Refinement-Based Enumeration of QBF Solutions

Andreas Plank[0000−0002−2653−0689], Clemens Hofstadler[0000−0002−3025−0604], Maximilian Heisinger[0000−0001−7297−6000], and Martina Seidl[0000−0002−3267−4494]

Institute for Symbolic Artificial Intelligence, Johannes Kepler University, Linz, Austria
{andreas.plank, clemens.hofstadler, maximilian.heisinger, martina.seidl}@jku.at

Abstract. Counting the number of solutions of true and false quantified Boolean formulas (QBFs) has received increased interest in recent years. However, the explicit enumeration of all solutions for a QBF is almost unexplored so far. For QBF solution counting, it has been shown that enumeration-based counting as employed in SAT is not complete for QBF. Solution enumeration runs into the same problem.
We propose a refinement-based approach to enumerate (all) solutions of true and false QBFs. To this end, we develop a novel framework to characterize the enumeration problem at quantifier level two and present a complete enumeration algorithm that can be interrupted at any time as soon as enough solutions are found. We evaluated our implementation called QEnum in three different case studies.

1 Introduction

Encoding and solving problems using quantified Boolean formulas (QBFs) offers a uniform method to address challenges across various application fields such as formal verification or artificial intelligence (see [19] for a survey). The QBF decision problem is the archetypical PSPACE-complete problem that has been well investigated over the last decades [3]. For QBF encodings of application problems it is important to know not only their truth value but also their solutions. The problem of finding all solutions of a QBF with free variables was already introduced in [2]. There, an approach was presented to find assignments to the free variables (variables which are not bound by a quantifier) such that the resulting QBF is true. In this work, we also consider the task of enumerating all solutions but in a more general setting. Let's consider a true QBF of the form $\Phi = \forall X \exists Y.\phi$. A model F of this formula is a set of propositional formulas such that for each $y \in Y$, there is a formula $f_y \in F$ over X and the propositional formula ϕ' obtained by replacing each y_i by its f_{y_i} is valid. Dually, let $\Psi = \exists X \forall Y.\phi$ be a false QBF. A counter-model G of Ψ is a set of propositional formulas over the variables X. Then there is a formula $g_y \in G$ for each $y \in Y$ such that the propositional formula ψ', which is obtained by replacing each y by g_y, is unsatisfiable. We are interested in enumerating all models of a true QBF and all counter-models of a false QBF.

A problem closely related to solution enumeration is the problem of solution counting also known as #QBF [10]. Contrary to #SAT [6] (counting models for propositional formulas), which has been a prominent research area in many different fields such as analysis of software vulnerability [4, 20], verification of neural networks [1, 13] and probabilistic reasoning [5, 16], the first practical solution counters for QBFs have been presented only in recent years [11, 18, 15]. An enumerative method for counting (counter-)models was introduced in [15]: Given a QBF Φ, a QBF solver is called to compute a solution F_1. Next, a new QBF Φ_1 is constructed from Φ such that F_1 is no longer a solution. Then, a solution F_2 of Φ_1 is computed. In QBF Φ_2, F_1 and F_2 are both excluded from the solution space. More solutions are enumerated and excluded until finally Φ_n does not have the same truth value as Φ anymore. While, in propositional logic, the exact model count is obtained in this way, this approach is not complete for QBFs. To compute the full model count of a true QBF, a SAT solver as well as a propositional model counter is used in [15]. While conceptually dual, this approach does not work for counting all solutions of false QBFs.

We present a novel approach that enumerates solutions at the second quantifier level. In contrast to previous work [15], the new procedure (1) works for true and false QBFs in a dual manner and (2) can be interrupted at any time as soon as enough solutions are found. The approach is implemented in the tool QEnum and evaluated in three case studies.

2 Preliminaries

Let $X_1, \ldots, X_n$ be pairwise disjoint non-empty sets of propositional variables. We consider *quantified Boolean formulas (QBFs)* of the form $\Phi = \Pi.\phi$, which consist of a *quantifier prefix* $\Pi = Q_1 X_1 \ldots Q_n X_n$ with *quantifiers* $Q_1, \ldots, Q_n \in \{\exists, \forall\}$, $Q_i \neq Q_{i+1}$ for $1 \leq i < n$ and of a (propositional) *matrix* ϕ. The matrix ϕ is a propositional formula over the variables $X_1 \cup \cdots \cup X_n$, the truth constants $\top$ (true) and $\bot$ (false), and the standard Boolean connectives. A variable $x \in X_i$ is called *existential* in Φ if $Q_i = \exists$ and *universal* otherwise.

We only consider *closed* formulas in this work, i.e., all variables occurring the propositional matrix ϕ also occur in the quantifier prefix Π. A QBF $\Pi.\phi$ is in *prenex conjunctive normal form (PCNF)* if ϕ is a conjunction of clauses, where a clause is a disjunction of literals and a literal is a variable or its negation. An *assignment* σ is a function $\sigma\colon X \to \{\top, \bot\}$ that maps a subset of the variables $X \subseteq X_1 \cup \cdots \cup X_n$ to truth values. It is called a *partial (X-)*assignment if X is a strict subset of $X_1 \cup \cdots \cup X_n$ and a *full* assignment otherwise.

For a propositional formula ϕ and an assignment σ, we denote by $\phi(\sigma)$ the formula obtained from ϕ by setting all variables in the domain of σ to their truth values as specified by σ. We call $\phi(\sigma)$ the *evaluation* of ϕ under σ. Thus, for example, if $\sigma = \{x \mapsto \top, y \mapsto \bot\}$, then $\phi(\sigma)$ denotes the propositional formula obtained from ϕ by setting the variable x to $\top$ and y to $\bot$. If $\phi(\sigma)$ simplifies to $\top$ (using the classical equivalences of Boolean algebra), we say that ϕ is *true* (or *satisfiable*) and that σ is a *model* (or a *satisfying assignment*) for ϕ.

The semantics of QBF is defined as follows: a QBF $\forall x \Pi.\phi$ is *true* if and only if both $\Pi.\phi(\{x \mapsto \top\})$ and $\Pi.\phi(\{x \mapsto \bot\})$ are true, and $\exists x \Pi.\phi$ is *true* if and only if $\Pi.\phi(\{x \mapsto \top\})$ or $\Pi.\phi(\{x \mapsto \bot\})$ is true. A QBF that is not true is *false*. For example, the QBF $\forall x \exists y.(x \leftrightarrow y)$ is true, while $\exists x \forall y.(x \leftrightarrow y)$ is false.

