Exploitation des symétries dans les formules booléennes quantifiées

Gilles Audemard, Said Jabbour, Lakhdar Saïs

Exploitation des symétries dans les formules booléennes quantifiées

Gilles Audemard, Said Jabbour et Lakhdar Saïs

CRIL CNRS – Université d'Artois
rue Jean Souvraz SP-18 F-62307 Lens Cedex France

{audemard,jabbour,sais}@cril.univ-artois.fr

Abstract

De nombreuses tâches et problèmes combinatoires exhibent des symétries. La résolution de tels problèmes conduit à répéter inlassablement l'étude de situations ou de sous-problèmes équivalents. Depuis plusieurs années, l'exploitation des symétries a permis une réduction significative de l'espace de recherche et la résolution de problèmes ouverts jusqu'alors. Ce paradigme important a été étudié de manière extensive dans de nombreux domaines, comme les problèmes de satisfaction de contraintes (CSP) ou la satisfiabilité de formules booléennes (SAT). L'approche consistant à rajouter des contraintes (symmetry breaking predicates en anglais) est l'une des techniques les plus utilisées pour casser les symétries. Après avoir montré pourquoi il est difficile d'étendre cette approche aux formules booléennes quantifiées, nous montrons comment générer une nouvelle formule équivalente à la formule de départ, mais ne contenant pas de symétries. L'évaluation expérimentale menée sur un des meilleurs solveurs QBF actuels montre des résultats très convaincants sur une grande variété d'instances QBF structurées.

1 Introduction

Depuis quelques années, la résolution des formules booléennes quantifiées (QBF) est un domaine de recherche en pleine ébullition. Cet intérêt peut être expliqué par plusieurs facteurs. D'une part, de nombreux problèmes en intelligence artificielle (planification, raisonnement non monotone, vérification formelle, ...) peuvent se réduire à des formules booléennes quantifiées. De l'autre, les progrès incessants réalisés sur la résolution du problème SAT permettent maintenant de s'attaquer à des formules de plus en plus grandes et difficiles. C'est ainsi que dernièrement, de nombreux solveurs pour résoudre des instances QBF ont été proposés (e.g. [10, 14, 11]). La plupart d'entre eux, étendent les derniers résultats obtenus sur la résolution du problème SAT. Ceci n'a rien de surprenant puisque les formules QBF sont une extension naturelle du problème SAT (problème de satisfaction d'une formule booléenne) où les variables sont quantifiées existentiellement ou universellement.

Certaines classes d'instances QBF encodent des problèmes dit « du monde réel » et contiennent de nombreuses symétries. On peut espérer une réduction de l'espace de recherche en éliminant ces symétries. En effet, l'exploitation des symétries a déjà montré son importance pour résoudre des problèmes combinatoires intraitables jusqu'alors. Ceci a par exemple été le cas pour les problèmes de satisfaction de contraintes (e.g. [13, 8] et pour le problème SAT (e.g. [2, 7, 5]).

Dans cet article nous introduisons un pré-traitement pour casser les symétries dans les formules booléennes quantifiées. Les symétries sont d'abord détectées en utilisant l'algorithme proposé dans Audemard-etal dans [4]. Pour casser les symétries des clauses sont ajoutées à la formule originale à l'aide de variables supplémentaires. La formule obtenue, équivalente en terme de validité à la formule originale, peut être résolue à partir de n'importe quel solveur QBF. Ce qui n'est pas le cas avec l'approche hybride QBF/SAT proposée dans [4] où les prédicats cassant les symétries étaient séparés de la formule QBF originale.

Le reste de l'article est organisé de la manière suivante. Après des définitions préliminaires sur les formules booléennes quantifiées, nous introduisons le cadre des symétries. Par la suite, nous décrivons notre approche pour casser les symétries dans les QBF. Après une validation

expérimentale montrant de bons résultats, nous concluons et proposons divers axes de recherche futurs.

2 Préliminaires

2.1 Formules Booléennes Quantifiées

Soit $\mathcal{P}$ un ensemble de variables propositionnelles. $\mathcal{L}_\mathcal{P}$ désigne alors le langage des formules booléennes quantifiées construites à partir de $\mathcal{P}$ en utilisant les formules booléennes classiques (incluant les constantes $\top$ et $\bot$) ainsi que les quantifications $\exists$ et $\forall$ s'appliquant sur les variables.

On considère les formules booléennes quantifiées mises sous forme prénexe : $\Phi = Q_1X_1, \ldots, Q_mX_m\Psi$ (ou encore $QX\Psi$, QX est le préfixe de Φ et Ψ sa matrice) tel que $Q_i \in \{\exists, \forall\}$, $X_1, \ldots, X_m$ sont des ensembles disjoints de variables et Ψ une formule booléenne. Les variables consécutives ayant le même quantificateur sont groupées. Une QBF Φ est dite sous forme normale si Φ est sous forme prénexe et si Ψ est sous forme normale conjonctive (CNF). Nous définissons $Var(\Phi) = \bigcup_{i\in\{1,\ldots,m\}} X_i$ l'ensemble des variables de Φ. Un littéral est une variable propositionnelle sous forme positive (l) ou négative ($\neg l$). $Lit(\Phi) = \bigcup_{i\in\{1,\ldots,m\}} Lit(X_i)$ est l'ensemble complet des littéraux de Φ, où $Lit(X_i) = \{x_i, \neg x_i | x_i \in X_i\}$. Pour une variable $x \in Var(\Phi)$ telle que $x \in X_i$, on définit $rank(x) = i$. L'ordre lexicographique $<_{lex}$ est définit comme suit : $x_1 <_{lex} x_2 <_{lex} \cdots <_{lex} x_i \ldots$.

2.2 Symétries dans les formules booléennes quantifiées

Soient $\Phi = Q_1X_1, \ldots, Q_mX_m\Psi$ une formule booléenne quantifiée et $\sigma : Lit(\Phi) \mapsto Lit(\Phi)$ une permutation sur les littéraux de Φ. La permutation σ appliquée à Φ est donc définie comme suit $\sigma(\Phi) = Q_1\sigma(X_1), \ldots, Q_m\sigma(X_m)\sigma(\Psi)$. Par exemple, si Ψ est sous forme clausale alors $\sigma(\Psi) = \{\sigma(c)/c \in \Psi\}$ et $\sigma(c) = \{\sigma(l)/l \in c\}$.

Définition 1 *Soit $\Phi = Q_1X_1, \ldots, Q_mX_m\Psi$ une formule booléenne quantifiée. Soit σ une permutation sur les littéraux de Φ. σ est une symétrie de Φ si et seulement si :*

- $\forall x \in Lit(\Phi)$, $\sigma(\neg x) = \neg\sigma(x)$
- $\sigma(\Phi) = \Phi$ *i.e.* $\sigma(\Psi) = \Psi$ *et* $\forall i \in \{1, \ldots, m\}$ $\sigma(X_i) = X_i$.

