Efficient Symmetry Breaking Predicates for Quantified Boolean Formulae

Gilles Audemard and **Said Jabbour** and **Lakhdar Sais**
CRIL - Université d'Artois - Lens, France
{audemard,jabbour,sais}@cril.univ-artois.fr

Abstract

Many reasoning task and combinatorial problems exhibit symmetries. Exploiting such symmetries has been proved useful in reducing the search space. In this paper, a formal approach for symmetry breaking in quantified boolean formula is proposed. It make use of a new efficient technique for encoding the additional symmetry predicates in prenex clausal form. The new asymmetric formula is equivalent to the original one with respect to the validity. Experimental evaluation shows significant improvements over a wide range of QBF instances.

1 Introduction

Solving Quantified Boolean Formulae (QBF) has become an attractive and important research area over the last years. Such increasing interest might be related to different factors, including the fact that many important artificial intelligence (AI) problems (planning, non monotonic reasoning, formal verification, etc.) can be reduced to QBF which is considered as the canonical problem of the PSPACE complexity class. Another important reason comes from the recent impressive progress achieved in the practical resolution of the satisfiability problem. Many solvers for QBFs have been proposed recently (e.g. [Giunchiglia *et al.*, 2001b; Letz, 2002; Zhang and Malik, 2002; Biere, 2004; Benedetti, 2005a]), most of them are obtained by extending satisfiability results. This is not surprising, since QBFs is a natural extension of the satisfiability problem (deciding whether a boolean formula in conjunctive normal form is satisfiable or not), where the variables are universally or existentially quantified.

It is well known that exploiting and removing symmetries is an important task to deal with the intractability of many combinatorial problems such as constraint satisfaction problems (CSP) [Cohen *et al.*, 2005] and satisfiability of boolean formula (SAT) [Benhamou and Sais, 1994; Crawford, 1992; Aloul *et al.*, 2002]. One of the most popular approach for breaking symmetry consists in adding new constraint, called Symmetry Breaking Predicates (SBP), to the original formula or to the constraint network. This approach independently proposed by Crawford [Crawford, 1992] and Puget [Puget, 1993] is solver independent and is performed in a preprocessing step.

Extending this approach to QBFs is a very challenging task. The problem rise from the presence of universal quantifier, while in SAT or CSP all the variables can be seen as existentially quantified. Adding the classical symmetry breaking predicates to the QBF formula do not preserve the validiy of the original formula. Recently, two promising approaches for breaking symmetry in QBF have been proposed [Audemard *et al.*, 2004; 2007]. In [Audemard *et al.*, 2004], the symmetry breaking predicates are not added to the original formula leading to a hybrid QBF-SAT formula. A QBF solver handling such hybrid formula is then proposed. Consequently, this first approach is clearly solver dependent. More recently, another technique was proposed in [Audemard *et al.*, 2007]. The authors proposed an original preprocessing technique, called `sbp4qbf`, which extends the symmetry breaking approach introduced by [Crawford, 1992] and improved by [Aloul *et al.*, 2002] for the satisfiabiliy problem. It consists in rewriting the quantifier prefix and adding new constraints in order to deal with the universal quantifiers.

In this paper, we a new formal approach for QBF symmetry breaking is proposed. Furthermore, we introduce a new and efficient way for encoding SBPs. Our approach allows us to propagate redundant existential literals and to removes universal symmetrical literals by considering some interpretations as models of the original formula.

The paper is organized as follows. After some preliminary definitions on quantified boolean formulae, symmetry framework in QBFs is presented. In section 3, we formalize symmetry breaking predicates for QBF and propose an efficient encoding. In section 4, an experimental evaluation of our approach is described. An experimental evaluation (section 4) and comparison (section 5) with the approach proposed in [Audemard *et al.*, 2007] is presented before concluding.

2 Technical Background

2.1 Quantified boolean formulae

Let $\mathcal{P}$ be a finite set of propositional variables. Then, $\mathcal{L}_\mathcal{P}$ is the language of quantified Boolean formulae built over $\mathcal{P}$ using ordinary boolean formulae (including propositional constants $\top$ and $\bot$) plus the additional quantification ($\exists$ and $\forall$) over propositional variables.
In this paper, we consider quantified boolean formula Φ in the prenex clausal form $\Phi = Q_1X_1, \ldots, Q_mX_m\psi$ (in short $QX\psi$, QX is called the prefix and ψ the matrix) where $Q_i \in \{\exists, \forall\}$, $X_1, \ldots, X_m$ are disjoint sets of variables and ψ a boolean formula in conjunctive normal form. Consecutive variables with the same quantifier are grouped. Without loss of generality, we can consider Q_m as the existential quantifier. We define $Var(\Phi) = \bigcup_{i\in\{1,\ldots,k\}} X_i$ the set of variables of Φ. A literal is the occurrence of propositional variable in either positive (l) or negative form ($\neg l$). $Lit(\Phi) = \bigcup_{i\in\{1,\ldots,k\}} Lit(X_i)$ the set of complete literals of Φ, where $Lit(X_i) = \{x_i, \neg x_i | x_i \in X_i\}$. Finally, we note ψ_X, the propositional formula ψ simplified by the assignation of all literals of X.

The semantic of QBF is associated with the notion of policy. A policy is a set of propositional models which respect some conditions. For a QBF with n universal variables, a policy contains 2^n propositional models. For more details, please refer to [Benedetti, 2005b] or [Coste-Marquis *et al.*, 2006].

2.2 Symmetries in Quantified Boolean Formulae

Let $\Phi = Q_1X_1, \ldots, Q_mX_m\psi$ be a QBF and σ a permutation over the literals of Φ i.e. $\sigma : Lit(\Phi) \mapsto Lit(\Phi)$. The permutation σ on Φ is then defined as follows: $\sigma(\Phi) = Q_1\sigma(X_1), \ldots, Q_m\sigma(X_m)\sigma(\psi)$. For example, if ψ is in clausal form then $\sigma(\psi) = \{\sigma(c) | c \in \psi\}$ and $\sigma(c) = \{\sigma(l) | l \in c\}$.