The notion of a model for a true QBF lifts the notion of a satisfying assignment to the quantified setting. Instead of mapping variables to truth constants, existential variables are now mapped to propositional formulas in (some of) the universal variables. Formally, a *model* for a true QBF $\Pi.\phi$ containing existential variables $y_1, \ldots, y_k$ is a set $F = \{f_{y_1}, \ldots, f_{y_k}\}$, where each f_{y_i} is a propositional formula in the universal variables that appear to the left of y_i in the prefix Π, such that the propositional formula obtained from ϕ by replacing each y_i by f_{y_i} simplifies to $\top$. For example, the set $F = \{f_y = x\}$ is a model for the true QBF $\forall x \exists y.(x \leftrightarrow y)$, because replacing y by $f_y = x$ yields $x \leftrightarrow x \equiv \top$. A model F for a true QBF is also called a *Skolem set* and the elements of F are called *Skolem functions*. Counter-models for false QBFs are defined dually: a set $H = \{h_{x_1}, \ldots, h_{x_l}\}$ of propositional formulas is a *counter-model* for a false QBF $\Pi.\phi$ with universal variables $x_1, \ldots, x_l$ if the propositional formula obtained from ϕ by replacing each x_i with h_{x_i} simplifies to $\bot$. Here, each h_{x_i} is a formula in the existential variables that appear to the left of x_i in the prefix Π. Counter-models are also called *Herbrand sets* and their elements are *Herbrand functions*.

Models and counter-models can be visualized as trees. Let $m = |X_1 \cup \cdots \cup X_n|$. Then a model F for a true QBF $\Phi = \Pi.\phi$ can be considered as a tree of height $m+1$, where the nodes in the k-th level ($k \in \{1, \ldots, m\}$) correspond to the k-th variable x_k in the quantifier prefix Π (from left to right; assuming an arbitrary but fixed order of the variables within each set X_i). Each node on level k has two children if x_k universal in Φ and exactly one child if it is existential. In the former case, the two edges are labeled by $\bot$ and $\top$, respectively. In the latter case, the label of the single edge is given by the evaluation of the Skolem function $f_{x_k} \in F$ under the partial assignment induced by the unique path from the root to the considered edge. Since f_{x_k} only depends on the universal variables that appear to the left of x_k in Π, this evaluation yields either $\top$ or $\bot$. Any path from the root to a leaf of this tree corresponds to a full assignment σ such that $\phi(\sigma)$ simplifies to $\top$. For counter-models, the roles of the quantifiers are exchanged, i.e., now each node on level k has two children if x_k is existential and only one child if it is universal. In the latter case, the label of the single edge is given by the evaluation of the Herbrand function h_{x_k}. A path from the root to a leaf corresponds to an assignment σ such that $\phi(\sigma)$ simplifies to $\bot$.

3 Introductory Example

In the following example, we demonstrate our enumerative approach with a simple 4×4 Tic-Tac-Toe game. Enumerative approaches exclude already identified solutions. As models for QBFs are represented as sets of propositional formulas, models for quantified formulas are excluded from the search space by appending *blocking sets* to the formula. All concepts will be formally introduced below.

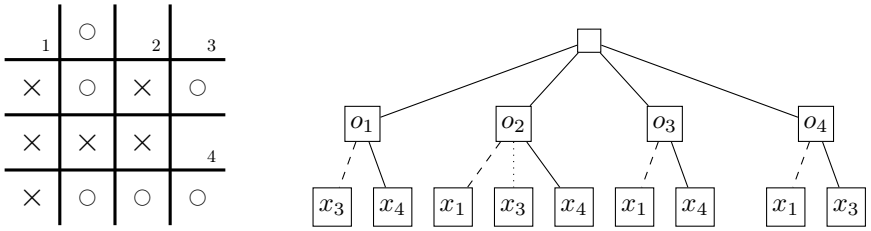


Fig. 1: Left: 4×4 Tic-Tac-Toe after 12 moves, with player ◯ to go. Right: Winning strategies for player ⨯, depending on the move of player ◯.

Example 1 (4×4 Tic-Tac-Toe). Consider a 4×4 Tic-Tac-Toe game after 12 moves have been played and it is now player ◯'s turn. The current state of the board is visualized in the left part of Figure 1. Our objective is to determine whether there exist one or more winning strategies for player ⨯ in their next move. If such strategies exist, we also aim to explicitly compute and enumerate all of them. This scenario can be encoded as the following QBF:

$$\forall O \exists X.\phi = \forall O \exists X.\text{valid-move}(O) \rightarrow (\text{valid-move}(X) \wedge \text{exclusive}(O, X) \wedge \text{win}(X)) \,.$$

Hereby, the universal variables $O = \{o_1, o_2, o_3, o_4\}$ represent the possible moves available to player ◯. In particular, the variable o_i encodes that player ◯ puts their mark on empty field i as shown in Figure 1. Analogously, we express the possible moves available to player ⨯ by the set of existential variables $X = \{x_1, x_2, x_3, x_4\}$. The predicate "valid-move" encodes that a player has to play precisely one move. The predicate "exclusive" ensures mutual exclusivity of the marks on the board, i.e., that at most one mark can be set on each position. Finally, the predicate "win(X)" represents the winning condition for player ⨯, specifying that their mark must be placed on either position 1,3, or 4 to win.

With the help of a QBF solver, one can check that the formula above is true. This confirms the existence of winning strategies for player ⨯. Moreover, from such a solver call, one can also extract a model of the QBF. This model, which we denote by F_1, encodes a winning strategy for player ⨯. For our example, say F_1 describes the strategy of picking the position with highest available index from the winning set $\{1, 3, 4\}$, excluding the position already marked by player ◯. This strategy is depicted with solid lines in Figure 1 (right).

To find further winning strategies, we exclude the strategy F_1 from our QBF search space, by adding it as a blocking set to the QBF. The updated formula $\forall O \exists X.(\phi \wedge \neg F_1)$ can be passed to a QBF solver again, returning a second strategy F_2. Say, strategy F_2 is defined by picking the position with lowest available index from the winning set $\{1, 3, 4\}$ (shown with dashed lines in Figure 1), excluding the position previously marked by player ◯. We have now identified all possible strategies to counter a move o_1, o_3, or o_4 by player ◯. Hence, the QBF $\forall O \exists X.(\phi \wedge \neg F_1 \wedge \neg F_2)$ is now false. However, if player ◯ puts their mark on position 2, there remains a third strategy, which involves selecting the middle index from the set $\{1, 3, 4\}$ (depicted with a dotted line in the right of Figure 1), a strategy

that has not been covered yet. We can enforce the QBF solver to identify this remaining strategy F_3 by excluding the other markings o_1, o_3, o_4 by disjoining them with the QBF obtaining $\forall O \exists X.\big((\phi \wedge \neg F_1 \wedge \neg F_2) \vee o_1 \vee o_3 \vee o_4\big)$. Note that F_3 is only a "partial" winning strategy, as it relies on the assumption that player ◯ makes a specific move. We will formally introduce and discuss such partial solutions in Section 5.

