A Unified Framework for Certificate and Compilation for QBF

Igor Stéphan, Benoit da Mota

A unified framework for Certificate and Compilation for QBF

Igor Stéphan and Benoit Da Mota

LERIA University of Angers, France
email: {damota, stephan}@info.univ-angers.fr

Abstract. We propose in this article a unified framework for certificate and compilation for QBF. We provide a search-based algorithm to compute a certificate for the validity of a QBF and a search-based algorithm to compile a valid QBF in our unified framework.

1 Introduction

The quantified Boolean formula (QBF) validity problem is a generalisation of the Boolean formulae satisfiability problem. While the complexity of Boolean satisfiability problem is NP-complete, it is PSPACE-complete for the QBF validity problem. This is the price for a more concise representation of many classes of formulae. Many important problems in several research fields have polynomial-time translations to the QBF validity problem : AI planning [24, 1], Bounded Model Construction [1], Formal Verification (see [5] for a survey).

Most of the recent decision procedures for QBF validity [11, 28, 19, 18] are extensions of the search-based Davis-Putnam procedure [15] for Boolean satisfiability. A search-based procedure for QBF chooses one Boolean variable, tries to solve two simpler subproblems and combines the results according to the semantics of the quantifier associated to the variable. Some other decision procedures are based on resolution principle (as Q-resolution [9] which extends the resolution principle for Boolean formulae to QBF or Quantor [6] which combines efficiently Q-resolution and expansion), quantifier-elimination algorithms [23, 22], or skolemization and SAT solvers [2].

Every finite two-player game can be modeled in QBF [17]. In this kind of applications, a decision procedure (the formula is valid or not) is not sufficient since a solution is needed. A solution of a QBF (a QBF model) is a set of Boolean functions [10]. One possibility to represent them is to build a tree-shape representation (called policy [13] or strategy [7]) but it is exponential in worst case and unfortunately also in usual ones. With a search-based procedure, it is very easy to build a solution of a QBF from the solutions of its two simpler subproblems [13, 7].

When a QBF solver returns valid or non-valid, there is no way to check if the answer is correct while in propositional logic the associated result to the decision (a model) is easy to check. A certificate for a valid QBF is any piece of information that provides self-supporting evidence of validity for that QBF [5].

A **sat**-certificate [4, 3] is a representation of a sequence of Boolean functions for a QBF that certifies its validity. This approach seems to us promising since the generated certificate is not linked to the representation of the input QBF but only to its semantics. The computation of a **sat**-certificate is described in [2] in the framework of sKizzo as a reconstruction from a trace. To the best of our knowledge, there is no result of how to build a **sat**-certificate of a QBF from the **sat**-certificates of its two simpler subproblems. It is an important issue since most of the QBF solvers are search-based procedures. Our first contribution is double (Section 4): we define an operator for **sat**-certificates in order to be able to build a **sat**-certificate of a QBF from the **sat**-certificates of its two simpler subproblems and we describe an algorithm which extends any search-based algorithm to build a **sat**-certificate for a valid QBF during the decision process and not a posteriori from a trace.

In general, a knowledge base is compiled off-line into a target language which is then used on-line to answer some queries. In QBF case, seen as a two-player game, one of the most useful query for the existential player is : what should I play to still be sure to win? Our second contribution is a unified framework for certificate and compilation of QBF (Section 3): the literal base representation which is an extension of the **sat**-certificate representation. In order to extend any search-based procedure to a QBF compiler, an important issue is how to compute the compilation of a QBF from the result of the compilation of its two simpler subproblems. Our third contribution is also double (Section 5): we define an operator for literal bases in order to be able to compute the compilation of a QBF from the literal bases of its two simpler subproblems and we describe an algorithm which extends any search-based algorithm to compile a valid QBF.

Finally we discusse related work (Section 6) and we draw a conclusion (Section 7).

2 Preliminaries

2.1 Propositional logic.

The set of propositional variables is denoted by $\mathcal{V}$. Symbols $\bot$ and $\top$ are the propositional constants ($\overline{\top} = \bot$ and $\overline{\bot} = \top$). Symbol $\wedge$ stands for conjunction, $\vee$ for disjunction, $\neg$ for negation, $\rightarrow$ for implication and $\leftrightarrow$ for bi-implication. A literal is a propositional variable or the negation of a propositional variable. A formula is in conjunctive normal form (CNF) if it is a conjunction of disjunctions of literals. Definitions of the language of propositional formula **PROP** and semantics of all the Boolean symbols are defined in standard way. A substitution is a function from propositional variables to **PROP**. This definition is extended as usual to a function from **PROP** to **PROP**: $[x \leftarrow F](G)$ is the formula obtained from G by replacing occurrences of the propositional variable x by the formula F. This definition is also extended as usual for the substitution of a formula by another formula. An interpretation v is a function from $\mathcal{V}$ to {**true**, **false**} ; the extension to **PROP** is denoted v^*. To an interpretation v is associated a set in the standard way and a substitution $\boldsymbol{v}$ as follows : if $v(x) =$ **true** then

$[x \leftarrow \top]$ is in $\boldsymbol{v}$, if $v(x) = \mathbf{false}$ then $[x \leftarrow \bot]$ is in $\boldsymbol{v}$. Propositional satisfaction is denoted $\models$ ($v \models F$ means $v^*(F) = \mathbf{true}$, the propositional formula F is satisfied by v and v is a model of F). Logical equivalence is denoted $\equiv$. To a Boolean function f (i.e. a function from $\{\mathbf{true}, \mathbf{false}\}^n$ to $\{\mathbf{true}, \mathbf{false}\}$) is associated a propositional formula ψ_f on the variables $\{x_1, \ldots, x_n\}$ such that $v^*(\psi_f) = \mathbf{true}$ if and only if $f(v(x_1), \ldots, v(x_n)) = \mathbf{true}$ for any interpretation v.

