Si l'affectation courante dans I est $t = e$, alors les deux extensions $I \cup \{t = e\}$ et $I \cup \{\sigma(t = e)\}$ de I, sont symétriques.*

Algorithme 1 La procédure de Recherche de symétries

Fonction Sym(I : Interprétation ; Var σ : permutation)
Sortie : Vrai si la permutation σ construite vérifie $\sigma(I) = I$
Faux sinon
Fonction Construction(Var σ : permutation)
Début
Si $\sigma(I) = I$ **Alors Retourner** Vrai
Choisir un type s et $e \in s$ tel que $0 \leq e \leq Mdn(s)$ et
Pour Chaque $a \in Cl(e, s)$ **Faire**
$\sigma(e) = a$
$Cl(e, s) = Cl(e, s) - a$
Si Construction(σ)=Vrai **alors Retourner** Vrai
Retourner Faux
Fin
Début
I=interpretation courante
Pour Chaque type s faire Partitionner(s, I).
Retourner Construction(σ)
Fin

Preuve Comme I est invariante par σ, alors $\sigma(C_I^t) = C_I^t$ (proposition 2). Les interprétations $I \cup \{t = e\}$ et $\sigma(I \cup \{t = e\})$ sont symétriques (définition 6). Or $\sigma(I \cup \{t = e\}) = \sigma(I) \cup \{\sigma(t = e)\} = I \cup \{\sigma(t = e)\}$. Ce qui implique que les deux extensions $I \cup \{t = e\}$ et $I \cup \{\sigma(t = e)\}$ sont symétriques.

Ainsi, les instanciations $t = e$ et $\sigma(t = e)$ déterminent deux extensions symétriques de l'interprétation I. Il suffit d'en considérer une seule, si l'on ne veut calculer que les modèles "non isomorphes". Comme on s'intéresse beaucoup plus au problème de la consistance de la théorie T, il est important de savoir le lien de cohérence entre les deux extensions. C'est ce que nous exprimons par la proposition suivante :

Proposition 5 *Si l'interprétation partielle I est cohérente dans $T = (S, F, C)$ alors, [l'extension $I \cup \{t = e\}$ est cohérente ssi $I \cup \{\sigma(t = e)\}$ est cohérente]*

Preuve C'est une conséquence de la proposition 4 qui garantie que les deux extensions sont symétriques et du fait que les interprétations symétriques sont équivalentes pour la cohérence.

La proposition 5 exprime un résultat important que nous exploitons pour élaguer l'arbre de recherche d'un modèle fini. Le gain se traduit par la coupure que nous réalisons quand l'instanciation $t = e$ mène à une incohérence. En effet, si l'instanciation

$t = e$ est inconsistante dans l'interprétation partielle I alors, l'instanciation symétrique $\sigma(t = e)$ le sera aussi. Grâce à cette information, le parcours dans l'arbre de recherche de la branche inutile correspondant à $\sigma(t = e)$, où $\sigma(t) = \sigma(e)$ est évité. C'est la coupure des symétries.

Nous montrons dans ce qui va suivre qu'une symétrie σ d'ordre k peut réaliser k-1 coupures de branches dans l'arbre de recherche d'un modèle fini. Il est alors important de détecter des symétrie d'un grand ordre pour assurer une meilleure utilisation.

Proposition 6 *Soient une théorie $T = (S, F, C)$, $\sigma \in Perm(S)$ et I l'interprétation courante. Si $\sigma(I) = I$, alors $\forall k \in \mathbb{Z}, \sigma^k(I) = I$.*

Preuve Par induction sur $k \in \mathbb{Z}$. La propriété est vraie pour k=0, car σ^0(I)=id(I)=I. Supposons que la propriété est vraie a l'ordre k, montrons que cela reste vrai pour $k+1$. On a $\sigma^{k+1}(I) = \sigma(\sigma^k(I)) = \sigma(I)$(hypothèse de récurrence). Donc $\sigma^{k+1}(I) = \sigma(I) = I$.

REMARQUE. — Si dans une théorie T=(S,F,C), l'interprétation I est invariante par σ^k, $\forall k \in \mathbb{Z}$ alors $\sigma^k(C^t_I) = C^t_I$, $\forall k \in \mathbb{Z}$.