Definition 1 *Let $\Phi = Q_1X_1, \ldots, Q_mX_m\psi$ be a quantified boolean formula and σ a permutation over the literals of Φ. σ is a symmetry of Φ iff*

1. $\forall x \in Lit(\Phi), \sigma(\neg x) = \neg\sigma(x)$
2. $\sigma(\Phi) = \Phi$ *i.e* $\sigma(\psi) = \psi$ *and* $\forall i \in \{1, \ldots, m\}$ $\sigma(X_i) = X_i$.

Let us note that each symmetry σ of a QBF Φ is also a symmetry of the boolean formula ψ. The converse is not true. So the set of symmetries of Φ is a subset of the set of symmetries of ψ.

A symmetry σ can be seen as a list of cycles $(c_1 \ldots c_n)$ where each cycle c_i is a list of literals $(l_{i_1} \ldots l_{i_{n_i}})$ st. $\forall 1 \leq k < n_i, \sigma_i(l_{i_k}) = l_{i_{k+1}}$ and $\sigma_i(l_{i_{n_i}}) = l_{i_1}$.

It is well known that breaking all symmetries might lead in the general case to an exponential number of clauses [Crawford *et al.*, 1996]. In this paper, for efficiency and clarity reasons, we only consider symmetries with binary cycles. Our approach can be extended to symmetries with cycles of arbitrary size.

Detecting symmetries of a boolean formula is equivalent to the graph isomorphism problem [Crawford, 1992; Crawford *et al.*, 1996] (i.e. problem of finding a one to one mapping between two graphs G and H). This problem is not yet proved to be NP-Complete, and no polynomial algorithm is known. In our context, we deal with graph automorphism problem (i.e. finding a one to one mapping between G and G) which is a particular case of graph isomorphism. Many programs have been proposed to compute graph automorphism. Let us mention NAUTY [McKay, 1990], one of the most efficient in practice.

Recently, Aloul *et al.* [Aloul *et al.*, 2002] proposed an interesting technique that transforms CNF formula ψ into a graph G_ψ where vertices are labeled with colors. Such colored vertices are considered when searching for automorphism on the graph (i.e. vertices with different colors can not be mapped with each others).

In [Audemard *et al.*, 2004], a simple extension to QBFs formulae is given. Such extension is simply obtained by introducing a different color for each set of vertices whose literals belong to the same quantifier group. In this way, literals from different quantifier groups can not be mapped with each others (see the second condition of the definition 1). Then, to detect such symmetries, NAUTY is applied on the graph representation of the QBF.

3 Breaking symmetries in QBFs

Symmetry breaking has been extensively investigated in the context of constraint satisfaction and satisfiability problems. The different approaches proposed to break symmetries can be conveniently classified as dynamic and static schemes. Dynamic breaking generally search and break symmetries using breaking predicates or not [Benhamou and Sais, 1994; Gent and Smith, 2000]. Static breaking schemes refer to techniques that detect and break symmetries in a preprocessing step. For SAT, symmetries are generally broken by generating additional constraints, called symmetry breaking predicates (SBP) [Crawford, 1992; Aloul *et al.*, 2002]. Such SBP eliminates all models from each equivalence class of symmetric models, except one. However, in the general case, the set of symmetry predicates might be of exponential size. In [Aloul *et al.*, 2002], Aloul *et al* extend the approach of Crawford [Crawford, 1992] by using group theory and the concept of non-redundant generators, leading to a considerable reduction in the SBP size.

We briefly recall the symmetry breaking technique introduced by Crawford in [Crawford *et al.*, 1996]. Let ψ be a CNF formula and $\sigma = (x_1, y_1) \ldots (x_n, y_n)$ a symmetry of ψ. The sbp_σ associated to σ is defined as follows:
$x_1 \leq y_1$
$(x_1 = y_1) \rightarrow x_2 \leq y_2$
...
$(x_1 = y_1) \ldots (x_{n-1} = y_{n-1}) \rightarrow x_n \leq y_n$

The sbp_σ defined above expresses that, when for all $i \in \{1, \dots, k-1\}$ x_i and y_i are equivalent (take the same truth value) and x_k is $true$, then y_k must be assigned to $true$ in order to avoid the exploration of isomorphic interpretations. In the next section, an extension of this approach to QBF is described.

3.1 Motivation

In the following example, we show the main difficulty behind the extension of SAT symmetry breaking predicates (SBP) to QBFs.

Example 1 *Let $\Phi = \forall x_1 y_1 \exists x_2 y_2\ \psi$ be a QBF where $\psi = (x_1 \vee \neg x_2) \wedge (y_1 \vee \neg y_2) \wedge (\neg x_1 \vee \neg y_1 \vee x_2 \vee y_2)$. The permutation $\sigma = \{(x_1, y_1)(x_2, y_2)\}$ is a symmetry of Φ. Breaking the symmetry σ using the traditional approach, induces the following sbp_σ : $(\neg x_1 \vee y_1) \wedge (\neg x_1 \vee \neg x_2 \vee y_2) \wedge (y_1 \vee \neg x_2 \vee y_2)$. As all the literals of the clause $(\neg x_1 \vee y_1)$ are universally quantified, the new obtained QBF by adding conjunctively the sbp_σ to the original one leads to an invalid QBF formula.*

To overcome this main drawback, we will demonstrate that for symmetries containing only universal cycles, we can add disjunctively the symmetry breaking predicates. The resulting formula is combined with the one obtained for existential symmetries leading to a general QBF symmetry breaking approach.

In this paper, we propose two possible ways to handle symmetry breaking predicates in quantified boolean formulas. The first one, produces a non clausal matrix. Whereas, the second one produce a formula in prenex clausal form usually used by most of the available QBF solvers.

3.2 Breaking Symmetries : Theoretical approach

Let us now give a formal description of our approach for breaking symmetries in QBF.

Definition 2 *Let $\Phi = Q_1 X_1 \dots Q_i X_i \dots Q_m X_m\ \psi$ be a QBF and σ a symmetry of Φ. We define $\sigma \uparrow X_i$ as the sub-sequence of the symmetry σ restricted to the cycles involving variables from X_i. Then, the symmetry σ can be rewriten following the prefix ordering as $\{\sigma_1 \dots \sigma_i \dots \sigma_m\}$ such that $\sigma_i = \sigma \uparrow X_i$. When σ follow the prefix ordering, it is called p-ordered.*

In the sequel, symmetries are considered to be p-ordered.