2.2 Quantified Boolean Formulae.

The symbol $\exists$ stands for the existential quantifier and $\forall$ stands for the universal quantifier(q stands for any quantifier). A binder Q is a string $q_1 x_1 \ldots q_n x_n$ with $x_1, \ldots, x_n$ distinct propositional variables and $q_1 \ldots q_n$ quantifiers. It is assumed that distinct quantifiers bind occurrences of distinct propositional variables. The empty string is denoted by ϵ. A (prenex) quantified Boolean formula (QBF) is constituted of a binder and a propositional formula called the matrix (only closed formulae are considered: each variable in the matrix is also in the binder). A QBF is in conjunctive normal form if its matrix is in conjunctive normal form. The semantics of QBF is defined as follows: for every propositional variable y and every QBF QM

$$\exists y QM = (Q[y \leftarrow \top](M) \vee Q[y \leftarrow \bot](M))$$

and

$$\forall y QM = (Q[y \leftarrow \top](M) \wedge Q[y \leftarrow \bot](M)).$$

A QBF F is valid if $F \equiv \top$. If y is an existentially quantified variable preceded by the universally quantified variables $x_1, \ldots, x_n$ we denote $\hat{y}_{x_1 \ldots x_n}$ its Skolem function from $\{\mathbf{true}, \mathbf{false}\}^n$ to $\{\mathbf{true}, \mathbf{false}\}$. A model for a valid QBF QM is a sequence $\psi_{\hat{y}_1}; \ldots; \psi_{\hat{y}_p}$ such that $[y_1 \leftarrow \psi_{\hat{y}_1}] \ldots [y_p \leftarrow \psi_{\hat{y}_p}](M)$ is a tautology [10] ($y_1, \ldots, y_p$ the existentially quantified variables of QM). For example, the QBF $\exists a \exists b \forall c((a \vee b) \leftrightarrow c)$ (since $\exists a \exists b \forall c((a \vee b) \leftrightarrow c) \equiv \exists a \exists b(((a \vee b) \leftrightarrow \top) \wedge ((a \vee b) \leftrightarrow \bot)) \equiv \bot$) but the QBF $\forall c \exists a \exists b((a \vee b) \leftrightarrow c)$ is valid and one of its possible model is $\psi_{\hat{a}}; \psi_{\hat{b}}$ with $\psi_{\hat{a}} = c$ and $\psi_{\hat{b}} = \bot$ (since $[a \leftarrow \psi_{\hat{a}}][b \leftarrow \psi_{\hat{b}}]((b \vee a) \leftrightarrow c) = ((\bot \vee c) \leftrightarrow c)$ is a tautology). A (Boolean) model of an unquantified Boolean formula corresponds exactly to a (QBF) model of its existential closure. A QBF is valid if and only if there exists a model. An interpretation v is said to be in a model $\psi_{\hat{y_1}}; \ldots; \psi_{\hat{y_p}}$ if for every i, $1 \leq i \leq n$, $v^*(\psi_{\hat{y_i}}) = v(y_i)$; for example the interpretation $v = \{c, a, \neg b\}$ is in the above model since $v^*(\psi_{\hat{a}}) = \mathbf{true} = v(a)$ and $v^*(\psi_{\hat{b}}) = \mathbf{false} = v(b)$ but $v' = \{\neg c, a, \neg b\}$ is not in it since $v'^*(\psi_{\hat{a}}) = \mathbf{false} \neq v'(a)$. Model-equivalence for QBF is defined in [25] as follows : Two QBF F and F' are model-equivalent (denoted $F \cong F'$) if every model of F is a model of F' and conversely; this equivalence is about preservation of models and not only preservation of validity; for example, $\forall c \exists a \exists b((a \vee b) \leftrightarrow c) \equiv \top$ but $\forall c \exists a \exists b((a \vee b) \leftrightarrow c) \not\cong \top$.

2.3 sat-certificate.

A **sat**-certificate [3] for a QBF F, with $y_1, \ldots, y_p$ its existentially quantified variables, is a sequence of pairs of formulae $(\phi_1, \nu_1); \ldots; (\phi_p, \nu_p)$, ϕ_i and ν_i de-

fined over the universally quantified variables of F preceding the variable y_i, $1 \leq i \leq p$. It is defined in [3] only for CNF QBF with sequences of pairs of binary decision diagrams (BDD) [8]. A **sat**-certificate is consistent if for every i, $1 \leq i \leq p$, $(\phi_i \wedge \nu_i) \equiv \bot$. If a **sat**-certificate is consistent then $\phi_1; \ldots; \phi_p$ and $\neg\nu_1; \ldots; \neg\nu_p$ are two models of the certified QBF (not necessarily the same). To certify the validity of a CNF QBF QM with a **sat**-certificate $(\phi_1, \nu_1); \ldots; (\phi_p, \nu_p)$ we have to check if $[\neg x_1 \leftarrow \nu_1][x_1 \leftarrow \phi_1] \ldots [\neg x_p \leftarrow \nu_p][x_p \leftarrow \phi_p](M)$ is a tautology. If the verification fails either the QBF is non valid or the **sat**-certificate is not correct; conversely, if the verification succeeds then the QBF is valid and the **sat**-certificate encodes a model [3]. For example, from [3], the sequence of pairs of formulae $(\phi_c, \nu_c); (\phi_e, \nu_e); (\phi_f, \nu_f)$ with $\phi_c = \neg a$, $\nu_c = a$, $\phi_e = (a \wedge b \wedge d) \vee (\neg a \wedge \neg d)$, $\nu_e = (\neg a \wedge d) \vee (a \wedge \neg d)$, $\phi_f = (a \wedge b \wedge \neg d) \vee (\neg a \wedge b \wedge d)$ and $\nu_f = (\neg a \wedge \neg d)$ is a **sat**-certificate for the CNF QBF

$$\xi = \forall a \forall b \exists c \forall d \exists e \exists f \mu \tag{1}$$

with

$$\begin{aligned}\mu = [&(\neg b \vee e \vee f) \wedge (a \vee c \vee f) \wedge (a \vee d \vee e) \wedge (\neg a \vee \neg b \vee \neg d \vee e) \wedge (\neg a \vee b \vee \neg c) \\ &\wedge (\neg a \vee \neg c \vee \neg f) \wedge (a \vee \neg d \vee \neg e) \wedge (\neg a \vee d \vee \neg e) \wedge (a \vee \neg e \vee \neg f)].\end{aligned}$$

This **sat**-certificate certifies the validity of this CNF QBF since $[\neg c \leftarrow \nu_c][c \leftarrow \phi_c][\neg e \leftarrow \nu_e]\ [e \leftarrow \phi_e][\neg f \leftarrow \nu_f][f \leftarrow \phi_f](M)$ is a tautology. One can remark that this **sat**-certificate is consistent and that

$$\phi_c; \phi_e; \phi_f = \neg a; (a \wedge b \wedge d) \vee (\neg a \wedge \neg d); (a \wedge b \wedge \neg d) \vee (\neg a \wedge b \wedge d) \tag{2}$$

and $\neg\nu_c; \neg\nu_e; \neg\nu_f = \neg a; \neg(\neg a \wedge d) \vee (a \wedge \neg d); \neg(\neg a \wedge \neg d)$ are a two different models for the QBF ξ.