4 Disjoint Solutions for QBFs and Their Enumeration

In this section, we recall and summarize the results from [15] for enumerating *disjoint* solutions for QBFs and the notion of *blocking Skolem/Herbrand sets*.

Definition 1 (Blocking Skolem/Herbrand Set). *Let $\Phi = \Pi.\phi$ be a true QBF and let F be a Skolem set for Φ. Then $\neg\phi_F$ is a* blocking Skolem set *for Φ, where $\phi_F = \bigwedge_{f_y \in F}(y \leftrightarrow f_y)$. Dually, for a false QBF $\Phi = \Pi.\phi$ with Herbrand set H, ϕ_H is a* blocking Herbrand set *for Φ, where $\phi_H = \bigwedge_{h_x \in H}(x \leftrightarrow h_x)$.*

Note that blocking Skolem sets are negated, while blocking Herbrand sets are not. In [15], we showed that by incrementally adding blocking Skolem/Herbrand sets to a QBF until the formula changes its truth value, one can efficiently enumerate (counter-)models for a QBF. However, in this way, not all (counter-)models can be computed, but only all *disjoint* ones. For space reasons, we only state the following definition for models of true QBFs. By replacing Skolem with Herbrand sets and exchanging the roles of universal and existential variables, one obtains a dual version for counter-models of false QBFs.

Definition 2 (Disjoint Models). *Two Skolem sets F and F' of a true QBF are* disjoint *if there exists an existential variable y and $f_y \in F$, $f'_y \in F'$ such that f_y and f'_y are not logically equivalent.*

The notion of disjointness can also be stated in terms of the tree representation of Skolem/Herbrand sets: two Skolem/Herbrand sets are disjoint if and only if their tree representations do not share a common path, i.e., there is no full path from the root to a leaf that appears in both trees. We summarize some properties of disjoint models described in [15]. A dual version for disjoint counter-models for false QBFs can be obtained by replacing blocking Skolem sets by blocking Herbrand sets, which are added disjunctively to the formula.

Proposition 1. *Let $\Phi = \Pi.\phi$ be a true QBF. (1) If F is a Skolem set of Φ and F' is a Skolem set of $\Pi.(\phi \wedge \neg\phi_F)$, then F and F' are disjoint. (2) If $F_1, \ldots, F_m$ are pairwise disjoint Skolem sets for Φ and if $\Pi.(\phi \wedge \neg\phi_{F_1} \wedge \cdots \wedge \neg\phi_{F_m})$ is false, then m is the maximal number of pairwise disjoint Skolem sets for Φ.*

Proposition 1 suggests an iterative approach to enumerate all disjoint (counter-)models for a given QBF. Details of this algorithm are described in [15]. However, the somewhat restrictive notion of disjoint (counter-)models is not adequate to describe all interesting solutions of a QBF. Example 1 demonstrates that some solutions, such as the third strategy, cannot be discovered by enumerating only disjoint models.

5 Enumeration of All Solutions

When all disjoint models for a true QBF $\Phi = \Pi.\phi$ are computed, the corresponding Skolem functions cover all possibilities to turn the matrix ϕ into $\top$ for at least one assignment of the universal variables in Φ. However, assignments with additional possibilities might exist. For instance, in Example 1, the two disjoint models F_1 and F_2 cover all possible moves to counter a move of player ◯ on positions 1, 3, or 4. However, there still exists a third possibility when player ◯ puts their mark on position 2. To formally describe such partial solutions, we introduce the following notion.

Definition 3 (Restricted Models). *Let $\Phi = \Pi.\phi$ be a true QBF and let P be a propositional formula in the universal variables of Φ. If F is a Skolem set for $\Pi.(P \rightarrow \phi)$, then (F, P) is a* restricted Skolem set *(or* restricted model*) for Φ.*

In a restricted Skolem set (F, P) for Φ, F is a model for Φ only for those assignments of the universal variables for which the formula P evaluates to true. For assignments that do not satisfy P, the relation of F and Φ can be arbitrary. A restricted Skolem set is a classical Skolem set if and only if $P \equiv \top$. Restricted counter-models are defined dually.

Definition 4 (Restricted Counter-models). *Let $\Phi = \Pi.\phi$ be a false QBF and let P be a propositional formula in the existential variables of Φ. If H is a Herbrand set for $\Pi.(\phi \wedge P)$, then the pair (H, P) is a* restricted Herbrand set *(or* restricted counter-model*) for Φ.*

Example 2. Consider the QBF $\Phi = \forall x \exists y.(x \vee y)$. Then $(\{f_y = \neg x\}, \top)$ and $(\{f_y = \bot\}, x)$ are two restricted models for Φ, but $(\{f_y = \bot\}, \neg x)$ is not a restricted model for Φ.

Once an algorithm has computed all disjoint solutions for a QBF, there might still exist restricted (counter-)models. In this section, we discuss two methods to enumerate these remaining partial solutions. The algorithms presented in the following do not actually enumerate *all* possible solutions for a given QBF, but they compute a set of solutions that allows us to obtain all possible solutions by combining them. We call such a set a *basis*.

Definition 5 (Basis). *Let $\Phi = \Pi.\phi$ be a true QBF with universal variables X and existential variables Y. A set $\mathcal{F} = \{(F_1, P_1), \ldots, (F_m, P_m)\}$ of restricted Skolem sets for Φ is called a* basis *of Skolem sets for Φ if it satisfies the following condition: for any Skolem set F of Φ and any X-assignment σ, there exists exactly one partial Skolem set $(F', P') \in \mathcal{F}$ such that $P'(\sigma) \equiv \top$ and $f_y(\sigma) \equiv f'_y(\sigma)$ for all $y \in Y$, where $f_y \in F$ and $f'_y \in F'$.*

A basis of Herbrand sets for false QBFs is defined dually. The condition of being a basis of Skolem/Herbrand sets consists of two parts: For each Skolem set F and each X-assignment σ, there is *at least one* and *at most one* partial Skolem set mimicking the behavior of F under σ. The "at least one" condition guarantees that a basis can generate all possible Skolem sets for Φ, while the "at most one" condition ensures that there is no redundancy within a basis.