Ainsi, nous généralisons le résultat de la proposition 5 à une puissance quelconque de σ. On a la proposition suivante :

Proposition 7 *Si l'interprétation partielle I est cohérente dans $T = (S, F, C)$ alors [l'extension $I \cup \{t = e\}$ est cohérente ssi $I \cup \{\sigma^k(t = e)\}, \forall k \in \mathbb{Z}$ est cohérente]*

Preuve C'est une conséquence des deux propositions 5 et 6. En effet, $\sigma^k(I) = I, \forall k \in \mathbb{Z}$ (proposition 6). Ce qui implique que les extensions $I \cup \{t = e\}$ et $I \cup \{\sigma^k(t = e)\}$, $\forall k \in \mathbb{Z}$ sont symétriques et donc équivalentes du point de vue cohérence (proposition 5).

La proposition 7 permet une meilleure utilisation de la symétrie σ. En effet, $k - 1$ coupures sont réalisées dans l'arbre de recherche quand $t = e$ est inconsistante. Ces coupures correspondent aux instanciations inconsistantes $\sigma^i(t = e)$ pour $i \in \{1, \ldots, k-1\}$ que nous évitons grâce aux symétries.

Plusieurs symétries peuvent exister pour une théorie donnée. Mais pour des raisons d'efficacité, nous en détectons et en exploitons une seule à chaque noeud de l'arbre de recherche. Cette dernière est combinée avec la symétrie triviale LNH pour avoir un meilleur gain.

Considérons une théorie ne contenant qu'un seul type dont les individus sont dans $D = \{0, \ldots, n-1\}$. Si à un noeud donné, l'ensemble des individus utilisés par le modèle partiel I est dans l'ensemble $\{0 \ldots mdn\}$. Si en plus, les individus du terme terminal t de l'instanciation courante sont tous dans $\{0 \ldots mdn\}$, alors les interprétations $I \cup \{t = m+1\} \ldots I \cup \{t = n-1\}$ sont trivialement symétriques. En plus de cette symétrie fournie par l'heuristique LNH, nous détectons moyennant la méthode décrite dans la figure 1 une symétrie σ sur les individus de la partie $\{0 \ldots mdn\}$. La symétrie σ fournit en général $k-1$ extensions équivalentes de l'interprétation I si k est son ordre. Une seule de ces dernières est retenue pour tester la cohérence de la théorie. Les deux symétries LNH et σ sont définies sur deux parties indépendantes du domaine, leurs combinaisons est faite de façon très naturelle. L 'association de ces deux symétries permet de réaliser au total $n - mdn - 1 + k$ coupures de branches en chaque noeud de l'arbre de recherche.

5. Expérimentations

Nous avons testé l'apport des symétries pour le générateur de modèles finis SEM [ZHA 95]. Dans un premier temps nous allons étudier le comportement de la recherche des symétries sur différents problèmes que nous présentons dans la section 5.1. Ensuite, Nous comparons SEM avec et sans l'avantage des symétries. Tous les modèles sont générés afin de savoir le nombre de modèles isomorphes supprimés par les symétries. Les temps de calcul que nous donnons sont en secondes et sont obtenus sur un K6II cadencé à 400MHZ avec 128MO de RAM et fonctionnant sous Linux 2.2. Le symbole '-' indique que le problème n'est pas résolu en temps raisonnable (2 heures).

5.1. *Les Problèmes*

Les problèmes sur lesquels nous avons travaillé sont des problèmes mathématiques décrits par J. ZHANG dans [ZHA 96]. Tous ces problèmes ne contiennent qu'un seul type. Le symbole AG exprime un groupe abélien, NG un groupe non commutatif, GRP un groupe non commutatif satisfaisant l'axiome $(xy)^4 = x^4y^4$, RU un anneau unitaire, RNB un anneau non booléen qui satisfait $x^7 = x$ et RNA un anneau auquel on rajoute un contre exemple d'associativité. Nous avons listé les axiomes des problèmes AG et RU dans le tableau 1. Les autres problèmes sont générés par l'ajout de différents axiomes.