3 Literal Base

In this section we present formally our proposal for a unified framework for certificate and compilation for QBF: the literal base representation. This representation extends the **sat**-certificate representation of [3].

Definition 1 (Literal base). *A* literal base *is a pair (Q, G) constituted*

- *either of $Q = \epsilon$ and $G = \top$ or $G = \bot$;*
- *either of a binder $Q = q_1 x_1 \ldots q_n x_n$, $n > 0$, and a sequence of pairs of formulae $G = (P_1, N_1); \ldots; (P_n, N_n)$ such that the formulae P_k and N_k, called* guards*, are only built on the variables $\{x_1, \ldots, x_{k-1}\}$ (or $\top$ or $\bot$ when $k = 1$).*

We denote $\mathcal{B}_Q$ the set of the literal bases for a binder Q, $LB = \bigcup_Q \mathcal{B}_Q$ the literal base language and define the function grds such that $grds((Q, G)) = G$.

A literal base is an explicit representation in the order of the binder of the dependencies that have to exist between an existentially quantified variable and the variables preceding it.

By the latter definition:

- if $Q = \epsilon$ then $\mathcal{B}_\epsilon = \{(\epsilon, \top), (\epsilon, \bot)\}$;
- if $Q = qx$ then $\mathcal{B}_{qx} = \{(qx, (\top, \top)), (qx, (\top, \bot)), \ \ (qx, (\bot, \top)), (qx, (\bot, \bot))\}$

If n is the number of variables of a binder Q then the size of $\mathcal{B}_Q$ is $\underbrace{2^{2^{\cdots^2}}}_{n+1}$.

We interpret a literal base language as a representation for a subset of the QBF language.

Definition 2 (Interpretation of a literal base). *The* interpretation function *is a function from LB to QBF denoted* $(.)^*$ *and is defined as follows :*

- *if* $lb = (\epsilon, G)$ *then* $lb^* = G$ *;*
- *if* $lb = (q_1 x_1 \ldots q_n x_n, (P_1, N_1); \ldots; (P_n, N_n)), n > 0$*, then*

$$lb^* = q_1 x_1 \ldots q_n x_n \bigwedge_{k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k))$$

If X is a subset of BL then X^* denotes $\{lb^* | lb \in X\}$. From here *non_valid* denotes a literal base whose interpretation is non valid and we extend the latter definition by $non\_valid^* = \bot$.

Clearly enough from Defintion 2, if a literal base $(Q, (P_1, N_1); \ldots; (P_n, N_n))$ is such that its interpretation is valid then necessarily for every universally quantified variable x_i, P_i and N_i can be replaced in the literal base by $\top$. The following literal base

$$\beta = (\forall a \forall b \exists c \forall d \exists e \exists f, (\top, \top); (\top, \top); (P_c, N_c); (\top, \top); (P_e, N_e); (P_f, N_f)) \quad (3)$$

with $P_c = \neg a$, $N_c = a$, $P_e = (\neg d \wedge c \wedge \neg a) \vee (d \wedge \neg c \wedge a)$,

$$N_e = (d \wedge c \wedge \neg a) \vee (\neg c \wedge \neg b \wedge a) \vee (\neg d \wedge \neg c \wedge a),$$

$$\begin{aligned} P_f = {} & (\neg e \wedge d \wedge c \wedge \neg a) \vee (\neg e \wedge \neg c \wedge \neg b \wedge a) \vee \\ & (d \wedge \neg c \wedge \neg b \wedge a) \vee (\neg e \wedge \neg d \wedge \neg c \wedge a) \vee (e \wedge d \wedge \neg c \wedge a) \end{aligned}$$

and

$$\begin{aligned} N_f = {} & (\neg e \wedge d \wedge c \wedge \neg b \wedge \neg a) \vee (e \wedge \neg d \wedge c \wedge \neg a) \vee \\ & (\neg e \wedge \neg c \wedge \neg b \wedge a) \vee (d \wedge \neg c \wedge \neg b \wedge a) \vee (e \wedge d \wedge \neg c \wedge a), \end{aligned}$$

is such that its interpretation is model-equivalent to (1) (i.e. $\beta^* \cong F$).

The following theorem establishes that for every QBF there exists a literal base such that its interpretation is model-equivalent to the QBF. By this theorem the literal base language may be considered as a target compilation language for QBF.

Theorem 1 (Completeness of LB). *Let QM be a QBF. Then there exists a literal base $lb \in \mathcal{B}_Q$ such that $lb^* \cong QM$.*

A **sat**-certificate for a QBF is easily extended to a literal base: we add the binder of the QBF and in the sequence of the **sat**-certificate for each universally quantified variable we add a couple $(\top, \top)$. Hence, the interpretation of the **sat**-certificate considered as a literal base has only one model which is the model of the QBF. In a **sat**-certificate $(\phi_1, \nu_1); \ldots; (\phi_p, \nu_p)$ formulae ϕ_i and ν_i, $1 \leq i \leq p$, only depend on the preceding universally quantified variables while in a literal base $(Q, (P_1, N_1); \ldots; (P_n, N_n))$ formulae P_i and N_i, $1 \leq i \leq n$, may depend on all the preceding variables.

The propositional fragment in which the propositional formulae of the literal bases are defined needs only to be complete and may be chosen w.r.t. its succinctness (see [14] for a survey on properties of propositional fragments).