Example 2 *Let $\Phi = \exists x_2 y_2 \forall x_1 y_1 \exists x_3 y_3\ (\neg x_1 \vee y_1 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee y_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_3 \vee \neg y_3) \wedge (x_1 \vee x_2) \wedge (x_1 \vee y_2) \wedge (\neg x_1 \vee \neg y_1 \vee \neg x_2 \vee \neg y_2)$. Φ has a symmetry $\sigma = \{(x_1, y_1)(x_2, y_2)(x_3, y_3)\}$. Reordering σ with respect to the prefix leads to $\sigma = \{\sigma_1, \sigma_2, \sigma_3\}$ st. $\sigma_1 = \sigma \uparrow X_1 = (x_2, y_2)$, $\sigma_2 = \sigma \uparrow X_2 = (x_1, y_1)$ and $\sigma_3 = \sigma \uparrow X_3 = (x_3, y_3)$.*

Definition 3 *Let Φ be a QBF, σ a symmetry of Φ and $c = (x, y)$ is a cycle of σ. A cycle c is called universal (resp. existential) if x and y are universally quantified (resp. existentially quantified). A symmetry σ is called universal (resp. totally universal) if it contains at least one universal cycle (resp. all cycles are universals), otherwise it is called existential.*

For existential symmetries of a QBF, classical SBP [Crawford *et al.*, 1996] can be translated linearly to a CNF formula thanks to new additional variables. The obtained set of clauses can be added to the QBF matrix while preserving its validity. This is formalized in property 1. The main problem arises when breaking universal symmetries (see example 1).

Property 1 *Let $\Phi = QX\psi$ be a quantified boolean formaula and σ an existential symmetry of ψ. Then Φ and $QX(\psi \wedge sbp_\sigma)$ are equivalent with respect to the validity.*

In a first step, our reasoning concerns totally universal symmetries. Let Φ be a QBF formula and $\sigma = \{(x_1, y_1)\}$ such that (x_1, y_1) is a universal cycle. The two interpretations $\{x_1, \neg y_1\}$ and $\{\neg x_1, y_1\}$ are symmetric. As x_1, y_1 are universally quantified, we can check one of them and consider the other one as a model of ψ. When x_1 is assigned to $true$, thanks to the sbp_σ we can propagate $y_1 = true$. Instead of finding a logical formulation implying $y_1 = true$ when $x_1 = true$, we transform ψ in such a way that when x_1 is assigned to $true$ and y_1 to $false$ then ψ becomes $true$. As the sbp_σ is $false$ under the interpretation $\{x_1, \neg y_1\}$ and $true$ otherwise, the formula $QX(\psi \vee \neg sbp_\sigma)$ is asymmetric respect to σ and is equivalent to Φ with respect to validity.

In the following property, we generalize this idea to a totally universal symmetry of arbitrary size.

Property 2 *Let $\Phi = QX\psi$ be a QBF formula and σ a totally universal symmetry of Φ. Then, $\Phi = QX\psi$ and $\Phi' = QX(\psi \vee \neg sbp_\sigma)$ are equivalent with respect to the validity.*

Proof :
proof of $\rightarrow$: It is obvious, since each model of ψ is also a model of $\psi \vee \neg sbp_\sigma$. Then a policy of Φ is also a policy of $QX(\psi \vee \neg sbp_\sigma)$.
proof of $\leftarrow$: Given $\mathcal{I}$ a policy of $QX(\psi \vee \neg sbp_\sigma)$, we built a policy $\mathcal{I}'$ of Φ. Let $I \in \mathcal{I}$, two cases can occur :

- $I \models \psi$, then $\mathcal{I}' = \mathcal{I}' \cup I$
- $I \not\models \psi$, then, $I \models \neg sbp_\sigma$, $\exists c = (x_i, y_i) \in \sigma$ where c is universal and I is of the form $(x_1, \dots, x_i, \neg y_i, \dots)$. As $\mathcal{I}$ is a total policy of $\psi \vee \neg sbp_\sigma$ then $\exists I' \in \mathcal{I}$ of the form $(x_1, \dots, \neg x_i, y_i, \dots)$ such that $I' \models \psi \vee \neg sbp_\sigma$ and $I' \not\models \neg sbp_\sigma$. Consequently, $I' \models \psi$ and $\sigma(I') \models \psi$ where $\sigma(I')$ is of the form $(x_1, \dots, x_i, \neg y_i, \dots)$. Then $\mathcal{I}' = \mathcal{I}' \cup \sigma(I')$.

By construction, it is clear that $\mathcal{I}'$ is a policy of Φ.

Example 3 *Let $\Phi = \forall x_1 \exists x_2 y_2\ (x_1 \vee x_2 \vee \neg y_2) \wedge (x_1 \vee \neg x_2 \vee y_2) \wedge (\neg x_1 \vee \neg x_2 \vee y_2) \wedge (\neg x_1 \vee x_2 \vee \neg y_2)$
Φ has a symmetry $\sigma = \{(x_1, \neg x_1)\}$. As $\{(x_1, \neg x_1)\}$ is a universal cycle, and $sbp_\sigma = (\neg x_1)$, we have:
$\Phi' = \Phi \vee \neg sbp_\sigma = \Phi \vee x_1$. Translating Φ' to clausal form we obtain : $\Phi' = \forall x_1 \exists x_2 y_2\ (x_1 \vee x_2 \vee \neg y_2) \wedge (x_1 \vee \neg x_2 \vee y_2)$*

The problem arises from symmetry with existential and universal cycles. the new formula $\Phi' = QX(\psi \vee \neg sbp_\sigma)$ described in property 2 is not equivalent to Φ wrt. validity. Let us consider a counter example. Let $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$ be a symmetry such that $\sigma \uparrow X_1 = (x_1, y_1)$ and $Q_1 = \exists$. If we start by assigning x_1 to *true* and y_1 to $false$, the sbp_σ becomes $false$, and $(\psi \vee \neg sbp_\sigma)$ is valid. This conclusion is wrong because it means that all formulae Φ which have a symmetry $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$ such that $\sigma \uparrow X_1 = (x_1, y_1)$ and $Q_1 = \exists$ are valid.