Une symétrie σ peut être vue comme une liste de cycles $(c_1 \ldots c_n)$ où chaque cycle c_i est une liste de littéraux $(l_{i_1} \ldots l_{i_{n_i}})$ tel que $\forall 1 \leq k < n_i, c_i(l_{i_k}) = l_{i_{k+1}}$ et $c_i(l_{i_{n_i}}) = l_{i_1}$. Dans cet article nous ne considérons que des symétries contenant des cycles binaires.

Notons que chaque symétrie σ d'une QBF Φ est également une symétrie de de la formule booléenne Ψ. La réciproque n'est pas vraie. L'ensemble des symétries de Φ est donc un sous-ensemble des symétries de Ψ.

Il a été prouvé dans [6, 7] que la détection des symétries d'une formule booléenne est équivalente au problème d'isomorphisme de graphe (problème consistant à trouver un « *mapping* » entre deux graphes G et H). La complexité du problème d'isomorphisme est un problème ouvert (il est communément admis qu'il est à la frontière entre les classes P et NP [6]). Dans notre contexte, nous avons besoin de détecter des automorphismes de graphes (trouver un « *mapping* » entre G et G). Beaucoup de programmes ont été proposés pour résoudre ce problème. Mentionnons tout particulièrement NAUTY [12] qui est l'un des plus efficace en pratique.

Afin de détecter les symétries dans une formule CNF, Aloul *et al*. [1, 2] ont proposé une technique intéressante transformant la formule en un graphe G_Ψ contenant des sommets colorés. Bien entendu, ces couleurs sont prises en compte lors de la recherche d'automorphisme de graphes (i.e. des sommets ne peuvent pas être permutés avec des sommets d'une couleur différente). La formule CNF est transformée en graphe de la manière suivante :

- Chaque variable est représentée par deux sommets, un pour le littéral positif, le second pour le littéral négatif. Une couleur gris foncé est donnée à ces sommets.
- Une arête relie deux littéraux opposés.
- Toute clause non binaire est représentée par un sommet de couleur gris clair elle est reliée à tous les litéraux qui la compose.
- Toute clause binaire $l_1 \vee l_2$ est représentée par une double arête entre les sommets l_1 et l_2. Ceci afin de réduire le nombre de sommets nécessaires.

Dans [4], les auteurs ont proposé une simple extension des travaux de Aloul et al [1] pour détecter les symétries dans une formule booléenne quantifiée. Pour réaliser cette extension, il est nécessaire de prendre en compte le préfixe de la QBF (voir la condition 2 de la définition 1). En effet, les symétries ne doivent pas permuter des variable n'appartenant pas au même groupe de quantificateurs. Pour distinguer de telles variables, une couleur par groupe de quantificateur est introduite dans la transformation de la formule en graphe.

3 Élimination des symétries dans les QBFs

L'exploitation des symétries a été largement étudié dans le contexte de la satisfaction de contrainte et le problème de satisfiabilité. Différentes approches ont été proposées

que l'on peut classifier en deux catégories, les unes dynamiques, et les autres statiques. L'élimination dynamique consiste à chercher et à éliminer les symétries durant la recherche [5, 8]. Les schémas d'élimination statiques font référence aux techniques de détection et de suppression des symétries en tant qu'étape de prétraitement. Les symétries sont généralement éliminées en ajoutant des contraintes appelées SBP (symmetry breaking predicates) [6, 1]. Un tel SBP élimine tous les modèles de chaque classe d'équivalence d'interprétations (modulo une symétrie) excepté un. Cependant, dans le cas général l'ensemble des symétries, et donc des prédicats générés, peut être de taille exponentiel. Récemment, Aloul *et al* [1, 3] ont étendu l'approche de Crawford [6] en utilisant la théorie de groupe et le concept de non redondance des générateurs, réduisant considérablement la taille des SBP.

Dans cette section, nous commençons par expliciter la difficulté à étendre la notion d'élimination des symétries du cas SAT au cas QBF. Nous montrons ensuite comment traiter cette difficulté et comment générer une nouvelle formule sans symétrie et équivalente pour la validité à celle de départ.

3.1 Motivation

L'exemple suivant, montre la difficulté à étendre la notion d'élimination des symétries du cas SAT au cas QBF.

Exemple 1 *Soit* $\Phi = \forall x_1 x_2 \exists x_3 x_4\ \Psi$ *une QBF telle que* $\Psi = (x_1 \vee \neg x_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee x_4)$.
La permutation $\sigma = \{(x_1, x_2)(x_3, x_4)\}$ *est une symétrie de* Φ. *pour éliminer cette symétrie* σ *en utilisant l'approche traditionnelle du cas SAT, on ajoute à la formule le SBP suivant :* $\{(\neg x_1 \vee x_2)$,$(\neg x_1 \vee \neg x_3 \vee x_4)$,$(\ x_2 \vee \neg x_3 \vee x_4)\}$. *Comme la clause* $(\neg x_1 \vee x_2)$ *est universellement quantifiée, La nouvelle QBF obtenue est invalide alors que* Φ *est valide.*

Nous montrons maintenant comment étendre la notion de SBP aux formules booléennes quantifiées. Après une présentation formelle de notre approche pour le cas d'une seule symétrie, une généralisation est décrite pour un ensemble quelconque de symétries.

3.2 Élimination d'une seule symétrie

Avant de décrire le traitement d'élimination des symétries dans les formules QBF, nous commençons par donner une brève description du SBP.
Soit Ψ une formule CNF et $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$ une symétrie de Ψ. Le SBP associé à σ est définit comme suit :
$x_1 \leq y_1$
$(x_1 = y_1) \Rightarrow x_2 \leq y_2$
$\vdots$
$(x_1 = y_1) \dots (x_{i-1} = y_{i-1}) \Rightarrow x_i \leq y_i$
$\vdots$
$(x_1 = y_1) \dots (x_{i-1} = y_{i-1}) \dots (x_{n-1} = y_{n-1}) \Rightarrow x_n \leq y_n$

Dans le cas SAT, le SBP décrit précédemment est transformé sous forme CNF et ajouté à la formule booléenne d'origine. Comme cela a été expliqué dans l'exemple 1, quand les formules QBF sont considérées, certaines clauses du SBP peuvent être universellement quantifiées. Ajouter de telles clauses à la formule QBF originale conduit à une invalidité alors que celle-ci ne l'est pas obligatoirement. Dans ce qui suit, on décrit comment outrepasser un tel problème.

Pour générer des SBP, il faut choisir un ordre des variables afin d'ordonner les cycles. En conséquence, le SBP correspondant dépend de l'ordre imposé. Dans le cas QBF comme dans le cas SAT, un tel ordre peut avoir un grand impact sur la taille de l'espace de recherche. Nauthy génère le SBP en suivant l'ordre lexicographique. Dans le cas QBF, on doit ordonner les cycles des symétries suivant celui du préfixe.