When a QBF is considered to model a finite two-player game, the validity of the QBF means that the "existential" player is sure to win if he follows the moves obtained from the (sequence of formulae of the) model. We are interested in the following question: since until now I have followed a (sequence of formulae of a) model, can I change my mind for the next move? We call this problem the "next move choice problem" and we define it formally.

Definition 3 (Next move choice problem for a subset X of QBF).

- Instance : *A formula $q_1 x_1 \ldots q_n x_n M$ from a subset X of QBF, a sequence of substitutions $[x_1 \leftarrow C_1] \ldots [x_i \leftarrow C_i]$ obtained from a (sequence of formulae of a) model for $q_1 x_1 \ldots q_n x_n M$ with $q_i = \exists$ and $C_1, \ldots, C_i \in \{\top, \bot\}$.*
- Query: *Does there exist a model for $q_{i+1} \ldots q_n x_n [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \overline{C_i}](M)$.*

Clearly enough, the next move choice problem is still PSPACE-complete if we consider $X = QBF$.

Considering again the QBF (1) and one of its model (2), we know that $\forall d \exists e \exists f [a \leftarrow \top][b \leftarrow \top][c \leftarrow \bot](\mu)$ is valid (since $[a \leftarrow \top][b \leftarrow \top](\phi_c) \equiv \bot$) but is $\forall d \exists e \exists f [a \leftarrow \top][b \leftarrow \top][c \leftarrow \top](\mu)$ also valid?

We introduce a new property, called "optimality", for literal bases in order to exhibit a QBF fragment in which the next move choice problem is polytime w.r.t the size of the literal base.

Definition 4 (Optimality of a literal base). *Let lb be a literal base such that $lb = (q_1 x_1 \ldots q_n x_n, (P_1, N_1); \ldots; (P_n, N_n))$ and $lb^* = q_1 x_1 \ldots q_n x_n G$. The literal base lb is optimal if the following holds. For all i, $1 \leq i \leq n$, let $[x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}]$ be an interpretation such that for all k, $1 \leq k < i$ if $C_k = \top$ then $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](P_k)$ else $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](N_k)$.*

Then

$$\models [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}](P_i)$$
if and only if there exists a model for
$$q_{i+1} x_{i+1} \ldots q_n x_n [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \top](G)$$

and

$$\models [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}](N_i)$$
if and only if there exists a model for
$$q_{i+1}x_{i+1} \ldots q_n x_n [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \bot](G).$$

We denote by OBL the set of optimal literal bases.

Considering again (3), β is an optimal literal base. Since the interpretation of β is model-equivalent to (2) (i.e. $\forall a \forall b \exists c \forall d \exists e \exists f \mu$) and $[a \leftarrow \top][b \leftarrow \top](N_c) \equiv \bot$ the QBF $\forall d \exists e \exists f [a \leftarrow \top][b \leftarrow \top][c \leftarrow \top](\mu)$ is not valid.

The most important property of optimal literal bases is that the next move choice problem is polytime and no more PSPACE-complete.

Theorem 2. *The next move choice problem for OBL^* is polytime.*

If a QBF modeling a finite two-player game is compiled off-line in an optimal literal base, the computation of any sequence of moves leading to victory is polytime. An optimal literal base may be seen as a dynamic decision tree. The property of optimality of a literal base is linked with the property of minimality of a QBF which expresses that the QBF matrix contains only the models needed by the (QBF) models.

Definition 5 (Minimality of a QBF). *A QBF is minimal if all the (propositional) models of the matrix are (at least) in one of its (QBF) model.*

For example, the QBF $\exists a \forall b((a \wedge b) \vee (a \wedge \neg b) \vee (\neg a \wedge b))$ is not minimal since the (Boolean) model $\{\neg a, b\}$ of the matrix is not in the only one model $\psi_{\hat{a}} = \top$.

Theorem 3. *Let lb be an optimal literal base. Then lb^* is a minimal QBF.*

The converse of Theorem 3 is false: The literal base $(\exists a \forall b, (\top, \top), (a, a))$ is not optimal (since there is no model with $\psi_{\hat{a}} = \bot$) but its interpretation is minimal.

4 Literal base and sat-certificate for search-based algorithms

In this section we are interested in the following problem: how to extend a search-based procedure in order to compute directly the **sat**-certificate and not a posteriori from a trace. To do this we define an operator for literal bases in order to be able to build a **sat**-certificate from the **sat**-certificates of its two simpler subproblems.

Definition 6. *The operator $\circ_x : \mathcal{B}_Q \times \mathcal{B}_Q \rightarrow \mathcal{B}_{\forall x Q}$ is defined as follows :*

$$\begin{array}{l}(Q, (P_1, N_1); \ldots; (P_n, N_n)) \circ_x (Q, (P'_1, N'_1); \ldots; (P'_n, N'_n)) \\ = (\ \forall x Q, (\top, \top); \\ \quad (((\neg x \vee P_1) \wedge (x \vee P'_1)), ((\neg x \vee N_1) \wedge (x \vee N'_1))); \ldots; \\ \quad (((\neg x \vee P_n) \wedge (x \vee P'_n)), ((\neg x \vee N_n) \wedge (x \vee N'_n))))\end{array}$$

In this definition, if x is interpreted to **true** (resp. **false**) then for all i, $1 \leq i \leq n$, $((\neg x \vee P_i) \wedge (x \vee P'_i)) \equiv P_i$ (resp. P'_i) and $((\neg x \vee N_i) \wedge (x \vee N'_i)) \equiv N_i$ (resp. N'_i). If $(Q, (P_1, N_1); \ldots; (P_n, N_n))$ and $(Q, (P'_1, N'_1); \ldots; (P'_n, N'_n))$ are **sat**-certificates and $Q = q_1 x_1 \ldots q_n x_n$ with $q_i = \forall$ then clearly enough $((\neg x \vee P_i) \wedge (x \vee P'_i)) \equiv \top \equiv ((\neg x \vee N_i) \wedge (x \vee N'_i))$.

We establish by the following theorem that the $\circ$ operator composes two **sat**-certificates in a new **sat**-certificate.