Example 3. Consider the QBF $\Phi = \forall x \exists y.(x \vee y)$. A basis of Skolem sets for Φ is given by $\{(\{f_y = \neg x\}, \top), (\{f_y = x\}, x)\}$. Another possible basis is $\{(\{f_y = \top\}, x), (\{f_y = \bot\}, x), (\{f_y = \top\}, \neg x)\}$.

Example 3 shows that a basis for a QBF is not unique and that different bases may contain different numbers of elements. The main reason for this variability is that any (nontrivial) basis element can be decomposed into mutually exclusive parts. Specifically, if (F, P) is an element of a basis $\mathcal{F}$ and P can be expressed as $P \equiv P_1 \vee P_2$, where P_1 and P_2 have no models in common, then (F, P) can be replaced by (F, P_1) and (F, P_2), increasing the number of elements in $\mathcal{F}$ by one.

Conversely, to reduce the number of elements in a basis, we can merge elements $(F_1, P_1), (F_2, P_2)$ into a single element $(F, P_1 \vee P_2)$, provided that P_1 and P_2 do not share a common model. The new Skolem functions in F are then given by $(P_1 \to f_{y,1}) \wedge (P_2 \to f_{y,2})$, where $f_{y,1} \in F_1$, $f_{y,2} \in F_2$. By repeatedly merging elements until no further reductions are possible, we can obtain a minimal basis. The minimality of a basis can, in fact, be characterized by a simple property of its restriction conditions P.

Definition 6 (Minimal Basis). *A basis $\mathcal{F}$ is* minimal *if there exists an assignment σ such that $P(\sigma) \equiv \top$ for all $(F, P) \in \mathcal{F}$.*

The first basis in Example 3 is minimal, while the second one is not. Minimal bases have a unique and minimal number of elements.

Proposition 2. *Let $\mathcal{F}$, $\mathcal{F}'$ be bases for a QBF Φ. If $\mathcal{F}$ is minimal, then $|\mathcal{F}| \leq |\mathcal{F}'|$. In particular, all minimal bases for Φ have the same number of elements.*

Proof. Let $\mathcal{F} = \{(F_1, P_1), \ldots, (F_m, P_m)\}$. Since $\mathcal{F}$ is minimal, there exists an assignment σ such that $P_i(\sigma) \equiv \top$ for all $i = 1, \ldots, m$. Since $\mathcal{F}$ is a basis, the evaluations of the F_i under σ are all pairwise different. For each of these m different evaluations, there has to exist a partial (counter-)model $(F'_i, P'_i) \in \mathcal{F}'$ with $P'_i(\sigma) \equiv \top$ and $f_y(\sigma) \equiv f'_y(\sigma)$ for all $y \in Y$, $f_y \in F_i$, $f'_y \in F'_i$. Since these elements must be pairwise distinct, it follows that $|\mathcal{F}'| \geq m$. The second claim follows from the first one.

For true QBFs of the form $\forall X \exists Y.\phi$ we can easily compute the total number of Skolem sets from a basis[1]. By replacing Skolem with Herbrand sets, one obtains an analogous result for false formulas with prefix $\exists X \forall Y \exists Z$.

Proposition 3. *Let $\Phi = \forall X \exists Y.\phi$ be a true QBF. Furthermore let $\mathcal{F} = \{(F_1, P_1), \ldots, (F_m, P_m)\}$ be a basis of Skolem sets for Φ. The total number of Skolem sets for Φ is given by*

$$\prod_{\substack{X\text{-assignment} \\ \sigma}} |\{i \in \{1, \ldots, m\} \mid P_i(\sigma) \equiv \top\}|.$$

[1] In fact, we could also consider true formulas with prefix $\forall X \exists Y \forall Z$, but with universal reduction [9] the rightmost universal quantifier can always be eliminated without changing the number of models.

Proof. Any Skolem set F for Φ is uniquely determined by the evaluations of all Skolem functions $f_y \in F$ under all X-assignments σ. Since these evaluations are independent for different assignments, it suffices to compute the number of possible evaluations for each σ and then multiply them to obtain the total number of Skolem sets.

For fixed σ, we claim that $|\{i \in \{1,\ldots,m\} \mid P_i(\sigma) \equiv \top\}|$ gives precisely the number of different possible evaluations of Skolem sets under σ. To see this, note that each partial Skolem set $(F_i, P_i) \in \mathcal{F}$ with $P_i(\sigma) \equiv \top$ gives exactly one possibility for the evaluations under σ, given by evaluating the partial Skolem functions in F_i under σ. Note that these evaluations all have to be different, because $\mathcal{F}$ is a basis. If there were different $(F_i, P_i), (F_j, P_j) \in \mathcal{F}$ with $P_i(\sigma) \equiv P_j(\sigma) \equiv \top$ and $f_{i,y}(\sigma) \equiv f_{j,y}(\sigma)$ for all $y \in Y$, then this would violate the "at most one"-part of being a basis. Thus, $|\{i \in \{1,\ldots,m\} \mid P_i(\sigma) \equiv \top\}|$ gives a lower bound for the number of different possible evaluations under σ. To see that it is also an upper bound, we have to show that the evaluation of every Skolem set F for Φ under σ can be obtained from one of the (F_i, P_i), but this follows directly from the "at least one"-part of the basis property.

Next, we present two methods for computing a basis of solutions for a given QBF. For a simpler presentation, we consider true QBFs with quantifier prefix $\forall X \exists Y$ and false QBFs with quantifier prefix $\exists X \forall Y \exists Z$.

5.1 SAT-Based Basis Computation

In [15], an approach was presented for computing the number of all models for a true QBF $\Phi = \forall X \exists Y.\phi$. We describe an adaptation of this method to compute a basis of Skolem sets for Φ. The total number of solutions follows from Proposition 3.

First, we compute all disjoint models for Φ using, for instance, [15, Alg. 1]. Then, in order to identify additional restricted models, we invoke a SAT solver on the propositional matrix ϕ. If there still exist restricted solutions, the solver returns assignments σ_X and σ_Y of the universal and existential variables, respectively, such that ϕ evaluates to true under the combination of σ_X and σ_Y. From these assignments, we can construct a partial Skolem set (F, P), where P a propositional formula with unique satisfying assignment σ_X, and $F = \{f_y = \sigma_Y(y) \mid y \in Y\}$. We then exclude this partial solution from the search space by conjunctively appending the formula $\neg(P \wedge \phi_F)$ to ϕ. This process is repeated iteratively until all satisfying assignments of the propositional matrix have been identified. This approach does not work for false formulas.