In conclusion, let σ a symmetry of Φ. If all cycles of σ are existentially quantified then Φ and $(\Phi \wedge sbp_\sigma)$ are equivalent and if all cycles of σ are universally quantified then Φ and $(\Phi \vee \neg sbp_\sigma)$ are equivalent. To deal with symmetries containing both universally and existentially cycles, one need to alternate sub parts of $(\Phi \wedge sbp_\sigma)$ and $(\Phi \wedge sbp_\sigma)$.

Definition 4 *Let $\Phi = Q_1X_1 \dots Q_mX_m\psi$ be a QBF and $\sigma = \{\sigma_1 \dots \sigma_m\}$ a symmetry of Φ. We define $sbp_\sigma \uparrow X_i$ the projection of sbp_σ on the set X_i*

Example 4 *Let $\Phi = \forall x_1y_1x_2y_2\exists x_3y_3\psi$ be a QBF formula such that $\sigma = \{(x_1, y_1)(x_2, y_2)(x_3, y_3)\}$ a symmetry of Φ.*

$$sbp_\sigma \uparrow X_1 = \quad x_1 \leq y_1$$
$$(x_1 = y_1) \rightarrow x_2 \leq y_2$$
$$sbp_\sigma \uparrow X_2 = \quad (x_1 = y_1) \wedge (x_2 = y_2) \rightarrow x_3 \leq y_3$$

Using the previous definitions, we can now describe the general formulation of symmetry breaking for quantified boolean formulae as follows:

Property 3 *Let $\Phi = Q_1X_1 \dots \exists X_m\psi$ be a QBF and $\sigma = \{\sigma_1 \dots \sigma_m\}$ a symmetry of Φ.*
Φ and Φ' are equivalent with respect to validity where $\Phi' = Q_1X_1 \dots \exists X_m(\dots(((\psi \wedge [sbp_\sigma \uparrow X_m]) \vee \neg[sbp_\sigma \uparrow X_{m-1}]) \wedge [sbp_\sigma \uparrow X_{m-2}]) \vee \dots)$

Proof : Suppose that Q_1 is universal.
We can restrict the symmetry breaking predicates to X_1. All cycles of $\sigma \uparrow X_1$ are universal. Then by property 2, Φ and $Q_1X_1 \dots Q_mX_m(\psi \vee \neg[sbp_\sigma \uparrow X_1])$ are equivalent. Let $\sigma_1 = (x_1, y_1) \dots ((x_p, y_p)$. and $\eta_1 = (x_1 = y_1) \wedge \dots \wedge (x_p = y_p)$. If $\eta_1 = true$ then $\{\sigma_2 \dots \sigma_m\}$ is a symmetry of ψ and also a symmetry of $QX(\psi \vee \neg[sbp_\sigma \uparrow X_1])$. By considering only σ_2, $QX\psi$ is equivalent to $QX(\psi \vee \neg[sbp_\sigma \uparrow X_1]) \wedge (\neg\eta_1 \vee sbp_{\sigma_2})$. From the definition 4, we can easily conclude that $sbp_\sigma \uparrow X_2 = (\neg\eta_1 \vee sbp_{\sigma_2})$.
Finally we have $QX\psi$ is equivalent to $(\psi \wedge [sbp_\sigma \uparrow X_2]) \vee (\neg[sbp_\sigma \uparrow X_1] \wedge [sbp_\sigma \uparrow X_2])$
As $(\neg[sbp_\sigma \uparrow X_1] \rightarrow [sbp_\sigma \uparrow X_2])$
$QX\psi$ and $QX(\psi \wedge [sbp_\sigma \uparrow X_2) \vee \neg[sbp_\sigma \uparrow X_1]$ are equivalent wrt validity. The proof follow by iterating the above process. Similarly the proof can be obtained when the first quantifier Q_1 is existential.

Let us note that in the case of existential symmetries σ_e, $\Phi' = QX(\psi \wedge spb_{\sigma_e})$ which is the formula of property 1. And in the case of totally universal symmetries σ_u $\Phi' = QX(\psi \vee \neg sbp_{\sigma_u})$ which is the formula associated to property 2. The following example illustrate how the formula Φ' is built.

Example 5 *Let us consider the QBF Φ of the example 2. $\sigma = \{(x_1, y_1)(x_2, y_2)(_3, y_3)\}$ is a symmetry of Φ. Φ is rewritten as :*
$\Phi' = \exists x_1y_1\forall x_2y_2\exists x_3y_3$
$((\psi \wedge [sbp_\sigma \uparrow X_3]) \vee \neg[sbp_\sigma \uparrow X_2]) \wedge [sbp_\sigma \uparrow X_1]$
where:
$sbp_\sigma \uparrow X_1 = x_1 \leq y_1$
$sbp_\sigma \uparrow X_2 = (x_1 = y_1) \rightarrow x_2 \leq y_2$
$sbp_\sigma \uparrow X_3 = (x_1 = y_1) \wedge (x_2 = y_2) \rightarrow x_3 \leq y_3$

This reasoning can easily be generalized to a set of symmetries $S = \{\sigma^1, \dots, \sigma^k\}$ by considering projections of all symmetries of S on X_i.

Let us note that the formula Φ' is not in prenex clausal form. As most of the available QBF solvers take as input a QBF formula in prenex clausal form, one need to translate Φ' to this most used standard format. For the other few solvers using general representation ([Zhang, 2006; Sabharwal *et al.*, 2006]) this translation is not necessary. Unfortunately, this transformation could be time and space consuming. To overcome this problem, we propose in the next section an linear time transformation approach.

3.3 Breaking Symmetries : clausal form

Let $\Phi = QX\psi$ be a quantified boolean formula and $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$ a symmetry of Φ. Suppose $(x_1 = y_1)$ and $(x_2 = y_2)$ $\dots (x_{i-1} = y_{i-1})$, our proposed transformation distinguish the two following cases :

1. Qx_i, $Q = \exists$: if $x_i = true$ then y_i has to be assigned to *true*
2. Qx_i, $Q = \forall$: $x_i = true$ and $y_i = false$, then we must satisfy ψ under this interpretation.

By considering all the symmetry σ, there exists a set of interpretations which we want to consider as models of ψ. These expected models are collected in a set called I_σ defined in the following definition.