Theorem 4. *Let $\forall x Q M$ be a QBF. If $lb_\top$ is a **sat**-certificate for $Q[x \leftarrow \top](M)$ and $lb_\bot$ is a **sat**-certificate for $Q[x \leftarrow \bot](M)$ then $(lb_\top \circ_x lb_\bot)$ is a **sat**-certificate for $\forall x Q M$.*

Algorithm 1 *search_certif_qbf*

In: Q : a binder of a QBF
In: M : a matrix of a QBF
Out: a **sat**-certificate or *non_valid*

```
if Q = qx then
 if q = ∃ then
  switch M do
   case ⊤ : return (∃x, (⊤, ⊥))
   case ⊥ : return non_valid
   case x : return (∃x, (⊤, ⊥))
   case ¬x : return (∃x, (⊥, ⊤))
  end switch
 else
  if M ≡ ⊤ then return (∀x, (⊤, ⊤)) else return non_valid end if
 end if
else
 Q = qxQ'
 lb+ := search_certif_qbf(Q', M[x ← ⊤])
 if lb+ = non_valid then
  if q = ∃ then
   lb− := search_certif_qbf(Q', M[x ← ⊥])
   if lb− = non_valid then return non_valid
   else return ((Q, (⊥, ⊤) ; grds(lb−)) end if
  else
   return non_valid
  end if
 else
  if q = ∃ then
   return (Q, (⊤, ⊥) ; grds(lb+))
  else
   lb− := search_certif_qbf(Q', M[x ← ⊥])
   if lb− = non_valid then return non_valid else return (lb+ ∘x lb−) end if
  end if
 end if
end if
```

We are now able to present the search-based algorithm *search_certif_qbf* which computes a **sat**-certificate for a QBF. The *search_certif_qbf* algorithm checks first if the binder is reduced to a single quantifier with its variable. In this case, if it is an existential quantifier four cases are possible, corresponding, in the order of the algorithm, to : $\exists x \top \equiv \exists x x$[1], $\exists x \bot \equiv \bot$, $\exists x x \cong \exists x((\neg x \vee \top) \wedge (x \vee \bot))$ and $\exists x \neg x \cong \exists x((\neg x \vee \bot) \wedge (x \vee \top))$. If the quantifier is universal then if $M \equiv \top$ then $\forall x M \equiv \top$ else $\forall x x \equiv \forall x \neg x \equiv \forall x \bot \equiv \bot$. If there are some quantifiers, since the algorithm is a search-based one, the most external quantifier is considered. If this quantifier is existential then if one of the recursive calls for the substitution by $\top$ (resp. $\bot$) for the variable x is different to *non_valid* the returned **sat**-certificate is $(Q, (\top, \bot); grds(lb^+))$ (resp. $(Q, (\bot, \top); grds(lb^-))$) which expresses that x must be **true** (resp. **false**). If the quantifier is universal then if at least one recursive call for the substitution by $\top$ or by $\bot$ for the variable x returns *non_valid* then *non_valid* is returned otherwise the Skolem functions of the two **sat**-certificates have to be combined to integrate the new argument x by $(lb^+ \circ_x lb^-)$ before this new **sat**-certificate is returned.

Theorem 5 (Correctness of *search_certif_qbf*(Q, M) *returns a* ***sat****-certificate for* QM *if the QBF is valid and non_valid otherwise.*

In case of search-based algorithms for CNF QBF, unit propagation and monotone literal propagation [11] may be easily added the *search_certif_qbf* algorithm.

5 Literal bases and QBF compilation for search-based algorithms

Since Theorem 1 establishes the completeness of the literal base language, LB may be considered as a target language for the compilation of a QBF. In this section we are interested in the following problem: how to extend a search-based procedure in order to compile a QBF in an optimal literal base. To do this we define an operator for literal bases which compile a QBF by the combination of the results of the compilation of its two simpler subproblems.

Definition 7. *Let* $Q' = q_2 x_2 \ldots q_n x_n$ *and* $Q = q_1 x_1 Q'$ *be two binders and* $lb, lb' \in \mathcal{B}_Q$*. The operator* $\oplus : \mathcal{B}_Q \times \mathcal{B}_Q \rightarrow \mathcal{B}_Q$ *is defined as follows :*

If $Q = \epsilon$ *then* $(lb \oplus lb') = (lb^* \vee lb'^*)$ *else*

$$
\begin{array}{l}
(Q, (P_1, N_1); \ldots; (P_n, N_n)) \oplus (Q, (P'_1, N'_1); \ldots; (P'_n, N'_n)) \\
= (Q, ((P_1 \vee P'_1), (N_1 \vee N'_1)); \\
\qquad (\mathcal{P}_2 \wedge (P'_2 \vee \mathcal{X}) \wedge (P_2 \vee \mathcal{X}'),\ \mathcal{N}_2 \wedge (N'_2 \vee \mathcal{X}) \wedge (N_2 \vee \mathcal{X}')); \ldots; \\
\qquad (\mathcal{P}_n \wedge (P'_n \vee \mathcal{X}) \wedge (P_n \vee \mathcal{X}'),\ \mathcal{N}_n \wedge (N'_n \vee \mathcal{X}) \wedge (N_n \vee \mathcal{X}')))
\end{array}
$$

[1] Since we need one solution, we privilege the interpretation of x to **true**

with $\mathcal{X} = ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1))$, $\mathcal{X}' = ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1))$ *and the recursive call:*

$$(Q', (\mathcal{P}_2, \mathcal{N}_2); \ldots; (\mathcal{P}_n, \mathcal{N}_n)) = \\ (Q', (P_2, N_2); \ldots; (P_n, N_n)) \oplus (Q', (P'_2, N'_2); \ldots; (P'_n, N'_n))$$