Définition 2 *Soit* $\Phi = Q_1 X_1 \dots Q_i X_i \dots Q_m X_m\ \Psi$ *une QBF et* σ *une symétrie de* Φ. *On définit* $\sigma \uparrow X_i$ *comme la sous-séquence de la symétrie* σ *restreinte aux cycles dont les variables appartiennent à* X_i. *une symétrie* σ *peut donc s'écrire sous forme* $\sigma = \{\sigma_1 \dots \sigma_i \dots \sigma_m\}$ *tel que* $\sigma \uparrow X_i = \sigma_i$. *Si pour tout* $i \in \{1 \dots m\}$, σ_i *est lexicographiquement ordonnée alors* σ *est dite c-ordonnée.*

Exemple 2 *Soit* $\Phi = \exists x_5, x_6 \forall x_1, x_2 \exists x_3, x_4\ (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_4)(x_1 \vee x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_1 \vee x_5)(x_1 \vee x_6) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_5 \vee \neg x_6)$
Φ *admet une symétrie* σ. *La génération de cette symétrie à partir de Nauty donnera* $\sigma = \{(x_1, x_2)(x_3, x_4)(x_5, x_6)\}$. *Réordonnée en respectant le préfixe de* Φ *cela donne* $\sigma = \{\sigma_1, \sigma_2, \sigma_3\}$ *avec* $\sigma_1 = \sigma \uparrow X_1 = (x_5, x_6)$, $\sigma_2 = \sigma \uparrow X_2 = (x_1, x_2)$, $\sigma_4 = \sigma \uparrow X_3 = (x_3, x_4)$. *Comme* σ_1, σ_2 *et* σ_3 *sont lexicographiquement ordonnées, la forme c-ordonnée de* σ *est* $\sigma = \{\sigma_1, \sigma_2, \sigma_3\} = \{(x_5, x_6)(x_1, x_2)(x_3, x_4)\}$

Dans le reste de ce papier, les symétries seront considérées c-ordonnées.

Définition 3 *Soit* Φ *une QBF,* σ *une symétrie de* Φ *et* $c = (x, y)$ *un cycle de* σ. *on appelle* x *(resp.* y*) un in-litéral (resp. out-litéral). Un cycle* $c = (x, y)$ *est dit universel si* x *et* y *sont universels sinon il est dit existentiel.*
Une symétrie σ *est dite universelle si elle contient au moins un cycle universel. Dans le cas contraire elle est dite existentielle.*

Pour les symétries existentielles, le SBP classique généré est ajouté à la formule originale en préservant sa validité. Cependant, pour les symétries universelles des prédicats supplémentaires sont nécessaires.

Dans le cas d'une symétrie universelle, les out-litéraux peuvent voir leur valeur imposée par le SBP. Pour contourner ce problème, les out-litéraux vont être requantifiées existentiellement avec un rang supérieur à celui du in-litéral associé dans le cycle. Afin de préserver la validité de la formule originale, nous introduisons une variable auxiliaire permettant de simuler l'universalité originale du out-litéral. La définition suivante formalise ceci.

Définition 4 *Soit* $\sigma = \{(x_1, y_1) \dots (x_k, y_k) \dots (x_n, y_n)\}$ *avec* $1 \leq k \leq n$ *une symétrie universelle c-ordonnée. On définit* $QSBP(\sigma) = \cup\{qsbp(\sigma(y_k)), 1 \leq k \leq n,$ y_k *universel*$\}$ *comme le* $QSBP$ *associé à* σ. *Le* $QSBP(\sigma)$ *est construit en utilisant les deux étapes suivantes :*

1. *Ajout de variables auxiliaires : Pour chaque cycle universel* $(x_i, y_i) \in \sigma$, *on associe une nouvelle variable universelle* y'_i *au out-litéral* y_i. *Le quantificateur universel de* y_i *est substitué par un quantificateur existentiel.*

2. *Génération de nouvelles contraintes :*

   - *Si* (x_1, y_1) *est un cycle universel alors* $qsb(\sigma(y_1)) = \{\neg x_1 \Rightarrow (y_1 \Leftrightarrow y'_1)\}$
   - *Pour tout* $k > 1$, *Si* (x_k, y_k) *est un cycle universel alors* $qsbp(\sigma(y_k))$ *contient les contraintes suivantes :*
     - $\neg x_k \Rightarrow (y_k \Leftrightarrow y'_k)$ *si* $x_k \neq \neg y_k$
     - $((\neg x_j \wedge y_j) \Rightarrow (y_k \Leftrightarrow y'_k)), \forall j$ *tel que* $1 \leq j < k$

Exemple 3 *En considérant la QBF de l'exemple 2, on est en présence d'une symétrie universelle c-ordonnée contenant un seul cycle universel* (x_1, x_2). *D'après la définition du QSBP,* x_2 *va être requantifiée existentiellement. Elle sera associée à une nouvelle variable, appelons là* x'_2, *et QSBP(*σ*)= qsbp(*$\sigma(x_2)$*)=* $\{(\neg x_5 \wedge x_6 \Rightarrow (x_2 \Leftrightarrow x'_2)) \wedge (\neg x_1 \Rightarrow (x_2 \Leftrightarrow x'_2))\}$

Soit $\sigma = \{c_1 \dots c_i \dots c_n\}$ tel que $c_i = (x_i, y_i)$. Pour connaître si y_i a sa valeur imposée, il faut connaître les valeurs affectées aux variables des cycles $\{c_1 \dots c_{i-1}\}$ et aussi la valeur de x_i. Ces conditions nous permettent de construire un graphe définit ci-dessous permettant de trouver un ordre entre les variables d'origine et des variables auxiliaires.

Définition 5 *Soit* $\Phi = Q_1X_1 \dots Q_iX_i \dots Q_mX_m\ \Psi$ *une QBF,* $\sigma = \sigma_1 \dots \sigma_i \dots \sigma_m$ *tel que* $\forall i \in \{1 \dots m\}$ *une symétrie universelle c-ordonnnée. Soit* j *tel que* $Q_j = \forall$ *et* $\sigma_j = \sigma \uparrow (X_j) = (x_1, y_1) \dots (x_n, y_n)$ *et* $Y' = \{y'_1 \dots y'_n\}$ *l'ensemble des variables ajoutées. On définit un nouveau rang de* $Var(\sigma_j) \cup Y'$ *comme suit :* $\forall (x_k, y_k) \in \sigma_j$,

- $rank(x_k) < rank(y'_k) < rank(y_k)$ *tq.* $1 \leq k \leq n$ *et*
- $rank(y_k) < rank(y'_{k+1})$ *tel que* $1 \leq k < n$.

Cette relation qui vient d'être introduite peut être exprimée sous forme d'un graphe dirigé appelé graphe de précédence.

Définition 6 *Soit* $\Phi = Q_1X_1 \dots Q_iX_i \dots Q_mX_m\ \Psi$ *une QBF,* $\sigma = \{\sigma_1 \dots \sigma_i \dots \sigma_m\}$ *une symétrie universelle et c-ordonnée tel que* $i \in \{1 \dots m\}$ *et soit* $\sigma_j = \sigma \uparrow X_j = \{(x_1, y_1), \dots (x_n, y_n)\}$ *tel que* $Q_j = \forall$. *On définit* $\mathcal{G}_{\sigma_j}(\mathcal{V}, \mathcal{A})$ *le graphe de précédence associé à* σ_j *avec* $Q_j = \forall$ *comme suit :*

- $\mathcal{V} = \bigcup_{1 \leq k \leq n} \{x_k, y_k, y'_k\}$
- $\mathcal{A} = \{\bigcup_{1 \leq k \leq n} \{(x_k, y'_k), (y'_k, y_k)\}\} \cup \{\bigcup_{1 \leq k < n} \{(y_k, y'_{k+1})\}\}$

L'étape suivante consiste à transformer le quantificateur Q_jX_j afin de respecter la définition du SBP. Ce dernier doit respecter les conditions sur l'ordre que l' on vient d'introduire liant les anciennes variables du groupe X_j aux variables auxiliaires. La méthode du tri topologique est utilisée à cette fin.