5.2 Refinement-Based Basis Computation

We now describe the refinement-based approach for computing a basis of solutions for arbitrary QBFs. The details of the method are described in Algorithm 1. For a simpler presentation, we state the algorithm for true QBFs of the form $\Phi = \forall X \exists Y.\phi$. The needed changes for false QBFs $\exists X \forall Y \exists Z.\phi$ are discussed later.

The core idea of the algorithm is to alternate between computing disjoint models and disjoint counter-models for iteratively refined formulas. For true formulas, we begin by computing all disjoint models and exclude them as blocking Skolem sets. The resulting formula becomes false, and we compute its disjoint counter-models. These counter-models indicate for which parts of the search space we have already identified all solutions. By excluding the counter-models as blocking Herbrand sets, we restrict our search to only those parts of the search space that still contain solutions. Thus, any newly identified models will be partial models of the original formula, restricted to the remaining parts of the search space. To keep track of these restrictions, we construct a formula P that encodes which parts of the search space are still under consideration. We then continue by computing disjoint models for the updated true formula, obtaining partial models that are restricted by P. This alternating process continues, refining the restrictions P of the constructed partial models in every iteration, until P becomes unsatisfiable, indicating that all partial solutions have been found.

Algorithm 1 works as follows. First, we initialize a propositional formula P. This formula represents those assignments of the universal variables X for which we have not yet identified all partial Skolem sets and is continuously updated as the algorithm progresses. Since initially all assignments lead to solutions, P is initialized as $\top$. Furthermore, the algorithm defines a propositional formula ψ, which is initially set to ϕ and will continuously be refined with found blocking Skolem and Herbrand sets. The stopping criterion of the algorithm is defined in line 2, where the loop terminates once all (partial) solutions for all assignments have been identified, which is the case if and only if P becomes unsatisfiable. Within the loop in line 3, first all disjoint solutions for the current QBF instance $\forall\exists.\psi$ are computed using a black-box algorithm computeDisjoint($\forall\exists.\psi$) (this could be, e.g., [15, Alg. 1]). By construction, this formula is always true and we obtain a set of all disjoint Skolem sets $\mathcal{F}'$ for $\forall\exists.\psi$. The formula ψ is constructed so that assignments that do not satisfy P are excluded from consideration. This allows us to ignore those parts of the search space for which all solutions have already been found. Therefore, the found Skolem sets for $\forall\exists.\psi$ are only partial models for $\forall\exists.\phi$, restricted to models of P. In line 4, we construct those restricted models and append them to the basis $\mathcal{F}$.

In line 5, the propositional formula ψ is enriched with blocking sets to exclude the disjoint solutions from $\mathcal{F}'$. Since $\forall\exists.\psi$ is now a false QBF, the next call to computeDisjoint($\forall\exists.\psi$) in line 6 yields a set $\mathcal{H}$ of all disjoint Herbrand sets for $\forall\exists.\psi$. As the universal variables appear in the first quantifier block, each of these Herbrand sets corresponds to precisely one assignment of the universal variables for which ψ evaluates to false. These assignments indicate new parts of the search space for which we have already identified all (partial) solutions for $\forall\exists.\phi$.

Thus, these assignments must be excluded in all subsequently identified partial solutions. To do this, the formula P is updated in line 7 by appending the negations of these assignments to P. Additionally, also the formula ψ is updated to exclude the exhausted parts of the search space by disjunctively appending

Algorithm 1: Refinement-Based Basis Computation – True Formula

Input: true QBF $\Phi = \Pi.\phi = \forall X \exists Y.\phi$
Output: A minimal basis $\mathcal{F}$ of Skolem sets for Φ

1: $\mathcal{F} \leftarrow \emptyset$, $P \leftarrow \top$, $\psi \leftarrow \phi$
2: **while** P is satisfiable **do**
3: $\quad \mathcal{F}' \leftarrow \text{computeDisjoint}(\Pi.\psi)$
4: $\quad \mathcal{F} \leftarrow \mathcal{F} \cup \{(F, P) \mid F \in \mathcal{F}'\}$
5: $\quad \psi \leftarrow \psi \wedge \bigwedge_{F \in \mathcal{F}'} \neg\phi_F$
6: $\quad \mathcal{H} \leftarrow \text{computeDisjoint}(\Pi.\psi)$
7: $\quad P \leftarrow P \wedge \bigwedge_{H \in \mathcal{H}} \neg\phi_H$
8: $\quad \psi \leftarrow \psi \vee \bigvee_{H \in \mathcal{H}} \phi_H$
9: **end while**
10: **return** $\mathcal{F}$

blocking Herbrand sets from $\mathcal{H}$ in line 8. This turns the QBF $\Pi.\psi$ back into a true formula and the next iteration begins.

Notably, the algorithm remains largely the same for false input formulas. The only differences occurs in line 5 and line 8, which must be replaced for false formulas by the following modified snippet.

5: $\psi \leftarrow \psi \vee \bigvee_{F \in \mathcal{F}'} \phi_F$
8: $\psi \leftarrow \psi \wedge \bigwedge_{H \in \mathcal{H}} \neg\phi_H$

The set $\mathcal{F}'$ now contains disjoint Herbrand sets and $\mathcal{H}$ disjoint Skolem sets. Consequently, we have adapted how blocking sets are appended to the formula ψ accordingly. The remainder of the algorithm remains unchanged.

We note that Algorithm 1 can be interrupted at any time to return all solutions that have been found until that point.

Theorem 1. *Algorithm 1 terminates and is correct.*

Proof. For termination, note that, because the universal variables appear in the first quantifier block, each Herbrand set $H \in \mathcal{H}$ computed in line 6 corresponds to precisely one assignment of the universal variables. Thus, by appending the negation of this Herbrand set $\neg\phi_H$ to P in line 7, this assignment gets excluded from the models of P. Since the Herbrand sets are also excluded from ψ in line 8, they cannot reoccur in subsequent iterations. Thus, in each iteration, we exclude at least one new assignment from the models of P. Consequently, after at most $2^{|X|}$ iterations P will become unsatisfiable and the algorithm will terminate.

For correctness, we show that the returned set $\mathcal{F}$ satisfies the property of Definition 5. The minimality of $\mathcal{F}$ follows directly from the construction. We split this proof into two parts: existence of a partial Skolem set with the desired property and uniqueness of it.

For existence, let F be an arbitrary Skolem set for Φ and let σ be an arbitrary X-assignment. At some point in the algorithm, a Herbrand set corresponding to σ is computed. Say this happens in the k-th iteration. Let $\mathcal{F}_k$ be the value of $\mathcal{F}$ at the beginning of the k-th iteration and let $\mathcal{F}'_k$ be the set of disjoint Skolem

sets computed during the k-th iteration. If there exists a partial model in $\mathcal{F}_k$ satisfying the condition of Definition 5 for F and σ, we are done. So assume that this is not the case. We will show that such a partial model is then constructed from $\mathcal{F}'_k$ during the k-th iteration.