This operator is the counterpart of the disjunction for the QBF. In the previous definition when $n = 1$,

$$(q_1 x_1, (P_1, N_1)) \oplus (q_1 x_1, (P'_1, N'_1)) = (q_1 x_1, ((P_1 \vee P'_1), (N_1 \vee N'_1)))$$

which defines the base case of recursivity of $\oplus$. We develop for the case $n = 2$ the disjunction of the matrices of the interpretation of two literal bases and show how we can find back Definition 7: Since $(q_2 x_2, (P_2, N_2)) \oplus (q_2 x_2, (P'_2, N'_2)) = (q_2 x_2, ((P_2 \vee P'_2), (N_2 \vee N'_2)))$, $\mathcal{P}_2 = (P_2 \vee P'_2)$ and $\mathcal{N}_2 = (N_2 \vee N'_2)$ then

$$\begin{array}{l} ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1)) \wedge ((\neg x_2 \vee P_2) \wedge (x_2 \vee N_2)) \vee \\ ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1)) \wedge ((\neg x_2 \vee P'_2) \wedge (x_2 \vee N'_2)) \\ \equiv (\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1)) \wedge \\ \quad (\neg x_2 \vee ((P_2 \vee P'_2) \wedge (P'_2 \vee ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1))) \wedge (P_2 \vee ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1))))) \wedge \\ \quad (\neg x_2 \vee ((N_2 \vee N'_2) \wedge (N'_2 \vee ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1))) \wedge (N_2 \vee ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1))))) \\ \equiv (\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1)) \wedge \\ \quad (\neg x_2 \vee (\mathcal{P}_2 \wedge (P'_2 \vee \mathcal{X}) \wedge (P_2 \vee \mathcal{X}'))) \wedge (\neg x_2 \vee (\mathcal{N}_2 \wedge (N'_2 \vee \mathcal{X}) \wedge (N_2 \vee \mathcal{X}'))) \end{array}$$

Definition 7 may be improved with no cost by applying as simplification rules some usual logical equivalences: $(x \wedge x) \equiv x$, $(x \vee x) \equiv x$, $(x \wedge \neg x) \equiv \bot$ and $(x \vee \neg x) \equiv \top$ with x a propositional variable; $(H \wedge \top) \equiv H$, $(H \wedge \bot) \equiv \bot$, $(H \vee \top) \equiv \top$ and $(H \vee \bot) \equiv H$ with H a propositional formula.

Theorem 6. *Let Q be a binder and $lb, lb' \in \mathcal{B}_Q$ such that $lb^* = QM$ and $lb'^* = QM'$. Then $(lb \oplus lb')^* = QM_\oplus$ with $M_\oplus \equiv (M \vee M')$.*

We are now able to present the search-based algorithm *search_comp_qbf* which compiles a QBF into an optimal literal base. The *search_comp_qbf* algorithm checks first if the binder is reduced to a single quantifier with its variable. If it is the case and if $M \equiv \top$, conversely to *search_certif_qbf* algorithm, $(\top, \top)$ is returned (since $\exists x \top \cong \exists x((\neg x \vee \top) \wedge (x \vee \top))$) in order to compose the two possibilities. If there are some quantifiers, since the algorithm is a search-based one, the first one is considered. Following semantics of QBF, if there is no model for one (resp. both) recursive call then there is no model for the QBF if the quantifier is universal (resp. existential) ; if there are models for both recursive calls then, for both quantifiers, $(Q, (\top, \bot)\ ;\ grds(lb^+)) \oplus (Q, (\bot, \top)\ ;\ grds(lb^-)$ is returned.

Literal bases generated by the *search_comp_qbf* compilation algorithm may be in worst case of exponential size.

Theorem 7 (Correctness of *search_comp_qbf*). *Let QM be a QBF. $search\_comp\_qbf(Q, M)$ returns a literal base lb such that $lb^* \cong QM$ if QM is valid and returns non_valid otherwise.*

Algorithm 2 *search_comp_qbf*

In: Q : a binder of a QBF
In: M : a matrix of a QBF
Out: an optimal literal base or *non_valid*

```
if Q = qx then
 if q = ∃ then
  switch M do
   case ⊤ : return (∃x, (⊤, ⊤))
   case ⊥ : return non_valid
   case x : return (∃x, (⊤, ⊥))
   case ¬x : return (∃x, (⊥, ⊤))
  end switch
 else
  if M = ⊤ then return (∀x, (⊤, ⊤)) else return non_valid end if
 end if
else
 Q = qxQ'
 lb+ := search_comp_qbf(Q', M[x ← ⊤])
 lb− := search_comp_qbf(Q', M[x ← ⊥])
 if q = ∃ then
  if lb+ = non_valid and lb− = non_valid then return non_valid end if
  if lb+ = non_valid then return (Q, (⊥, ⊤) ; grds(lb−)) end if
  if lb− = non_valid then return (Q, (⊤, ⊥) ; grds(lb+)) end if
  return (Q, (⊤, ⊥) ; grds(lb+)) ⊕ (Q, (⊥, ⊤) ; grds(lb−))
 else
  if lb+ = non_valid or lb− = non_valid then
   return non_valid
  else
   return (Q, (⊤, ⊥) ; grds(lb+)) ⊕ (Q, (⊥, ⊤) ; grds(lb−))
  end if
 end if
end if
```

We can now establish that the literal base generated by the *search_comp_qbf* algorithm is optimal.

Theorem 8 (Optimality of *search_comp_qbf*). *Let QM be a valid QBF. Then search_comp_qbf(Q, M) is optimal.*

In case of search-based algorithms for CNF QBF, unit propagation may be easily added ; but, conversely to *search_certif_qbf* algorithm, monotone literal propagation can not be applied to the *search_comp_qbf* algorithm since it does not preserve all the models.

6 Related work

QBF certificates. To the best of our knowledge, there exist only two suggestions for QBF certificates and methods to generate them[2]. The first approach [20] is a method to generate a list of pairs of the form (v, f_v) where f_v are the Skolem functions for fresh variables v from the classical extension rule for propositional logic [26]. The second approach proposed in [4, 3] introduces the **sat**-certificate. It is described independently of any algorithm, but with binary decision diagrams (BDD) [8] and only for formulae in CNF. The computation of a **sat**-certificate is described in [2] in the framework of sKizzo as a reconstruction from a trace: the "inference log". An external certifier application (ozziKs) is charged with interpreting the content of the log in order to construct certificates [4]. Since the solver can choose between five different inference strategies there are many different kinds of instructions in the inference logs. It results in the need for a heavyweight proof checker. This approach is based on a trace of what the solver is doing and it probably does not scale well because of the growth of this trace. It can take more time to generate the **sat**-certificate from the trace than it took to generate the model [4].