5.2. *Comportememnt des symétries dans* SEM

Le tableau 2 montre le comportement des symétries sur 3 problèmes différents. En ce qui concerne les groupes abéliens, on remarque que 82 % des appels à la recherche des symétries fournissent un cycle de symétrie (rapport des colonnes *trouvés* sur *Appels* du tableau 2). Ceci permet de supprimer un grand nombre d'interprétations isomorphes (rubrique *supprimés*). Dans les problèmes AG la détection prend un peu plus de temps que pour les autres problèmes car les conditions nécessaires sont toujours vérifiés et ne réduisent pas l'espace de recherche de la symétrie.

Groupe Abélien	Anneau Unitaire
	$s(x,0)=0$
$h(x,0)=x$	$s(x,y)=s(y,x)$
$h(0,x)=0$	$s(x,m(x))=0$
$h(x,g(x))=x$	$s(x,s(y,z))=s(s(x,y),z)$
$h(g(x),x)=x$	$p(x,1)=x$
$h(h(x,y),z)=h(x,h(y,z))$	$p(x,y)=p(y,x)$
$h(x,y)=h(y,x)$	$s(p(x,y),p(x,z))=p(x,s(y,z))$
	$s(p(x,z),p(y,z))=p(s(x,y),z)$

TAB. 1. *Problèmes* AG *et* RU

Dans les problèmes de groupes non abéliens, l'ajout de la clause $f(1,2) \neq f(2,1)$ exprimant la non symétrie, induit une contrainte sur les éléments 1 et 2. Ces derniers ne peuvent être symétriques qu'entre eux (cf remarque de la section 3.1.2). Ceci renforce le pouvoir des conditions nécessaires et permet de réduire le temps de recherche des symétries.

Pour le problème RNA, codant les anneaux non associatifs, on rajoute la clause $p(p(2,3),4) \neq p(2,p(3,4))$ qui exprime la propriété de non associativité. Cette clause ajoute des contraintes de conditions nécessaires sur les éléments {0,1,2,3,4} et dans la détection des symétries. Ceci raffine énormément la partition du domaine et donc réduit l'espace de recherche des symétries. On peut remarquer que l'apport des symétries croit en rapport avec la croissance de la taille du domaine, donnant ainsi espoir de résoudre des instances de grandes tailles.

Pb	Taille	Temps	Temps Sym	Noeuds	Appels	Trouvés	Supprimés
AG	24	68	55	89 213	12 033	9803	118698
	25	79	59	95 635	13 009	10737	133088
NAG	20	171	36	338 669	29388	25921	321123
	21	206	37	357 095	32480	27177	344578
RNA	8	5	0.4	51 491	7872	0	0
	9	10	0.8	81928	11 243	0	0
	10	62	5.3	252720	41 163	18 902	93184

TAB. 2. *Comportement de l'algorithme de recherche des symétries.*

5.3. *Comparaison de* SEM *avec et sans symétrie*

Les tableaux 3 et 4 montrent les résultats expérimentaux de SEM avec et sans symétries obtenus par notre première implémentation. On remarque que des résultats satisfaisants sont obtenus sur les problèmes NAG, GRP, RU et RNB. sur ces problèmes on génère des modèles avec un ordre de grandeur plus important en utilisant la détection des symétries. Pour les problèmes RNA, des symétries ont été détectées mais leurs utilisation s'avère peu efficace car souvent une inconsistance apparaît avant les coupures de symétries. Une solution envisageable (heuristique) est de choisir le prochain terme terminal à instancier en fonction des symétries calculées, cela est fait dans [MES 99]. Ces résultats encourageants montrent que l'heuristique LNH seule est insuffisante pour supprimer les nombreuses symétries existant dans les théories multi types.

		SEM avec symétries			SEM sans symétries		
Pb	Taille	Nb Modèles	Temps	Noeuds	Nb Modèles	Temps	Noeuds
AG	28	35	176	148 089	162	328	642 103
	30	64	277	189 406	262	559	870 097
	32	548	864	350 502	2 295	940	2 037 525
NAG	22	1	193	339 105	3	413	849 128
	24	357	1 062	1 251 861	1 130	2 636	3 526 828
	26	2	1 299	1 432 737	3	3 260	4 252 561
	28	14	2 220	2 065 498	51	6 934	8359103
	30	51	3 801	2 847 643	-	-	-
	31	0	4 296	2 876 674	-	-	-
GRP	24	24	613	647 591	48	2 018	2 419 623
	26	0	675	674 359	0	2 322	2 536 477
	28	0	839	712 491	0	2 794	2 635 461
	30	0	1 084	758 269	0	3 443	2 745 475
	32	654	3 133	1 866 914	-	-	-
	33	0	3 376	1 907 738	-	-	-