Any partial model (F', P) constructed in line 4 during the k-th iteration satisfies $P(\sigma) \equiv \top$, because, by assumption, the Herbrand set that excludes σ is only computed and appended to P afterwards. Moreover, this also implies that all remaining partial solutions for the assignment σ are contained in $\mathcal{F}'_k$. This means that the X-assignment σ can be extended to a full assignment τ on $X \cup Y$ such that $\psi(\tau) \equiv \top$ (there had been partial solutions before) and $(\psi \wedge \bigwedge_{F' \in \mathcal{F}'_k} \neg\phi_{F'})(\tau) \equiv \bot$ (there are no more partial solutions left). In particular, because F does not appear in $\mathcal{F}_k$ by assumption, we can choose τ so that $\tau(y) \equiv f_y(\sigma)$ for all $f_y \in F$. But this means that there exists $F' \in \mathcal{F}'_k$ with $f'_y \in F'$ such that $\phi_{F'}(\tau) = (\bigwedge_{y \in Y} y \leftrightarrow f'_y)(\tau) \equiv \top$, showing that $f_y(\sigma) \equiv \tau(y) \equiv f'_y(\tau) \equiv f'_y(\sigma)$ for all $y \in Y$ as required in Definition 5.

For uniqueness, let σ be an arbitrary X-assignment. We show that once a partial Skolem set (F, P) has been computed and F has been excluded from ψ in line 5, no subsequent partial model (F', P') will be computed such that $P'(\sigma) \equiv \top$ and $f_y(\sigma) \equiv f'_y(\sigma)$ for all $y \in Y$, $f_y \in F$, $f'_y \in F'$. This implies the uniqueness condition in Definition 5.

Let σ_F denote the extension of σ to the existential variables, defined by $\sigma_F(y) \equiv f_y(\sigma)$ for all $y \in Y$. By definition, we have $\phi_F(\sigma_F) \equiv \top$. Thus, once $\neg\phi_F$ is appended to ψ, σ_F no longer satisfies ψ. Disjunctively appending blocking Herbrand sets H to ψ in line 8 does not affect this as long as $\phi_H(\sigma) \equiv \bot$. Hence, another Skolem set F' with $f_y(\sigma) \equiv f'_y(\sigma)$ for all $y \in Y$ can only appear after a blocking Herbrand set with $\phi_H(\sigma) \equiv \top$ has been appended to ψ. However, in this case, P is also updated to exclude σ as a model, ensuring that from that point onward $P(\sigma) \equiv \bot$. Thus, we never compute another partial Skolem set (F', P') that satisfies both $P'(\sigma) \equiv \top$ and $f_y(\sigma) \equiv f'_y(\sigma)$ for all $y \in Y$.

Example 4. For $\Phi = \Pi.\phi = \forall x, y \exists a, b.(y \vee \neg a) \wedge (x \vee \neg y \vee \neg b) \wedge (\neg x \vee \neg y \vee \neg a \vee \neg b)$ the full assignment tree of Φ is depicted in Figure 2a. We use Algorithm 1 to compute a basis of Skolem sets for Φ. To this end, we initialize $\mathcal{F}$ to the empty set, P to $\top$, and ψ to ϕ. Then we compute all disjoint models for Φ.

Disjoint models: Disjoint models for $\Pi.\psi$ are $F_1 = \{f_a(x, y) = \bot, f_b(x, y) = \bot\}$ and $F_2 = \{f_a(x, y) = y, f_b(x, y) = \neg y\}$. Since P is $\top$, we update $\mathcal{F}$ to $\mathcal{F} = \{(F_1, \top), (F_2, \top)\}$. Moreover, we exclude F_1 and F_2 from the search space by appending them as blocking Skolem sets. This updates ψ to $\psi' = \phi \wedge \neg\phi_{F_1} \wedge \neg\phi_{F_2}$. We have now identified all disjoint models for Φ. The resulting formula is now false, so we proceed by computing counter-models for $\Pi.\psi'$. The full assignment tree of the false QBF $\Pi.\psi'$ is depicted in Figure 2b.

Disjoint counter-models: The disjoint counter-models for $\Pi.\psi'$ are $H_1 = \{h_x = \bot, h_y = \bot\}$, $H_2 = \{h_x = \bot, h_y = \top\}$, and $H_3 = \{h_x = \top, h_y = \bot\}$. Hence, there are no more partial solutions for both assignments that map x to $\bot$ as well as for the assignment $\sigma_3 = \{x \mapsto \top, y \mapsto \bot\}$. We update P to $P = \neg\phi_{H_1} \wedge \neg\phi_{H_2} \wedge \neg\phi_{H_3} \equiv x \wedge y$, in order to exclude these assignments from

all future partial models. Moreover, also ψ' is updated to

$$\psi'' = \psi' \vee \phi_{H_1} \vee \phi_{H_2} \vee \phi_{H_3} = (\phi \wedge \neg\phi_{F_1} \wedge \neg\phi_{F_2}) \vee \phi_{H_1} \vee \phi_{H_2} \vee \phi_{H_3}.$$

By adding the counter-models H_1, H_2 and H_3 as blocking Herbrand sets, any extension of the assignments encoded in H_1, H_2, and H_3 to the existential variables a and b leads to a satisfying assignment for ψ'' (see Figure 2c). As $\Pi.\psi''$ is a true QBF, we continue by computing models. All further models for $\Pi.\psi''$ are only partial models for Φ, restricted to models of $P = x \wedge y$.

Disjoint models: The formula $\Pi.\psi''$ only has one model, namely $F_3 = \{f_a(x, y) = \bot, f_b(x, y) = \top\}$. Note that our construction only ensures that F_3 renders the original formula ϕ true for the assignment $\{x \mapsto \top, y \mapsto \top\}$. For all other assignments, we cannot make any guarantees. For instance, under the assignment $\{x \mapsto \bot, y \mapsto \top\}$ the Skolem functions in F_3 let ϕ evaluate to $\bot$. We update $\mathcal{F}$ to $\mathcal{F} = \{(F_1, \top), (F_2, \top), (F_3, x \wedge y)\}$ and append F_3 as a blocking Skolem set to ψ''. This yields $\psi''' = \psi'' \wedge \neg\phi_{F_3}$. As $\Pi.\psi'''$ is now a false formula again, we compute its counter-models. The full assignment tree of $\Pi.\psi'''$ is depicted in Figure 2d.