QBF compilation. Knowledge compilation with a subset of the propositional language as a target language has been widely study (see [14] for a "knowledge compilation map"), but it is not the case for QBF compilation: [16] focuses on selected propositional fragments and quantifier elimination while [12] focuses on complexity of QBF built on the same selected propositional fragments. The compiler for CNF QBF proposed in [25] extends a quantifier-elimination decision procedure [22] as follows: for a CNF QBF $q_1x_1 \dots q_{n-1}x_{n-1}q_nx_nM$, we compute the formulae M_n, P_n and N_n defined on $\{x_1, \dots, x_{n-1}\}$ such that $M \equiv (M_n \wedge ((\neg x_n \vee P_n) \wedge (x_n \vee N_n)))$; if $q_n = \exists$ then the process is recursively called on $q_1x_1 \dots q_{n-1}x_{n-1}(M_n \wedge (P_n \vee N_n))$ otherwise the process is recursively called on $q_1x_1 \dots q_{n-1}x_{n-1}(M_n \wedge (P_n \wedge N_n))$. The target language of this approach is similar to the literal base language: $(q_1x_1 \dots q_nx_n, (P_1, N_1); \dots; (P_n, N_n))$ is a literal base. Since $(P_n \vee N_n)$ is not CNF, the expansion of the existential quantifier for CNF is involved with a quadratic size increase of the formula [21]. Clearly enough the literal base generated by this quantifier-elimination compiler is optimal and it is usually smaller than the literal base generated by the *search_comp_qbf* with out simplifications.

7 Concluding remarks

We have described in this article a unified framework for **sat**-certificate and compilation of QBF. We have proposed a search-based procedure to compute

[2] The approach proposed in [27] is a method to generate a subset of the clauses of a QBF formula in prenex normal form which is non-valid from traces of search-based solvers. Since this approach is focused on non-validity, it is out of the scope of this paper which is focused on validity.

sat-certificates which is very useful since most QBF solvers are search-based decision procedures.

Literal bases generated by the *search_comp_qbf* compilation algorithm may be in worst case of exponential size what complies with complexity results [13]. Anyway, we think that compilation is useful since all the solutions are kept and decision over existentially quantified variables may be not fully described in the QBF. In that case, for each existentially quantified variable, the two different possibilities are computed in polynomial time thanks to optimality and if both substitutions take part of a solution, the choice is left to the user, following its preferences.

Acknowledgement. We would like to thank the referees for their comments which helped improve this paper.

References

1. A. Ayari and D. Basin. Qubos: Deciding quantified boolean logic using propositional satisfiability solvers. In *Proceedings of the 4th International Conference on Formal Methods in Computer-Aided Design (FMCAD'02)*, pages 187–201, 2002.
2. M. Benedetti. Evaluating QBFs via Symbolic Skolemization. In *Proceedings of the 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'05)*, number 3452 in LNCS, pages 285–300. Springer, 2005.
3. M. Benedetti. Extracting certificates from quantified boolean formulas. In *Proceedings of 9th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 47–53, 2005.
4. M. Benedetti. skizzo: a suite to evaluate and certify QBFs. In *Proceedings of the 20th International Conference on Automated Deduction (CADE'05)*, pages 369–376, 2005.
5. M. Benedetti and H. Mangassarian. Experience and perspectives in qbf-based formal verification. *Journal on Satisfiability, Boolean Modeling and Computation*, 2008 (to appear).
6. A. Biere. Resolve and Expand. In *Proceedings of the 7th International Confrerence on Theory and Applications of Satisfiability Testing (SAT'04)*, pages 59–70, 2004.
7. L. Bordeaux. Boolean and interval propagation for quantified constraints. In *First International Workshop on Quantification in Constraint Programming*, 2005.
8. R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986.
9. H. K. Büning, M. Karpinski, and A. Flögel. Resolution for quantified boolean formulas. *Information and Computation*, 117(1):12–18, 1995.
10. H. K. Büning, K. Subramani, and X. Zhao. Boolean functions as models for quantified boolean formulas. *Journal of Automated Reasoning*, 39(1):49–75, 2007.
11. M. Cadoli, M. Schaerf, A. Giovanardi, and M. Giovanardi. An algorithm to evaluate quantified boolean formulae and its experimental evaluation. *Journal of Automated Reasoning*, 28(2):101–142, 2002.
12. S. Coste-Marquis, D. Le Berre, F. Letombe, and P. Marquis. Propositional fragments for knowledge compilation and quantified boolean formulae. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*, pages 288–293, 2005.

13. S. Coste-Marquis, H. Fargier, J. Lang, D. Le Berre, and P. Marquis. Representing policies for quantified boolean formulae. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 286–296, 2006.
14. A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
15. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communication of the ACM*, 5, 1962.
16. H. Fargier and P. Marquis. On the use of partially ordered decision graphs in knowledge compilation and quantified boolean formulae. In *AAAI*, 2006.
17. M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1990.
18. E. Giunchiglia, M. Narizzano, and A. Tacchella. Clause/term resolution and learning in the evaluation of quantified boolean formulas. *Journal of Artificial Intelligence Research*, 26:371–416, 2006.
19. F. Bacchus H. Samulowitz. Using SAT in QBF. In *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming, Lecture Notes in Computer Science*, volume 3709, pages 578 – 592, 2005.
20. T. Jussila, A. Biere, C. Sinz, D. Kröning, and C. Wintersteiger. A first step towards a unified proof checker for qbf. In *Proceedings of the Tenth International Confrerence on Theory and Applications of Satisfiability Testing (SAT'07)*, pages 201–214, 2007.
21. F. Lonsing and A. Biere. Nenofex: Expanding NNF for QBF Solving. In *Proceedings of the Eleventh International Conference on Theory and Applications of Satisfiability Testing (SAT'08)*, pages 196–210, 2008.
22. G. Pan and M.Y. Vardi. Symbolic Decision Procedures for QBF. In *International Conference on Principles and Practice of Constraint Programming*, 2004.
23. D.A. Plaisted, A. Biere, and Y. Zhu. A satisfiability procedure for quantified Boolean formulae. *Discrete Applied Mathematics*, 130:291–328, 2003.
24. J. Rintanen. Constructing conditional plans by a theorem-prover. *Journal of Artificial Intelligence Research*, 10:323–352, 1999.
25. I. Stéphan. Finding models for quantified boolean formulae. In *First International Workshop on Quantification in Constraint Programming*, 2005.
26. G. S. Tseitin. On the complexity of derivation in propositional calculus. In A. O. Slisenko, editor, *Studies in Constructive Mathematics and Mathematical Logic, Part 2*, pages 115–125. Consultants Bureau, New York, 1970.
27. Y. Yu and S. Malik. Validating the result of a quantified boolean formula (QBF) solver: theory and practice. In *Proceedings of the 2005 conference on Asia South Pacific design automation (ASP-DAC '05)*, pages 1047–1051, 2005.
28. L. Zhang. Solving QBF with combined conjunctive and disjunctive normal form. In *National Conference on Artificial Intelligence (AAAI'06)*, 2006.