Dans le cas d'une symétrie singulière, un tel graphe est acyclique. La figure 1, représente le graphe de précédence de $\sigma_j = \sigma \uparrow (X_j)$ tel que $Q_j = \forall$ et $\sigma_j = \{(x_1, y_1) \dots (x_n, y_n)\}$. Un ordre possible des variables est le suivant : $[x_1 \dots x_n y'_1 y_1 \dots y'_n y_n]$

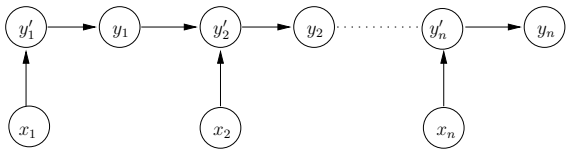

Figure 1: graphe de précédence de σ_j

Définition 7 *Soit* $\Phi = Q_1X_1 \dots Q_iX_i \dots Q_mX_m\ \Psi$ *une QBF,* $\sigma = \{\sigma_1 \dots \sigma_i \dots \sigma_m\}$ *une symétrie universelle c-ordonnée. Pour* $j \in \{1 \dots m\}$ $\sigma_j = \sigma \uparrow X_j = \{(x_1, y_1), \dots (x_n, y_n)\}$. *Chaque groupe de quantificateur* Q_jX_j *de* Φ *est réécrit comme suit :*

- *Si* $Q_j = \forall$ *et* $\sigma \uparrow X_j \neq \emptyset$ *alors* $Q_j^P X_j^P = \forall (X_j \backslash Var(\sigma_j)) x_1 \dots x_n \forall y'_1 \exists y_1 \dots \forall y'_n \exists y_n$ *(voir figure 1) tel que* $X_j^P = X_j \cup \{y'_1 \dots y'_n\}$

- *sinon* $Q_j^P X_j^P = Q_j X_j$ *avec* $X_j^P = X_j$

Φ *est donc réécrit comme*
$\Phi^S = Q_1^P X_1^P \dots Q_i^P X_i^P \dots Q_m^P X_m^P \; \Psi \wedge SBP \wedge QSBP$

Illustrons cette construction en considérant la formule Φ de l'exemple 1. La nouvelle QBF est définie par :
$\Phi^S = \forall x_1 x'_2 \exists x_2 x_3 x_4 \; \Psi^S$ avec :
$\Psi^S = \{(x_1 \vee \neg x_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \neg x_4)$
$\wedge(\neg x_1 \vee x_3) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_2)$
$\wedge(\neg x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_2 \vee x_4)$
$\wedge(x_1 \vee \neg x_2 \vee x'_2) \wedge (x_1 \vee x_2 \vee \neg x'_2)\}$
On impose entre les variables x_1, x_2 et x'_2 la relation de précédence suivante : $rank(x_1) < rank(x'_2) < rank(x_2)$. La figure 2 représente l'arbre de recherche de la nouvelle QBF Φ^S.

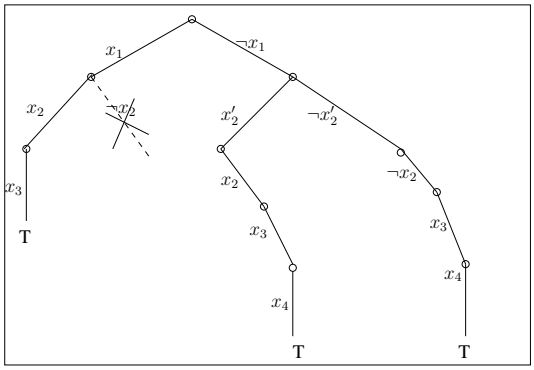

Figure 2: Arbre de recherche Φ^S de l'exemple 1

Lorsque x_1 est affecté à vrai, à partir de la clause $(\neg x_1 \vee x_2)$ on déduit que x_2 est vrai aussi. Si x_1 est affecté à faux, le quantificateur original de x_2 est déduit par substitution. Ceci est garantit grâce à l'ajout de la contrainte $\neg x_1 \Rightarrow (x_2 \Leftrightarrow x'_2)$ qui peut s'écrire sous la forme de conjonction des deux clauses :$(x_1 \vee \neg x_2 \vee x'_2) \wedge (x_1 \vee x_2 \vee \neg x'_2)$ exprimant le fait que lorsque x_1 est affecté à faux, x_2 et x'_2 deviennent équivalents. Comme le rang de x'_2 quantifié universellement est inférieur au rang de x_2 quantifié quant à lui existentiellement, on peut tout simplement remplacer les occurrences de x_2 par x'_2 et ôter x_2 du préfixe. Ce qui permet de retrouver le x_2 (représenté par x'_2) d'origine. Cette combinaison entre SBP et QSBP permet de préserver la validité de la formule d'origine.

Propriété 1 *Soit Φ une QBF et σ une symétrie, alors Φ est valide si et seulement si Φ^S est valide.*

Preuve : La QBF Φ^S est obtenue en ajoutant le SBP pour les symétries existentielles, et le SBP et le $QSBP$ pour les symétries universelles à la formule QBF originale et en réécrivant son préfixe. Pour montrer que les deux QBFs sont équivalentes il suffit de montrer que la nouvelle formule respecte exactement la définition du SBP. Pour les symétries existentielles le résultat est immédiat. Considérons donc que notre symétrie est universelle et c-ordonnée.

Soit $\sigma = \{\sigma_1 \dots \sigma_i \dots \sigma_m\}$ telle que $\{\sigma_1 \dots \sigma_{j-1}\}$ existentielles et σ_j universelle. avec $\{\sigma_1 \dots \sigma_{j-1}\} = \{(x_1, y_1) \dots (x_{k-1}, y_{k-1})\}$ et $\sigma_j = \{(x_k, y_k) \dots (x_n, y_n)\}$. On distingue deux cas :

- $\forall\, (1 \leq i \leq (k-1)$ on a : $x_i = y_i$.

  Dans ce cas notre symétrie nous permet encore d'effectuer des coupures dans l'arbre de recherche. Après avoir affecté l'ensemble $X'_k \backslash Var(\sigma_k)$, on affecte l'ensemble $\{x_k \dots x_n\}$. Supposons que $x_k = \dots x_{k+s-1} = true$ et $x_{k+s} = false$. D'après la définition du SBP, $y_k = \dots y_{k+s-1} = true$. Puisque $x_{k+s} = false$, d'après la définition du QSBP, $y'_{k+s} \Leftrightarrow y_{k+s}$ et on peut remplacer les ocurrences de y_{k+s} par y'_{k+s}.