Definition 5 *Let $\Phi = QX\psi$ be a quantified boolean formula and $\sigma = (x_1, y_1), \dots, (x_n, y_n)$ a symmetry of Φ. We define I_σ as the set of following interpretations:*
$I_\sigma = \bigcup_{i=1..n}\{((x_1 = y_1), \dots, (x_{i-1} = y_{i-1}), x_i, \neg y_i)) \mid$ *such that x_i is universal*$\}$

In order to break symmetries of Φ, we want to satisfy the above two cases, i.e. propagate existential literals thanks to sbp_σ and consider all interpretations of I_σ as models of ψ. To this end, we introduce a new existential variable r_σ added to the innermost quantifier group and we rewrite ψ as follows: $(\psi \vee \neg r_\sigma) \wedge f_\sigma(r_\sigma, sbp_\sigma)$. The function f_σ, allows us to break the symmetry σ. It forces r_σ to be $false$ for all interpretations belonging to I_σ and *true* otherwise. Furthermore, it induces propagations for existential cycles.

Definition 6 *Let $\sigma = \{(x_1, y_1) \dots (x_{lu}, y_{lu}) \dots (x_n, y_n)\}$ be a symmetry of*

Φ such that $\forall i > lu$, (x_i, y_i) is an existential cycle (lu the rank of the last universal cycle in σ). The function $f_\sigma(r, sbp_\sigma)$ is defined as follow:

1. *if $0 < i < lu$ and $(x_1 = y_1) \wedge \ldots (x_{i-1} = y_{i-1})$ then:*
   - *if (x_i, y_i) is a universal cycle:* $x_i \wedge \neg y_i \rightarrow \neg r$ *and* $\neg x_i \wedge y_i \rightarrow r$
   - *else* $x_i \rightarrow y_i$ *and* $\neg x_i \wedge y_i \rightarrow r$
2. *if $i = lu$ and $(x_1 = y_1) \wedge \ldots (x_{i-1} = y_{i-1})$ then :* $x_{lu} \wedge \neg y_{lu} \rightarrow \neg r$ *and* $\neg x_{lu} \rightarrow r$ *and* $(x_{lu} = y_{lu}) \rightarrow r$
3. *if $i > lu$ and $(x_1 = y_1) \wedge \ldots (x_{i-1} = y_{i-1})$ then :* $x_i \rightarrow y_i$

Example 6 *Let $\Phi = \exists x_1, y_1 \forall x_2, y_2 \exists x_3, y_3 \psi$ be a QBF formula and $\sigma = \{(x_1, y_1), (x_2, y_2)\}$ a symmetry of Φ. Using the formula $f_\sigma(r_\sigma, sbp_\sigma)$, the different values taken by r_σ according to the different interpretations of x_1, x_2, y_1, y_2 are depicted in Figure 1.*

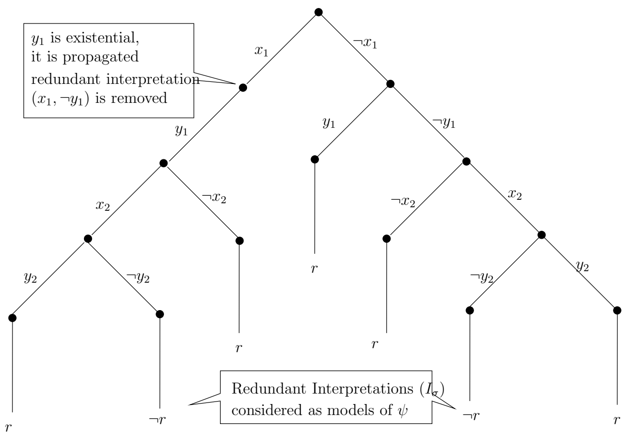

Figure 1: $f_\sigma(r_\sigma, sbp_\sigma)$: a search tree

Finally, we have the following property.

Property 4 *Let $\Phi = QX\psi$ be a QBF and σ a symmetry of Φ, then Φ and $\Phi^r = QX\exists r\ (\psi \vee \neg r) \wedge f_\sigma(r, sbp_\sigma)$ are equivalent wrt. validity.*

Example 7 *Let us consider the QBF $\Phi = \forall x_1, y_1, x_2, y_2 \exists x_3, y_3 \psi$ and $\sigma = \{(x_1, y_1)(x_2, y_2)\}$ a symmetry of Φ.*

$$\Phi^r = \forall x_1, y_1, x_2, y_2 \exists x_3, y_3 \alpha_1, \alpha_2, r$$
$$\begin{array}{lll}
(\psi \vee \neg r) & \wedge & \\
(\neg\alpha_1 \vee x_2 \vee y_2 \vee \alpha_2) & \wedge & (x_1 \vee y_1 \vee \alpha_1) \wedge \\
(\neg\alpha_1 \vee \neg x_2 \vee \neg y_2 \vee \alpha_2) & \wedge & (\neg x_1 \vee \neg y_1 \vee \alpha_1) \wedge \\
(\neg\alpha_1 \vee \neg x_1 \vee \neg x_2 \vee y_2 \vee \neg r) & \wedge & (\neg x_1 \vee y_1 \vee \neg r) \wedge \\
(\neg\alpha_1 \vee y_1 \vee \neg x_2 \vee y_2 \vee \neg r) & \wedge & (x_1 \vee \neg y_1 \vee r) \wedge \\
(\neg\alpha_1 \vee x_2 \vee \neg r) & \wedge & (\neg\alpha_1 \vee \neg\alpha_2 \vee r)
\end{array}$$

The variables α_1 (resp. α_2) is introduced to expresses that $(x_1 = y_1)$ (resp. $(x_1 = y_1) \wedge (x_2 = y_2)$). The last 10 clauses are those of $f_\sigma(r, sbp_\sigma)$.

In the following, we show how to generalise the previous transformation to an arbitrary set of symmetries. Let Φ a QBF and $S = \{\sigma^1 \ldots \sigma^n\}$ the set of symmetries of Φ. For each universal symmetry σ^i, we introduce an additional variable r^i_σ to break it. For the set S, the equivalent formula is obtained by adding all the symmetries of Φ as described in the following definition.