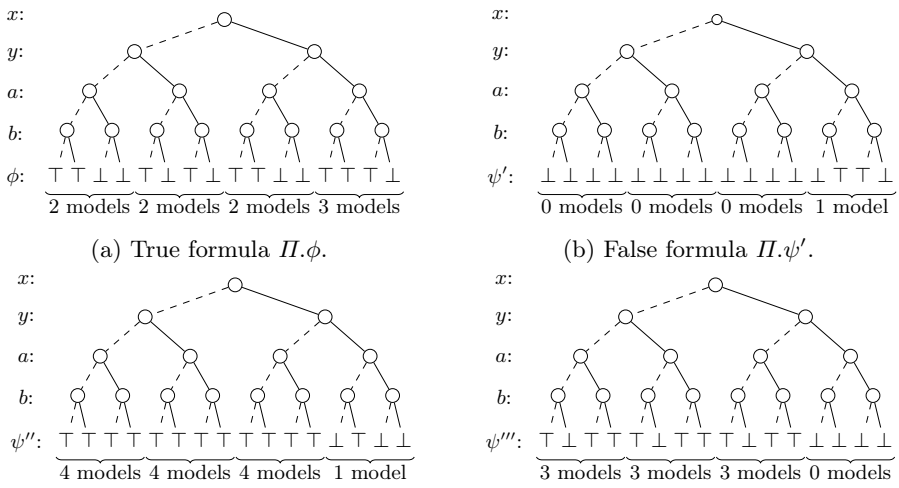
Disjoint counter-models: The formula $\Pi.\psi'''$ only has one counter-model, given by $H_4 = \{h_x = \top, h_y = \top\}$. Hence, we have found all partial solutions for the last open assignment that maps both x and y to $\top$. We update P accordingly to $P = P \wedge \neg\phi_{H_4} \equiv \bot$. For completeness, we update ψ''' by appending H_4 as a blocking Herbrand set. Since P is now unsatisfiable, we can stop our computation and know that a minimal basis of Skolem sets is given by $\mathcal{F} = \{(F_1, \top), (F_2, \top), (F_3, x \wedge y)\}$. Using Proposition 3, we can also deduce that the total number of Skolem sets for Φ is $2 \cdot 2 \cdot 2 \cdot 3 = 24$.

6 Evaluation

We implemented the presented approach for solution enumeration in C++, using DepQBF 6.03 [12] as backend solver. In our tool QEnum[2], the resolution proofs for true and false QBFs produced by this solver are processed by the QBF certification framework QRPCert [14], extracting the Skolem and Herbrand functions in the AIGER format.[3] The Skolem (Herbrand) functions are appended conjunctively (disjunctively) using the incremental interface of DepQBF. For true QBFs, the propositional model-counter Ganak [17] is called, while OuterCount [19] is called for false QBFs. We note that our tool is the first to enumerate and count all solutions of false QBFs. Thus, for this use case, no comparison to other tools is possible. All experiments were performed on dual-socket AMD EPYC 7313 16 cores @ 3.7 GHz machines running Ubuntu 24.04 with a 32 GB memory limit.

Case Study 1: Tic-Tac-Toe Encoding In our first case study, we consider smaller encodings of the game Tic-Tac-Toe with boards of size 4×4 and 5×5, similar to the example discussed in Section 3. We evaluated how many solutions we

[2] https://github.com/PlankAndreas/QEnum
[3] http://fmv.jku.at/aiger/



(a) True formula $\Pi.\phi$. (b) False formula $\Pi.\psi'$.

(c) True formula $\Pi.\psi''$. (d) False formula $\Pi.\psi'''$.

Fig. 2: Assignment tree of the formula from Example 4 as Algorithm 1 progresses. Dashed edges indicate assignments to $\bot$ and solid edges to $\top$.

Table 1: Experiments with Tic-Tac-Toe encoding.

Tic-Tac-Toe		formula structure		#solutions		runtime [s]	
size	open positions	#vars	#clauses	#disjoint	#total	disjoint	total
4x4	3	5	31	1	4	0.23	0.42
	4	6	261	2	24	0.35	1.14
	5	8	3131	1	64	0.22	0.81
	6	9	46 663	1	64	0.44	1.17
5x5	7	10	823 551	1	64	56.35	63.72
	8	11	16 777 225	1	64	28 828.64	29 301.60

could enumerate within a time limit of 12 h and a memory limit of 32 GB. This set is of interest because the enumerated solutions can be manually checked and evaluated. The results of these evaluations are presented in Table 1. All instances are true. Notably, we observe a rise in runtime for the formula with seven open positions, driven by a significant increase in the number of clauses. Interestingly, while computing all models requires some effort, the computation time is still significantly lower than that needed to compute the disjoint models. Although the runtime increases for the final formulas, the computed models remain small.

Case Study 2: Solution Enumeration of Benchmarks from QBF Evaluation In the second case study, we evaluate how many solutions can be explicitly enumerated for benchmarks used in the 2QBF and PCNF tracks of QBFEval 2022 within a time frame of 300 s consisting of 950 formulas. This is possible, because our approach is based on an anytime algorithm, which returns a subset of the

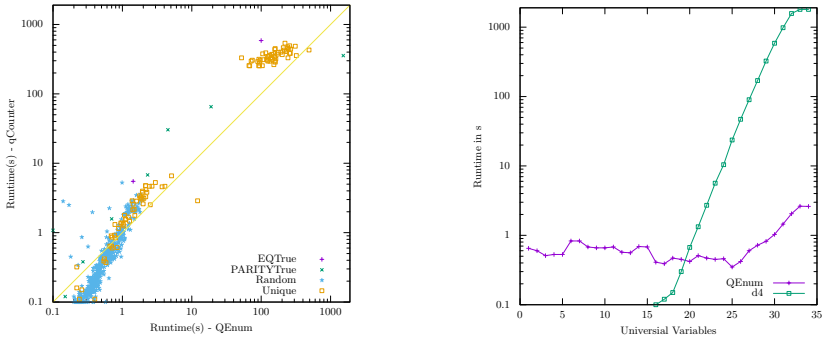


Fig. 3: QEnum vs qCounter (left) and d4 (right) for full model counting.

solutions if it is interrupted before termination. In total, we could find 292 solutions. One solution was found for 158 formulas. For 19 formulas between two and ten solutions were returned. Finally, more than ten solutions were found for ten formulas (with 42 being the highest model count that was computed within less than one minute). Note that these formulas are very hard and need to be simplified by preprocessors [8] which modify the solution space.