  - si $y'_{k+s} = true$ le SBP devient vrai (ne permet plus de couper des branches) et en utilisant la définition du $QSBP$ $\forall j > (k+s) \mid y'_j \Leftrightarrow y_j$. Dans ce cas $Q_j^P X_j^P = Q_j X_j$.
  - Si $y'_{k+s} = false$ on passe aux valeurs affectés à $x_{k+s+1} \dots x_n$

- $\exists (x_{j_1}, y_{j_1}) \mid (k-1) \leq j_1 \leq n$ et $(x_{j_1} \neq y_{j_1})$.

  - si $x_{j_1} = false$ et $y_{j_1} = true$. Le SBP devient Vrai et d'après le QSBP, $y'_{j_1} \Leftrightarrow y_{j_1}$ pour tout $i > (j-1)$. Dans ce cas $Q_j^P X_j^P = Q_j X_j$.
  - si $x_{j_1} = true$ et $y_{j_1} = false$, dans ce cas il existe $j_0 \leq j_1$ tel que $x_{j_0} = false$ et $y_{j_0} = true$. Dans ce cas d'après le QSBP pour tout $k \leq i \leq n$, $y'_i \Leftrightarrow y_i$ et $Q_j^P X_j^P = Q_j X_j$. □

La plupart des solveurs QBF suivent l'ordre imposé par le préfixe et utilisent uniquement les heuristiques pour choisir une variable parmi celles d'un même groupe de quantificateur. Comme notre algorithme est un prétraitement qui modifie la formule de départ pour éliminer ses symétries, nous devons obliger la variable x_1 a être plus externe que la variable x'_2. Pour ce faire, une nouvelle variable existentielle α est introduite telle que $rank(x_1) < rank(\alpha)$ et $rank(\alpha) < rank(x'_2)$. Dans ce cas, notre nouvelle QBF devient : $\forall x_1 \exists \alpha \forall x'_2 \exists x_2 x_3 x_4 \; \Psi^S$

3.3 Élimination d'un ensemble de symétries

Dans la section 3.2, la transformation d'une formule ne contenant qu'une seule symétrie est explicitée. Quand une formule QBF contient plusieurs symétries, le traitement se complique du fait que ces symétries ne peuvent être cassées les unes indépendamment des autres en utilisant l'approche

décrite dans la section précédente. Les interactions entre ces différentes symétries nous imposent d'effectuer des modifications qui les prennent en compte.

Illustrons ces interactions entre les symétries en considérant l'exemple d'une formule QBF Φ ayant deux symétries σ et σ' telles que $\sigma \uparrow X_1 = \{(x_1, x_4)\}$ et $\sigma' \uparrow X_1 = \{(x_3, x_4)\}$ avec $Q_1 = \forall$. Le $QSBP$ généré à partir de la première (resp. deuxième) symétrie est $\neg x_1 \Rightarrow (x_4 \Leftrightarrow x'_4))$ (resp. $\neg x_3 \Rightarrow (x_4 \Leftrightarrow x'_4))$). On peut voir, que les deux symétries partagent le même out-litéral x_4. Comme expliqué précédemment, pour un cycle universel donné, le $QSBP$ exprime les conditions sous lesquelles le out-litéral et son litéral auxiliaire correspondant sont équivalents. En considérant les deux symétries σ et σ', on peut constater aisément que x'_4 est équivalent à x_4 seulement si aucune des deux symétries ne peut impliquer x_4. Pour l'ensemble des deux symétries, le QSBP est donc donné par la formule : $(\neg x_1 \wedge \neg x_3) \Rightarrow (x_4 \Leftrightarrow x'_4))$ qui peut être vue comme une `corrélation` entre les deux symétries.

Dans le cas général, en présence de plusieurs symétries, un litéral est équivalent à son correspondant seulement si ce litéral ne peut être impliqué par aucune des symétries.

Définition 8 *Soit Φ une QBF, et*

- $\sigma = \{\sigma_1 \dots \sigma_i \dots \sigma_m\}$ *et* $\sigma' = \{\sigma'_1 \dots \sigma'_i \dots \sigma'_m\}$ *deux symétries de Φ. Soit $k \in \{1 \dots m\}$ tel que $Q_k = \forall$, $(x_k, y_k) \in \sigma \uparrow X_k$ et $(x'_k, y'_k) \in \sigma' \uparrow X_k$ et $y_k = y'_k = y_{k_0}$ et*
- $qsbp(\sigma(y_{(k_0)})) = \{\alpha_1 \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0})) \dots \alpha_N \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0})\}$ *et*
- $qsbp(\sigma'(y_{k_0})) = \{\beta_1 \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0})) \dots \beta_M \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0})\}$.

On définit l'opérateur binaire η exprimant la `corrélation` entre σ et σ' par rapport à y_{k_0} comme suit :

$\eta(\sigma(y_{k_0}), \sigma'(y_{k_0})) = \{(\alpha_1 \wedge \beta_1) \Rightarrow (x \Leftrightarrow x') \dots (\alpha_1 \wedge \beta_M) \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0}) \dots (\alpha_N \wedge \beta_1) \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0}) \dots (\alpha_N \wedge \beta_M) \Rightarrow (y_{k_0} \Leftrightarrow y'_{k_0})\}$.

Définition 9 *Soit $\mathcal{S} = \{\sigma^1 \dots \sigma^n\}$ l'ensemble des symétries d'une QBF. Pour une variable $y \in X_k$ telle que $Q_k = \forall$ On définit,*

- $\mathcal{S}[y] = \{\sigma^j, \exists x \in X_k$ *tel que* $(x, y) \in \sigma^j \uparrow X_k\} = \{\sigma'^1 \dots \sigma'^{|\mathcal{S}[y]|}\}$.
- $QSBP(\mathcal{S}[y]) = \eta(\eta \dots \eta(qsbp(\sigma'^1), \sigma'^2), \dots, \sigma'^{|\mathcal{S}[y]|}) \dots)$.
- $Var(\mathcal{S}) = \bigcup(Var(X_i),\ 1 \leq i \leq m$ *et* $Q_i = \forall)$

Propriété 2 *Soit $\mathcal{S} = \{\sigma^1 \dots \sigma^n\}$ l'ensemble des symétries d'une QBF $\Phi = Q\,\Psi$. La nouvelle matrice de Ψ^S est définie comme suit :*

$$\Psi \wedge (\bigwedge_{1 \leq i \leq n} SBP(\sigma^i)) \wedge (\bigwedge_{y \in Var(\mathcal{S})} QSBP(\mathcal{S}[y]))$$

Notons que pour une symétrie σ d'une formule QBF, La taille du $SBP(\sigma)$ généré est linéaire en la taille de σ [1]. La propriété ci-dessous donne l'ordre de grandeur de la complexité de QSBP(σ).