Definition 7 *Let Φ be a QBF and S a set of symmetries of Φ and $U = \{\sigma^1 \ldots \sigma^p\}$ the subset of S containing only universal symmetries. we define*

$$\Phi^r = QX \exists r^1_\sigma r^2_\sigma \ldots r^p_\sigma r(\psi \vee \neg r) \wedge f_U(r, SBP_U) \wedge SBP_\exists$$

where

- $r = \wedge(r^1_\sigma, r^2_\sigma \ldots r^p_\sigma)$
- $f_U(r, SBP_U) = \bigwedge_{\sigma^i \in U} f_{\sigma^i}(r^i_\sigma, sbp_{\sigma^i})$
- *$SBP_\exists$ (resp. SBP_U) is the SBP corresponding to all existential (resp. universal) symmetries.*

Property 5 *Let Φ a QBF formula and S the set of all its symmetries, then Φ and Φ^r are equivalent with respect to validity.*

In the sequel, we propose some useful simplification of our proposed transformation that we use in our experimental evaluation.

Let $\Phi = Q_1X_1 \ldots Q_mX_m$ be a QBF and $U = \{\sigma^1 \ldots \sigma^p\}$ be the set of universal symmetries of Φ such that $Var(U) \in \{X_i \cup \cdots \cup X_n\}$. Let c be a clause of ψ such that $Var(c) \in \{X_1 \cup \cdots \cup X_{i-1}\}$. then it is not necessary to add $\neg r$ in c. Indeed, if $\Phi_{X_1 \ldots X_{i-1}}$ is the QBF obtained after the assignment of all the variables in $\{X_1 \cup \cdots \cup X_{i-1}\}$, then σ remains a symmetry of $\Phi_{X_1 \ldots X_{i-1}}$.

Obviously, the new built formula allows us to avoid some isomorphic interpretations. However, adding $\neg r$ to all clauses of ψ eliminates some useful propagation. For example, if $(x \vee y)$ is a clause such that neither x and y belong to the symmetries of Φ then $(x \vee y \vee \neg r)$ is the new clause of Φ^r. When x is assigned to $false$, then y is propagated in ψ, which is not the case in Φ^r. Furthermore, in the worst case, if y is assigned to $false$, $\neg r$ is propagated and the solver will lose time in checking $f_U(r, sbp_U)$ i.e. to prove the inconsistency induced by the sbp.
Not surprisingly, in practice adding $\neg r$ in all clauses of ψ make generally Φ^r more difficult to be solved than Φ. To avoid such a case, the goal is to add $\neg r$ just in some chosen clauses and leave the others unchanged.

For our experiments, as binary clauses might be helpful during search, we add $\neg r$ only in non binary clauses, containing at least one literal appearing in a universal symmetry.

Let $\Phi = Q_1X_1 \dots Q_mX_m$ be a QBF and $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$ a symmetry of Φ. Suppose that exists a cycle (x_p, y_p) of σ such that x_p and y_p are pure literals. In this case, if we generate the *sbp* of σ, then the literals x_p and y_p are not pure. As the pure literals play an important role in QBF solving [Giunchiglia *et al.*, 2004], this might have a great impact on the efficiency of the QBF solvers. To avoid this problem, we propose the two following possibilities. The first one, consists in eliminating the cycle (x_p, y_p) from σ and propagating x_p and y_p following theirs quantifiers. The second one consists in reducing σ to the set of cycles $\{(x_1, y_1) \dots (x_{p-1}, y_{p-1})\}$. For our experiments, we choose this second option.

Finally, if σ is of the form $(x, \neg x)$ where x is a universal literal, then x is requantified existentially and the clauses $c = (\neg x)$ is added to the matrix.

Complexity

Adding new existentially quantified variables, the size of the classical SBP is known to be linear in the size of symmetry. According to the formula of $f_\sigma(r_\sigma, sbp_\sigma)$, its size remain linear in the size of the symmetry σ. In the worst case, for a totally universal symmetry $\sigma = \{(x_1, y_1) \dots (x_n, y_n)\}$, we introduce one variable and four clauses for each cycle. Consequently, the size of $f_\sigma(r_\sigma, sbp_\sigma)$ is in $O(5n)$.

4 Experiments

The experimental results reported in this section are obtained on a Xeon 3.2 GHz (2 GB RAM) and performed on a large panel of symmetric instances available from [Giunchiglia *et al.*, 2001a]. This set of QBF instances contains different families like `toilet`, `k_*`, `FPGA`, `qshifter`. As a comparison, we run the state-of-the-art DPLL-like solver SEMPROP [Letz, 2002] on QBF instances with and without breaking symmetries. The time limit is fixed to 900 seconds. Results are reported in seconds. The symmetry computation time is not reported (less than one second in most cases). U indicates if the instances contain universal symmetries, NB_S represents the number of symmetries, SAT indicates if the instances is valid or not. Φ represents the time needed to solve the instances without breaking symmetries and Φ^r the time to solve the instances after breaking symmetries with the following restrictions. First, in a case of existential symmetry, we keep only the first cycle. Second, in a case of universal symmetry, we cut the symmetries from the last universal cycle.

Table 1 gives results on a large variety of QBF instances. Breaking symmetries with our method significantly improves SEMPROP performances on many QBF instances leading to more solved instances. As an example, for the `biu_*` family, without breaking symmetries, SEMPROP solves one instance whereas adding symmetry breaking predicates, we are able to solve 8 instances. For `C5135_*` family we can solve one instance. On `toilet_c*` family the cpu time decreases in more than one order of magnitude when using symmetry breaking predicates.

Furthermore, the existence of universal symmetries seems to be an important factor for reducing the search time, especially when these symmetries occurs in the outermost quantifier. It is the case for example for the `biu` family.

Not surprisingly, we have also noticed that symmetries between literals occurring in the innermost quantifier (recall that it is existential) group are useless. Indeed, such symmetries does not lead to a great reduction in the search tree, since their corresponding variables are assigned last i.e when the formula is considerably reduced by the previous assignments. Finally, for QBF families containing only existential symmetries, breaking them does not lead to a great improvement in the search time exept in some cases (see for example families (`k_grz`), `toilet_a`).

5 Comparison between our encoding and sbp4qbf algorithm

Let us now compare our method with `sbp4qbf` [Audemard *et al.*, 2007].