TAB. 3. *Groupes - Comparaison.*

6. Conclusion

Nous avons montré quelques résultats sur les symétries pour les modèles finis des théories du premier ordre. L'approche LNH s'avère un cas particulier de ces symétries. Un algorithme de détection de symétrie est donné. L'approche LNH est combinée avec les symétries détectées et exploitées dans la méthode SEM. Les premiers résultats expérimentaux sont encourageants. Plusieurs voies pour améliorer l'implantation sont possibles. Raffiner les conditions nécessaires pour réduire l'espace de recherche des symétries, trouver d'autres symétries qui ne nécessitent pas de phase de détection. Enfin étudier la structure des problèmes car des symétries particulières peuvent apparaître. Ce qui permet alors de les combiner avec les symétries détectées dynamiquement, comme nous le faisons avec celles de LNH.

		SEM avec symétries			SEM sans symétries		
Pb	Taille	Nb Modèles	Temps	Noeuds	Nb Modèles	Temps	Noeuds
RU	14	1	11	9 184	1	21	25 421
	16	335	49	49 914	1 745	162	51 327
	18	4	172	100 523	19	848	218 955
	20	4	659	217 502	21	5977	2 343 901
	21	1	825	237 697	-	-	-
	23	1	1 373	261 957	-	-	-
	25	9	3 565	603 360	-	-	-
	27	80	4 566	656 606	-	-	-
	28	4	5 843	723 818	-	-	-
RNA	10	0	62	252 720	0	55	252 720
	12	0	231	504 258	0	209	504 258
	14	0	432	595 340	0	592	595 340
	15	0	621	646 421	0	592	646 421
RNB	36	0	586	208 129	0	3 832	1 916 142
	38	0	733	208 129	0	4 872	1 916 142
	40	0	917	208 129	0	6 178	1 916 142
	41	0	1 017	208 129	0	6 871	1 916 142
	42	0	1 106	208 129	-	-	-
	43	0	1 231	208 129	-	-	-
	44	0	1 360	208 129	-	-	-
	46	0	1 634	208 129	-	-	-

TAB. 4. *Anneaux - Comparaison.*

7. Bibliographie

[AUD 00] AUDEMARD G., HENOCQUE L., «On the generation of non isomorphic finite groups», *proceedings of JNPC2000 - Marseille*, June 2000.

[BEN 94] BENHAMOU B., SAIS L., «Tractability through Symmetries in Propositional Calculus», *Journal of Automated Reasoning*, vol. 12, 1994, p. 89-102.

[BEN 99] BENHAMOU B., HENOCQUE L., «A Hybrid Method for finite Model Search in Equational theories.», *Fundamenta Informaticae*, vol. 39(1-2), 1999, p. 21-38.

[CRA 92] CRAWFORD J., «A theorical Analisys of Reasoning by Symmetry in First Order Logic», *Workshop on Tarctable Reasonning,AAI92*, , 1992.

[CRA 96] CRAWFORD J., GINSBERG M. L., LUCK E., ROY A., «Symmetry-Breaking Predicates for Search Problems», AIELLO L. C., DOYLE J., SHAPIRO S., Eds., *KR'96 : Principles of Knowledge Representation and Reasoning*, p. 148–159, Morgan Kaufmann, San Francisco, California, 1996.

[FUJ 93] FUJITA M., SLANEY J., BENNETT F., «Automatic Generation of Some Results in Finite Algebra», *Proceedings of International Join Conference on Artificial Intelligence*, 1993.

[MES 99] MESEGUER P., TORRAS C., «Solving Strategies for Highly Symmetric CSPs», *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.

[PEL 98] PELTIER N., «A new method for automated finite model building exploiting failures and symmetries», *Journal of Logic and Computation*, vol. 8, 1998, p. 511–543.

[ZHA 95] ZHANG J., ZHANG H., «SEM : a System for Enumerating Models», MELLISH C. S., Ed., *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995, p. 298–303.

[ZHA 96] ZHANG J., «Constructing Finite Algebras with FALCON», *Journal of Automated Reasoning*, vol. 17, 1996, p. 1–22.