Propriété 3 *Soit σ une symétrie d'une QBF. Dans le pire cas, la complexité spatiale de $QSBP(\sigma)$ est en $O(|\sigma|^2)$.*

Preuve : Soit $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$. Dans le pire cas $\forall i \in \{1 \dots n\}$ (x_i, y_i) est un cycle universel. Dans ce cas $QSBP(\sigma) = \bigcup_{1 \leq i \leq n} \{qsbp(\sigma(y_i))\}$. Soit $|QSBP(\sigma)| = \sum_{1 \leq i \leq n} |qsbp(\sigma(y_i))|$. D'après la définition du $qsbp(\sigma(y_i))$ sa taille est égale à $2(i-1)+2 = 2i$ (voir définition 4). donc, $|QSBP(\sigma)| = \sum_{1 \leq i \leq n} 2i$ ce qui est égal $n(n+1)$

Notons que dans l'implantation de notre algorithme, on exploite avec quelques modifications le SBP généré par Shatter qui introduit des variables auxiliaires pour obtenir une taille linéaire. En exploitant ces mêmes variables supplémentaires, la taille du QSBP qu'on génère devient linéaire.

3.3.1 Transformation du préfixe

Dans la section 3.2, nous avons vu comment réécrire le préfixe d'une formule QBF contenant une seule symétrie. Dans le cas de plusieurs symétries, le processus consiste à construire le graphe de précédence correspondant à l'ensemble des symétries et à appliquer le tri topologique pour déterminer un ordre possible sur les variables. Ceci est illustré dans l'exemple 4.

Exemple 4 *Soit $\Phi = Q_1 X_1 \dots Q_m X_m\ \Psi$ une QBF, σ et σ' deux symétries de Φ avec $\sigma \uparrow X_1 = \{(x_1, x_2)(x_3, x_4)(x_5, x_6)\}$, $\sigma' \uparrow X_1 = \{(x_1, x_3)(x_2, x_5)\}$ et $Q_1 = \forall$. Le graphe de précédence associé à σ et σ' est représenté dans la figure 3. Le quantificateur $Q_1^P X_1^P$ est réécrit sous forme de : $\forall x_1 \exists \alpha \forall x'_2, x'_3 \exists x_2 x_3 \forall x'_6 \exists x_6$*

Comme pour le $QSBP$, les connections entre les différentes symétries nous contraignent à réécrire le préfixe de la QBF lorsqu'on considère un ensemble de symétries. Dans ce cas, le graphe de précédence associé à l'ensemble des symétries peut être cyclique comme le montre l'exemple 5 :

Exemple 5 *Soit $\Phi = Q_1 X_1 \dots Q_i X_i \dots Q_m X_m \Psi$ une QBF, σ et σ' deux symétries de Φ tel que $\sigma \uparrow X_i =$*

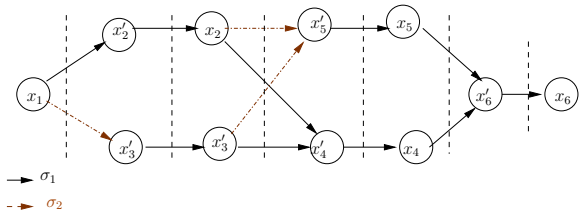

Figure 3: graphe de précédence de $\sigma_1 \uparrow X_1$ et $\sigma_2 \uparrow X_1$ (exemple 4)

$\{(x_1, x_2)(x_3, x_4)\}$ *et* $\sigma' \uparrow X_i = \{(x_1, x_4)(x_2, x_3)\}, Q_i = \forall$. *Soit* $\mathcal{G}_{\sigma,\sigma'} = (\mathcal{V}, \mathcal{A})$ *Le graphe de précédence associé à* σ *et* σ' *tel que :*

- $\mathcal{A} = \{x_1 \dots x_4\} \cup \{x'_2, x'_3, x'_4\}$
- $\mathcal{V} = \bigcup_{1<i\leq 4}\{(x_i, x'_i)\} \cup \{(x_1, x'_2), (x_2, x'_4), (x_3, x'_4), (x_1, x'_4), (x_2, x'_3), (x_4, x'_3)\}$

On remarque aisément que $\mathcal{G}_{\sigma,\sigma'}$ contient le cycle suivant : $[x'_3, x_3, x'_4, x_4, x'_3]$. Or, le tri topologique ne peut pas être appliqué sur un graphe cyclique. Pour notre exemple, on ne peut pas choisir arbitrairement un ordre entre x_3 et x_4. Une des solutions est de modifier l'ordre lexicographique en un ordre permettant d'obtenir un graphe acyclique. C'est le cas en considérant l'ordre suivant $[x_1, x_2, x_4, x_3]$. Des investigations sont en cours pour trouver une méthode générale afin de rendre le graphe acyclique.

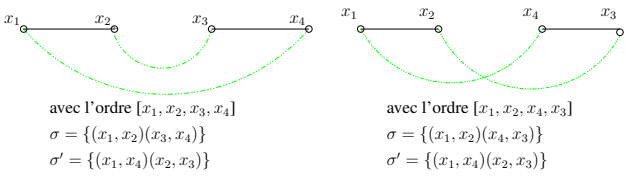

Figure 4: Représentation axiale des symétries σ et σ'

La figure 4 est constituée de deux représentations axiales des symétries de l'exemple 5 en considérant l'ordre lexicographique($[x_1, x_2, x_3, x_4]$) et l'ordre $[x_1, x_2, x_4, x_3]$. On peut remarquer que pour l'ordre lexicographique on est en présence d'un cycle de symétrie (x_2, x_3) contenu à l'intérieur du cycle de symétrie (x_1, x_4) ce qui n'est pas le cas avec le deuxième ordre. On peut montrer que si une représentation axiale est telle que pour toute symétrie σ aucun de ces cycles n'est contenu à l'intérieur d'un autre cycle de la même symétrie, alors notre graphe de précédence est acyclique. Comme on ne dispose pas pour l'instant d'algorithme permettant de trouver un ordre de telle sorte que notre représentation axiale ne contient pas cette propriété, nous opérons une restriction sur les symétries afin de rendre le graphe de précédence acyclique. Cette restriction est exprimée formellement dans la définition suivante.

Définition 10 *Soit* $\Phi = Q_1X_1 \dots Q_iX_i \dots Q_mX_m\Psi$ *une QBF,* $\sigma = \sigma_1 \dots \sigma_n$ *une symétrie universelle c-ordonnée tel que* $\sigma_j = \sigma \uparrow X_j = \{(x_1, y_1), \dots (x_n, y_n)\}$, $1 \leq j \leq n$. *On définit pour tout* $j \in \{1 \dots m\}$ *tel que* $Q_j = \forall$ *la restriction de* σ_j *comme* $R_1(\sigma_j) = \{(x_1, y_1)\} \cup \{(x_k, y_k) |\ y_{k-1} <_{lex} y_k,\ 1 < k \leq n\}$.
Pour tout $j \in \{1 \dots m\}$ *tel que* $Q_j = \exists$, $R_1(\sigma_j) = \sigma_j$.
La restriction de σ *est définie comme étant :* $R_1(\sigma) = max_{1\leq k\leq n}\{\bigcup_{1\leq j\leq k}\{R_1(\sigma_j) \mid R_1(\sigma_j) = \sigma_j, 1 \leq j \leq (k-1),\ R_1(\sigma_k) \subset \sigma_k\}\}$

Pour l'exemple 5, on obtient $R_1(\sigma) = \sigma = \{(x_1, x_2)(x_3, x_4)\}$ et $R_1(\sigma') = \{(x_1, x_4)\}$. Le SPB et $QSBP$ sont générés donc à partir de $R_1(\sigma)$ et $R_1(\sigma')$.