To break symmetries, the authors of `sbp4qbf` propose a new order of the quantifier. For example, let Φ be a QBF and $\{(x_1, y_1)\}$ a symmetry of Φ such that, $Var(X_1) = \{(x_1, y_1)\}$, and $Q_1 = \forall$. The equivalent *sbp* is $x_1 \leq y_1$. So, if x_1 is assigned to $true$, then we can directly assigny_1 to $true$ and if y_1 is assigned with $false$, x_1 can be assigned to $false$. From this algorithm, the new prefix is rewritten as $\forall x_1 \exists \alpha_1 \forall y'_1 \exists y_1 Q_2 X_2 \dots Q_m X_m \; \psi \wedge sbp \wedge qsbp$ where $qsbp$, are the added predicates to preserve the equivalence. This new prefix constrains x_1 to be assigned before y_1. With our method no restriction is needed. The only modification of the prefix concern the set of existential variables added to the innermost quantifier.

In `sbp4qbf`, when there is two universal symmetries like $\sigma^1 = \{r_1, (x_1, y_1) \dots\}$ and $\sigma^2 = \{r_2, (z_1, -y_1) \dots\}$ such that y_1 is a universal literal. To keep the equivalence, σ^2 is reduced to $\sigma'^2 = \{r_2\}$ and a part of the symmetry σ^2 is removed. It is not the case here with our proposed method.

Obviously, the main difference between `sbp4qbf` and our method is in the way used to manage the universal literals. Indeed, considers a simple universal symmetry $\{(x_1, y_1)\}$, if x_1 is assigned to $true$, y_1 is propagated in `sbp4qbf`, where with our method, $\{x_1, \neg y_1\}$ is considered as a model which is an other alternative to the algorithm proposed in [Audemard *et al.*, 2007].

At the end, we compared experimentally our results with the ones obtained in [Audemard *et al.*, 2007]. Table 2 provides more detailed results on the different QBF families. The second column (NB) represents the number of instances in each family. The third column (U) indicates if the instances contain universal symmetries (Y) or not (N). For each family, S and TT represents the total number of solved instances and the total run-time needed for solving all the instances (900 seconds are added for each unsolved one) respectively and finally Φ^S represent the broken formula using the algorithm described in [Audemard *et al.*, 2007]. In general case, `sbp4qbf` seems to be more efficient than our method but it exists instances, where our method outperforms `sbp4qbf`.

Instances	U	NB_S	SAT	Φ^r	Φ
biu.mv.xl_ao-p005-IPF02-c02	Y	15	F	–	**1.45**
biu.mv.xl_ao-p005-OPF03-c09	Y	15	T	**17.83**	–
biu.mv.xl_ao-p010-IPF02-c07	Y	15	T	**1.25**	–
biu.mv.xl_ao-p010-IPF02-c08	Y	15	T	**0.83**	–
biu.mv.xl_ao-p010-IPF05-c05	Y	94	T	**0.15**	–
biu.mv.xl_ao-p010-IPF05-c07	Y	28	T	**16.72**	–
biu.mv.xl_ao-p010-IPF05-c10	Y	91	T	**0.16**	–
biu.mv.xl_ao-p010-OPF02-c08	Y	91	T	**1.36**	–
biu.mv.xl_ao-p010-OPF02-c10	Y	91	T	**1.35**	–
C5315.blif_0.10_1.00_0_1_out_exact	Y	37	T	**788.41**	–
k_path_p-10	Y	4	F	**18.73**	39.48
k_path_p-11	Y	4	F	**20.35**	71.39
k_path_p-12	Y	7	F	**130.25**	357.70
k_path_p-13	Y	8	F	**203.14**	643.14
k_path_p-14	Y	8	F	–	**39.64**
k_path_p-14	Y	8	F	–	**39.64**
k_path_n-19.qdimacs	Y	11	T	–	**579.72**
ncf_8_16_4_u.4	Y	1	F	**168.39**	176.67
ncf_8_16_4_u.9	Y	1	F	**313.67**	336.21
ncf_8_16_8_d.2	Y	1	F	**121.71**	129.42
ncf_8_16_8_d.4	Y	1	F	**168.28**	180.17
ncf_8_16_8_d.6	Y	1	F	**157.22**	169.51
ncf_4_16_4_d.6	Y	1	F	19.28	**3.06**
ncf_16_32_4_u.9	Y	1	T	70.45	**61.36**
ncf_4_16_2_u.8	Y	2	T	90.19	**1.69**
ncf_8_16_8_edau.3	Y	1	F	**305.25**	318.73
ncf_8_16_8_edau.6	Y	1	T	**455.41**	479.48
ncf_8_16_8_euad.7	Y	1	T	**669.75**	717.30
lut4_3_fAND	Y	11	T	**0.28**	26.74
lut4_AND_fXOR	Y	11	F	**423.40**	–
lut4_2_fXOR	Y	9	T	4.51	**0.07**
ken.flash08.C-f4	Y	1	F	438.24	**8.26**
term1.blif_0.10_0.20_0_0_inp_exact	Y	3	F	**151.88**	160.19
TOILET7.1.iv.13	Y	2	F	**12.82**	70.37
TOILET6.1.iv.11	Y	2	F	**1.28**	4.39
TOILET7.1.iv.14	Y	2	T	11.83	**7.07**
toilet_c_10_01.15	Y	2	F	**2.48**	12.05
toilet_c_10_01.16	Y	2	F	**7.47**	48.31
toilet_c_10_01.17	Y	2	F	**23.75**	158.30
toilet_c_10_01.18	Y	2	F	**77.86**	570.81
toilet_c_10_01.19	Y	2	F	**259.48**	–
toilet_c_10_01.20	Y	2	T	**1.87**	–
S-adeu-26	Y	46	F	**8.49**	14.69
S-adeu-37	Y	46	F	**1.21**	3.32
S-adeu-40	Y	46	F	**7.51**	12.53
S-edau-26	Y	46	F	**4.17**	6.23
S-edau-43	Y	46	F	1.66	2.71
S-adeu-23	Y	48	F	37.77	**4.85**
S-adeu-40	Y	46	F	**7.51**	12.53
T-adeu-14	Y	46	F	**7.47**	25.70
T-adeu-26	Y	46	F	**2.04**	4.08
T-adeu-40	Y	46	F	1.65	2.97
T-edau-14	Y	46	F	**2.97**	6.65
x40.11	N	2	T	**687.46**	728.11
k_branch_n-9	N	1	T	**291.31**	292.66
k_branch_p-10	N	1	F	**170.64**	182.57
k_branch_p-11	N	1	F	331.90	334.92
k_branch_p-12	N	1	F	**192.48**	206.89
k_branch_p-9	N	1	F	**38.88**	41.47
k_poly_p-20	N	2	F	**314.37**	334.85
k_poly_p-21	N	2	F	348.21	349.71
k_poly_p-9	N	2	F	21.59	22.92
k_ph_n-10	N	1	T	**651.07**	–
k_grz_n-10	N	6	T	**148.00**	161.22
k_grz_n-12	N	6	T	**729.89**	874.60
k_grz_n-13	N	6	T	**360.63**	425.38
k_grz_n-15	N	8	T	**325.21**	338.32
k_grz_n-16	N	8	T	**298.36**	326.51
k_grz_n-17	N	8	T	**255.17**	263.69
k_grz_p-10	N	4	F	**153.96**	164.11
k_grz_p-11	N	4	F	**79.35**	84.79
k_grz_p-12	N	4	F	**191.12**	205.59