Case Study 3: Solution Counting We compare the model counting capabilities of our tool with exact solution counters. We focus on crafted benchmarks, because the formulas considered in the second case study are beyond the scope of modern solution counters. Further, the exact and complete counters d4 [11] and qCounter [15] only work for true formulas. First, we compare the runtime of the SAT-based solution counter qCounter with QEnum. We consider crafted benchmark families as in [7], unique-SAT encodings, as well as randomly generated formulas for which the full model count is known by construction. These formulas were already considered in [11, 15] for the evaluation of solution counters. The results are shown in the left part of Figure 3. QEnum outperforms the comparable tool qCounter for the harder formulas. For both tools, the crafted formulas are hard by construction (as they rely on the Q-resolution based solver DepQBF). For the recursive model counter d4 all those formulas are easy. However, there are also cases where QEnum outperforms d4 as shown on the right of Figure 3 for formulas $\Phi_i = \forall X \exists Y.\phi$ such that $|X| = i$, $|Y| = 35 - i$, and ϕ is a simple CNF formula over these variables. Obviously, the recursive search of d4 does not involve any advanced QBF reasoning techniques.

Conclusion The case studies illustrate the practical feasibility of our enumeration-based approach and show that it is comparable to exact state-of-the-art model counters. In the future, we plan to extend our approach to QBFs with arbitrary quantifier structure, e.g., by combining the recursive search of d4 with the refinement-based approach of QEnum. We also plan to investigate the potential of preprocessing, which requires developing solution reconstruction techniques.

Acknowledgements We thank the anonymous reviewers for their feedback. This research was funded by the Austrian Science Fund (FWF) [10.55776/COE12]. C.H. was supported by the LIT AI Lab funded by the state of Upper Austria.

Bibliography

[1] Baluta, T., Shen, S., Shinde, S., Meel, K.S., Saxena, P.: Quantitative verification of neural networks and its security applications. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. pp. 1249–1264. ACM, New York, NY, USA (2019)

[2] Becker, B., Ehlers, R., Lewis, M., Marin, P.: ALLQBF solving by computational learning. In: Automated Technology for Verification and Analysis. pp. 370–384. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)

[3] Beyersdorff, O., Mikolás, J., Lonsing, F., Seidl, M.: Quantified Boolean Formulas. In: Handbook of Satisfiability, vol. 336, pp. 1177–1221. IOS Press, Amsterdam, Netherlands (2021)

[4] Biondi, F., Enescu, M.A., Heuser, A., Legay, A., Meel, K.S., Quilbeuf, J.: Scalable approximation of quantitative information flow in programs. In: Proc. of Int. Conf. on Verification, Model Checking, and Abstract Interpretation. LNCS, vol. 10747, pp. 71–93. Springer, Cham (2018)

[5] Chakraborty, S., Meel, K.S., Vardi, M.Y.: Algorithmic improvements in approximate counting for probabilistic inference: From linear to logarithmic SAT calls. In: Proc. of Int. Joint Conf. on Artificial Intelligence. pp. 3569–3576. IJCAI/AAAI Press, USA (2016)

[6] Gomes, C.P., Sabharwal, A., Selman, B.: Model counting. In: Handbook of Satisfiability, pp. 993–1014. IOS Press, Amsterdam, Netherlands (2021)

[7] Heisinger, S., Seidl, M.: True crafted formula families for benchmarking quantified satisfiability solvers. In: Intelligent Computer Mathematics. pp. 291–296. Springer Nature Switzerland, Cham (2023)

[8] Heule, M., Järvisalo, M., Lonsing, F., Seidl, M., Biere, A.: Clause elimination for SAT and QSAT. Journal of Artificial Intelligence Research **53**, 127–168 (06 2015)

[9] Kleine Büning, H., Lettmann, T.: Propositional logic: deduction and algorithms, Cambridge Tracts in Theoretical Computer Science, vol. 48. Cambridge University Press, USA (1999)

[10] Ladner, R.E.: Polynomial space counting problems. SIAM J. Comput. **18**(6), 1087–1097 (1989)

[11] Lagniez, J.M., Capelli, F., Plank, A., Seidl, M.: A top-down tree model counter for quantified boolean formulas. In: Proc. of the 33rd. Int. Conf. on International Joint Conference on Artificial Intelligence IJCAI (2024)

[12] Lonsing, F., Egly, U.: DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In: Proc. of the 26th Conf. on Automated Deduction. LNCS, vol. 10395, pp. 371–384. Springer, Cham (2017)

[13] Narodytska, N., Shrotri, A.A., Meel, K.S., Ignatiev, A., Marques-Silva, J.: Assessing heuristic machine learning explanations with model counting. In: Proc. of the Int. Conf. on Theory and Applications of Satisfiability Testing. LNCS, vol. 11628, pp. 267–278. Springer, Cham (2019)

[14] Niemetz, A., Preiner, M., Seidl, M., Biere, A.: Resolution-based certificate extraction for QBF - (tool presentation). In: Proc. of the 15th Int. Conf. on Theory and Applications of Satisfiability Testing. LNCS, vol. 7317, pp. 430–435. Springer, Berlin, Heidelberg (2012)
[15] Plank, A., Möhle, S., Seidl, M.: Counting QBF solutions at level two. Constraints **29**(1), 22–39 (2024)
[16] Sang, T., Beame, P., Kautz, H.A.: Performing Bayesian inference by weighted model counting. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence. pp. 475–482. AAAI Press / The MIT Press, Dagstuhl, Germany (2005)
[17] Sharma, S., Roy, S., Soos, M., Meel, K.S.: Ganak: A scalable probabilistic exact model counter. In: Proc.of Int. Joint Conf. on Artificial Intelligence. pp. 1169–1176. Int. Joint Conf. on Artificial Intelligence Organization, Macao, China (2019)
[18] Shaw, A., Juba, B., Meel, K.S.: An approximate skolem function counter. In: Conf. on Artificial Intelligence, AAAI. pp. 8108–8116. AAAI Press, Vancouver, Canada (2024)
[19] Shukla, A., Biere, A., Pulina, L., Seidl, M.: A survey on applications of quantified Boolean formulas. In: Proc. of the Int. Conf. on Tools with Artificial Intelligence. pp. 78–84. IEEE, USA (2019)
[20] Zhou, Z., Qian, Z., Reiter, M.K., Zhang, Y.: Static evaluation of noninterference using approximate model counting. In: Proc. of IEEE Symposium on Security and Privacy. pp. 514–528. IEEE Computer Society, San Francisco, CA, USA (2018)