Le second problème rencontré concerne les out-litéraux apparaissant positivement dans certaines symétries et négativement dans d'autres. L'exemple 6 illustre cette difficulté.

Exemple 6 *Soit* $\Phi = Q_1X_1 \dots Q_iX_i \dots Q_mX_m\Psi$ *une QBF,* σ *et* σ' *deux symétries de* Φ *tel que* $\sigma \uparrow X_1 = \{(x_1, x_4)\}$ *et* $\sigma' \uparrow X_1 = \{(x_2, -x_4)\}, Q_1 = \forall$. *En appliquant le processus de transformation décrit dans ce papier, la nouvelle formule* Φ^S *s'écrira :* $\forall x_1, x_2 \exists\alpha \forall x'_4 \exists x_4 \dots Q_iX_i \dots Q_mX_m\Psi \wedge SBP \wedge qsbp(\sigma(x_4), \sigma'(x_4))$. *En affectant* x_1 *et* x_2 *à vrai on implique* x_4 *et* $\neg x_4$ *et la QBF* Φ^S *est invalide.*

Une des solutions est d'adopter un autre ordre de telle sorte qu'aucun out-litéral n'apparaîsse positivement et négativement. Pour notre exemple on pourra remplacer l'ordre lexicographique par $[x_4, x_1, x_2]$. Une autre solution consiste à voir qu'à partir du SBP =$(\neg x_1 \vee x_4) \wedge (\neg x_2 \vee \neg x_4)$ on obtient par Q-résolution la clause universelle $(\neg x_1 \vee \neg x_2)$. Dans ce cas il va falloir considérer x_2 comme un out-litéral et la nouvelle formule devient : $\forall x_1 \exists\alpha \forall x'_2 \exists x_2 \exists \forall x'_4 \exists x_4 \dots Q_iX_i \dots Q_mX_m\Psi \wedge SBP \wedge qsbp(\sigma(x_4), \sigma'(x_4)) \wedge (\neg x_1 \Rightarrow (x_2 \Leftrightarrow x'_2))$, préservant ainsi l'équivalence. Une solution générale à ce problème est en cours d'investigation.

Pour nos expériences on a choisi d'effectuer une restriction qui consiste, en présence d'un out-litéral apparaissant positivement et négativement, à garder dans les symétries où ce litéral apparaît négativement uniquement les cycles le précédant. Ceci est formalisé dans la définition suivante.

Définition 11 *Soit* $\Phi = Q_1X_1 \dots Q_iX_i \dots Q_mX_m\Psi$ *une QBF,* $\mathcal{S} = \{\sigma^1 \dots \sigma^n\}$ *l'ensemble des symétries de* Φ, $y \in X_i$ *tel* $Q_i = \forall$ *et* y *apparaît comme out-litéral positivement et négativement dans* $\mathcal{S}$. *Soit* $\sigma = \{\sigma^1 \dots \sigma^p\} \subseteq \mathcal{S}$ *l'ensemble des symétries contenant* $\neg x$ *comme out-liéral. Pour tout* $j \in \{1 \dots p\}$, *on définit la restriction de* $\sigma^j = \{(x_1, y_1) \dots (x_n, y_n)\}$ *tel que* $y_k = \neg y$ *comme* $R_2(\sigma^j) = \{(x_1, y_1) \dots (x_{k-1}, y_{k-1})\}$

Les SPB et QSBP sont générés pour l'ensemble $\mathcal{S}$, à partir de $\{R_2(\sigma^1) \dots R_2(\sigma^p), \sigma^{p+1} \dots \sigma^n\}$.

4 Expérimentations

Dans [4], les auteurs montraient que de nombreuses instances QBF contiennent des symétries, et que la recherche des symétries est, dans une grande majorité des cas, peu coûteuse en temps.

Nous basons notre étude expérimentale sur un large panel d'instances (504) disponibles sur [9]. Ces instances sont partagées en diverses familles comme `toilet`, `k_*`, `FPGA`, `qshifter`. Nous comparons le solveur SEMPROP [11], l'un des plus efficaces dans la pratique, sur les instances originales (Φ) et sur les instances augmentées du QSBP (Φ^S). Les résultats sont obtenus sur un Pentium IV 2.8 Ghz avec 1 Go de RAM. Le temps CPU est limité à 1000 secondes.

La figure 5 compare le solveur SEMPROP sur les instances originales (Φ) et sur les instances augmenté du QSBP (Φ^S) introduit dans cet article (le temps pour calculer les symétries est pris en compte). Chaque point représente une instance. Un point en dessus (resp. en dessous) de la diagonale indique que l'instance originale Φ a été résolue plus rapidement (resp. plus lentement) que l'instance Φ^S par SEMPROP. Les deux axes ont une échelle logarithmique.

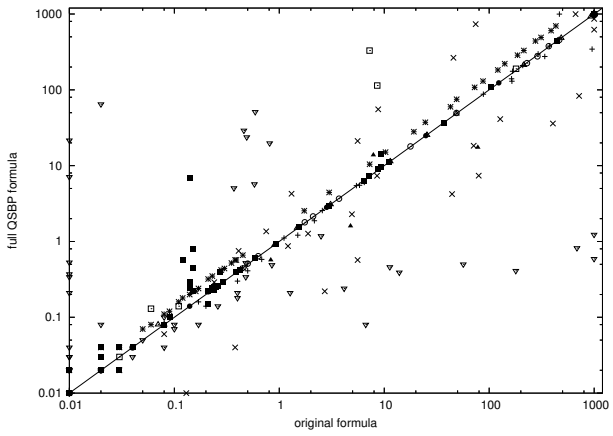

Figure 5: instances totalement asymétriques

Sur diverses familles, comme `toilet`, `k_path`, `qshifter`, l'ajout des prédicats cassant les symétries accélère la recherche de un à deux ordres de grandeur. Sur certaines instances les résultats sont très comparables. C'est le cas par exemple sur la famille `toilet_a` où les symétries concernent uniquement le groupe de quantificateur le plus interne. Il semble que ces symétries ne sont pas intéressantes pour accélérer à la recherche.

De plus, il existe beaucoup d'instances où les prédicats cassant les symétries empirent les performances de SEMPROP (par exemple les instances issues de (`k_poly`, `k_grz`). Dans la grande majorité des cas, cela est dû à la taille du QSBP qui peut être importante.

Afin de palier à ce problème, en plus des restrictions R_1 et R_2 introduites plus haut, la taille de symétries cassées est bornée par rapport au nombre de variables, mais sans jamais enlever de symétries universelles.

La figure 6 montre une comparaison de SEMPROP sur les instances originales et sur les instances augmentées du QSBP réduit.