Table 1: SEMPROP with and without symmetries breaking

			Φ		Φ^S		Φ^r	
family	NB	U	S	TT	S	TT	S	TT
fpga	8	Y	6	1834	**7**	**921**	6	2231.71
biu	23	Y	1	19801	**8**	13668	**8**	**13539**
toilet_c	53	Y	51	2656	**53**	**49**	53	374.61
C5315	4	Y	0	3600	0	3600	1	**3488**
k_path	40	Y	28	12915	**34**	**7999**	24	15001.79
qshifter	6	Y	6	67	6	61	6	**40**
TOILET	7	Y	6	988	6	**926**	6	**926**
term1	6	Y	6	**174**	6	177	6	**164.13**
strategic	100	N	86	13482	86	**13477**	86	**13477**
k_branch	42	N	21	20096	21	**20069**	21	**20069**
k_lin	21	N	5	14553	5	14551	5	14551
k_grz	37	N	23	15242	**24**	**14997**	**24**	**14997**
k_poly	42	N	42	**2031**	42	2068	42	2068
toilet_a	22	N	22	**12**	22	20	22	20
TOTAL	411		303	107420	**320**	**92583**	310	100945

Table 2: Results on different QBF families

6 Conclusion

In this paper, a new approach to break symmetries in quantified boolean formulae is proposed. To preserve the equivalence, this new formula allows to propagate the existential literals from the *sbp* and remove universal symmetrical literals by introducing an auxiliary variable and considering some interpretations as models of the original formula. Experiments show the interest behind breaking these symmetries. Important improvement are obtained on different QBF families. As future work, we want to use our theoretical approach on non clausal solvers.

References

[Aloul *et al.*, 2002] F. Aloul, A. Ramani, I. Markov, and K. Sakallah. Solving difficult instances of boolean satisfiability in the presence of symmetry. Technical report, University of Michigan, 2002.

[Audemard *et al.*, 2004] G. Audemard, B. Mazure, and L. Sais. Dealing with symmetries in quantified boolean formulas. In *proceedings of SAT*, pages 257–262, 2004.

[Audemard *et al.*, 2007] G. Audemard, S. Jabbour, and L. Sais. Symmetry breaking for quantified boolean formulas. In *proceedings of IJCAI*, pages 2262–2267, 2007.

[Benedetti, 2005a] M. Benedetti. sKizzo: a Suite to Evaluate and Certify QBFs. In *Proc. of CADE*, pages 369–376, 2005.

[Benedetti, 2005b] Marco Benedetti. Extracting certificates from quantified boolean formulas. In *IJCAI*, pages 47–53, 2005.

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

[Biere, 2004] Armin Biere. Resolve and expand. In *proceedings of SAT (Selected Papers)*, pages 59–70, 2004.

[Cohen *et al.*, 2005] David A. Cohen, Peter Jeavons, Christopher Jefferson, Karen E. Petrie, and Barbara M. Smith. Symmetry definitions for constraint satisfaction problems. In *proceedings of CP*, pages 17–31, 2005.

[Coste-Marquis *et al.*, 2006] Sylvie Coste-Marquis, Hélène Fargier, Jérôme Lang, Daniel Le Berre, and Pierre Marquis. Representing policies for quantified boolean formulae. In *proceedings of KR*, pages 286–297, 2006.

[Crawford *et al.*, 1996] J. Crawford, M. Ginsberg, E. Luck, and A. Roy. Symmetry-breaking predicates for search problems. In *proceedings of KR*, pages 148–159. 1996.

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

[Gent and Smith, 2000] I. Gent and B. Smith. Symmetry breaking in constraint programming. In *Proceedings of ECAI*, pages 599–603, 2000.

[Giunchiglia *et al.*, 2001a] E. Giunchiglia, M. Narizzano, and A. Tacchella. Quantified Boolean Formulas satisfiability library (QBFLIB), 2001. `http://www.qbflib.org`.

[Giunchiglia *et al.*, 2001b] E. Giunchiglia, M. Narizzano, and A. Tacchella. QuBE : A system for deciding Quantified Boolean Formulas Satisfiability. In *Proceedings of IJCAR*, pages 364–369, 2001.

[Giunchiglia *et al.*, 2004] Enrico Giunchiglia, Masimo Narizzano, and Armando Tachella. Monotone literals and learning in qbf reasonning. In *Proceedings of CP*, pages 260–273, 2004.

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

[McKay, 1990] B. McKay. nauty user's guide (version 1.5). Technical report, 1990.

[Puget, 1993] J.F. Puget. On the satisfiability of symmetrical constraint satisfaction problems. In *proceedings of ISMIS*, pages 350–361, 1993.

[Sabharwal *et al.*, 2006] Ashish Sabharwal, Carlos Ansótegui, Carla P. Gomes, Justin W. Hart, and Bart Selman. Qbf modeling: Exploiting player symmetry for simplicity and efficiency. In *in proceedings of SAT*, pages 382–395, 2006.

[Zhang and Malik, 2002] L. Zhang and S. Malik. Towards a symmetric treatment of satisfaction and conflicts in quantified boolean formula evaluation. In *Proceedings of the CP*, pages 200–215, 2002.

[Zhang, 2006] Lintao Zhang. Solving QBF by combining conjunctive and disjunctive normal forms. In *in proceedings of AAAI*, 2006.