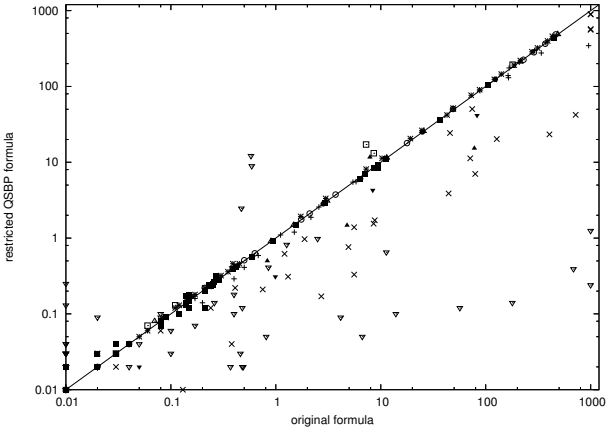

Figure 6: instances partiellement asymétriques

Comme on peut le voir, les prédicats cassant les symétries donnent de très bons résultats avec cette simple restriction. A part quelques instances, la recherche est accélérée de plusieurs ordres de grandeur. En restreignant quelques peu les symétries cassées le QSBP se retrouve réduit, donnant de bons résultats en pratique.

Pour terminer, la table 1 reporte, sur quelques instances, le nombre de variables (V), le nombre de clauses (C), le nombre de symétries (nb sym), le temps pour calculer les symétries, le temps pour résoudre l'instance originale, le temps pour résoudre l'instance totalement asymétrique et le temps pour résoudre l'instance partiellement asymétrique.

Mis à part pour les instances `qshifter` (et `z4`), les symétries sont calculées extrêmement rapidement par `Nauty` et n'influent pas sur le temps de résolution total. Pour terminer, on peut reporter que SEMPROP résout 368 sur les 504 instances originales en moins de 1000 secondes, 372 instances totalement asymétriques et 375 instances partiellement asymétriques (c'est un strict sur-ensemble des instances originales résolues par SEMPROP).

5 conclusion

Dans cet article, une nouvelle approche pour casser les symétries dans les instances QBF a été présentée. Notre pré-traitement calcule les symétries de la formule

instances						run time		
instance	V	C	validity	nb sym	sym time	orig	full sym	restricted sym
lut4_AND_fXOR	75	675	N	10	0.04	-	69.00	**5.15**
toilet_c_08_05.10	663	4953	Y	18	0.27	**0.02**	64.00	0.02
toilet_c_10_01.18	580	2709	N	12	0.07	685.00	0.80	**0.39**
toilet_c_10_01.19	612	2858	N	12	0.07	-	0.60	**0.24**
k_path_p-12	805	2238	N	7	0.07	404	36.00	**23.00**
k_path_p-13	877	2444	N	8	0.11	719	83.00	**73.00**
k_path_p-14	943	2626	N	8	0.15	45	265.00	**24.00**
k_path_p-20	1357	3790	N	11	0.25	-	-	**558.00**
TOILET7.1.iv.13	399	1491	N	9	0.04	78.0	17.4	**15.27**
TOILET7.1.iv.14	430	1608	Y	9	0.04	**7.8**	13.8	11.67
qshifter_7	263	32768	Y	2	500	**8.40**	4.30	4.30
qshifter_8	520	131072	Y	2	500	**82.00**	41.00	41.00

Table 1: Comparaison du temps de résolution

originale, et génère une nouvelle formule équivalente pour la validité et ne contenant pas de symétries. Les résultats expérimentaux ont montré une amélioration significative de SEMPROP [11], un des meilleurs solveurs actuels, sur des classes d'instances QBF. Ces résultats intéressants ont été obtenus au moyen de quelques restrictions opérées sur la taille des symétries détectées. Ces restrictions permettent de réduire la taille des SBP et QSBP générés qui peuvent être très conséquents dans certains cas. Ce travail est une première étape d'une intégration totale des prédicats cassant les symétries dans les QBF. Le but du prochain travail consiste, dans un premier temps, à étendre l'élimination aux cas des symétries dont les cycles ne sont pas binaires. Dans un second temps, nous souhaitons élaborer un algorithme capable de trouver un ordre tel que le graphe de précédence ne soit pas cyclique (on notera toutefois que les graphes de précédence des instances testées jusqu'à maintenant sont acycliques) et tel que les out-litéraux n'apparaîssent pas positivement et négativement dans l'ensemble des symétries. Enfin, des expérimentations sont en cours sur des solveurs QBF qui ne sont pas basés sur DPLL.

References

[1] Fadi A. Aloul, Arathi Ramani, Igor L. Markov, and Karem A. Sakallah. Solving difficult instances of boolean satisfiability in the presence of symmetry. Technical Report CSE-TR-463-02, University of Michigan, 2002.

[2] Fadi A. Aloul, Arathi Ramani, Igor L. Markov, and Karem A. Sakallah. Solving difficult SAT instances in the presence of symmetry. In *Proceedings of the 39th Design Automation Conference (DAC 2002)*, pages 731–736. ACM Press, 2002.

[3] Fadi A. Aloul, Arathi Ramani, Igor L. Markov, and Karem A. Sakallah. Shatter: Efficient breaking for boolean satisfiability. In *proceedings of the international conference on Design Automation Conference (DAC)*, pages 836–839. ACM Press, 2003.

[4] Gilles Audemard, Bertrand Mazure, and Lakdhar Sais. Dealing with symmetries in quantified boolean formulas. In *Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT2004)*, 2004.

[5] Belaid Benhamou and Lakhdar Sais. Tractability through symmetries in propositional calculus. *Journal of Automated Reasoning*, 12(1):89–102, February 1994.

[6] James Crawford. A theorical analysis of reasoning by symmetry in first order logic. In *Proceedings of Workshop on Tractable Reasonning, AAI92*, pages 17–22, July 1992.

[7] James Crawford, Matthew L. Ginsberg, Eugene Luck, and Amitabha Roy. Symmetry-breaking predicates for search problems. In *KR'96: Principles of Knowledge Representation and Reasoning*, pages 148–159. Morgan Kaufmann, San Francisco, California, 1996.

[8] Ian P. Gent and Barbara Smith. Symmetry breaking in constraint programming. In W. Horn, editor, *Proceedings of European Conference on Artificial Intelligence ECAI-2000*, pages 599–603. IOS Press, 2000.

[9] E. Giunchiglia, M. Narizzano, and A. Tacchella. Quantified Boolean Formulas satisfiability library (QBFLIB), 2001. `www.qbflib.org`.

[10] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. QuBE : A system for deciding Quantified Boolean Formulas Satisfiability. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'01)*, Siena, Italy, June 2001.

[11] Reinhold Letz. Lemma and model caching in decision procedures for quantified boolean formulas. In *Proceedings of Tableaux 2002*, pages 160–175, Copenhagen, Denmark, 2002.

[12] Brendan D. McKay. nauty user's guide (version 1.5). Technical Report TR-CS90 -02, Computer Science Department, Australian National University, 1990.

[13] Jean François Puget. Symmetry breaking revisited. In *SymCon'02 – Symmetry in Constraints - CP02 Workshop*, 2002.

[14] Lintao Zhang and Sharad Malik. Towards a symetric treatment of satisfaction and conflicts in quantified boolean formula evaluation. In *Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming (CP'02)*, pages 200–215, Ithaca, NY, USA, Sept. 2002.