8 Proofs

By abuse of notation we define the interpretation function for guards as follows:

$$((P_1, N_1); \ldots; (P_n, N_n))^* = \bigwedge_{1 \leq k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k))$$

and the $\oplus$ operator only for guards as follows:

$$\begin{array}{l} ((P_1, N_1); \ldots; (P_n, N_n)) \oplus ((P'_1, N'_1); \ldots; (P'_n, N'_n)) \\ = (((P_1 \vee P'_1), (N_1 \vee N'_1)); \\ \quad (\mathcal{P}_2 \wedge (P'_2 \vee \mathcal{X}) \wedge (P_2 \vee \mathcal{X}'),\ \mathcal{N}_2 \wedge (N'_2 \vee \mathcal{X}) \wedge (N_2 \vee \mathcal{X}')); \ldots; \\ \quad (\mathcal{P}_n \wedge (P'_n \vee \mathcal{X}) \wedge (P_n \vee \mathcal{X}'),\ \mathcal{N}_n \wedge (N'_n \vee \mathcal{X}) \wedge (N_n \vee \mathcal{X}'))) \end{array}$$

with $\mathcal{X} = ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1)), \mathcal{X}' = ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1))$ and

$$((\mathcal{P}_2, \mathcal{N}_2); \ldots; (\mathcal{P}_n, \mathcal{N}_n)) = ((P_2, N_2); \ldots; (P_n, N_n)) \oplus ((P'_2, N'_2); \ldots; (P'_n, N'_n))$$

Theorem 1 *Let QM be a QBF. Then there exists a literal base $lb \in \mathcal{B}_Q$ such that $lb^* \cong QM$.*

Proof of Theorem 1. This theorem is direct since every formula is equivalent to a conjunction of disjunctions of literals whose last variable can be distributed. □

Theorem 2 *The next move choice problem for OBL^* is polytime.*

Proof of Theorem 2. This theorem is a direct consequence of the definition of optimality for literal bases. □

Theorem 3 *Let lb be an optimal literal base. Then lb^* is a minimal QBF.*

Proof of Theorem 3. If the interpretation of the literal base is not minimal then there exists a (propositional) model for the matrix which is not in a sequence of satisfying Skolem functions then there exists a sequence of guards satisfied by this model but at least one of this guard should not be satisfied since the literal base is not optimal.

□

Theorem 4 *Let $\forall x QM$ be a QBF. If $lb_\top$ is a **sat**-certificate for $Q[x \leftarrow \top](M)$ and $lb_\bot$ is a **sat**-certificate for $Q[x \leftarrow \bot](M)$ then $(lb_\top \circ_x lb_\bot)$ is a **sat**-certificate for $\forall x QM$.*

Proof of Theorem 4. Let $\forall x_1 q_2 x_2 \ldots q_n x_n M$ be a QBF.
Let $B_\top = (Q, (P_2, N_2); \ldots; (P_n, N_n))$ be a **sat**-certificate for $Q[x \leftarrow \top](M)$ and $B_\bot = (Q, (P'_2, N'_2); \ldots; (P'_n, N'_n))$ be a **sat**-certificate for $Q[x \leftarrow \bot](M)$. If x_i, $2 \leq i \leq n$, is a universally quantified variable then $P_i = N_i = P'_i = N'_i = \top$ then $((\neg x_i \vee P_i) \wedge (x_i \vee P'_i)) \equiv \top$ and $((\neg x_i \vee N_i) \wedge (x_i \vee N'_i)) \equiv \top$. If x_i, $2 \leq i \leq n$, is

an existentially quantified variable with its associated function $\hat{x}_i^{\top}$ for $B_{\top}$ (the sequence of these Boolean functions $\hat{x}_i^{\top}$ satisfies $Q[x \leftarrow \top](M)$) and with its associated function $\hat{x}_i^{\bot}$ for $B_{\bot}$ (the sequence of these Boolean functions $\hat{x}_i^{\bot}$ satisfies $Q[x \leftarrow \bot](M)$) then the interpretation v associated to the Boolean function $\hat{x}_i$ such that $\hat{x}_i(\mathbf{true}) = \hat{x}_i^{\top}$ and $\hat{x}_i(\mathbf{false}) = \hat{x}_i^{\bot}$ verifies $v^*(((\neg x_i \vee P_i) \wedge (x_i \vee P_i'))) = \mathbf{true}$ and $v^*(((\neg x_i \vee N_i) \wedge (x_i \vee N_i'))) = \mathbf{false}$. Moreover, the sequence of these Boolean functions $\hat{x}_i$ satisfies $\forall x_1 q_2 x_2 \ldots q_n x_n M$.

□

Theorem 5 *Let QM be a QBF.*
*$search\_certif\_qbf(Q, M)$ returns a **sat**-certificate for QM if the QBF is valid and non_valid otherwise.*

Proof of Theorem 5.

- *Base case:* Let suppose that $Q = \exists x$.
  - If $M \equiv \top$ then $search\_certif\_qbf(\exists x, M)$ returns $B = (\exists x, (\top, \bot))$ then $B^* = \exists x((\neg x \vee \top) \wedge (x \vee \bot)) \cong \exists x x \equiv \exists x M$.
  - If $M \equiv \bot$ then $\exists x M$ is not valid and $search\_certif\_qbf(\exists x, M)$ returns *non_valid*.
  - If $M \equiv x$ then $search\_certif\_qbf(\exists x, M)$ returns $B = (\exists x, (\top, \bot))$ then $B^* = \exists x((\neg x \vee \top) \wedge (x \vee \bot)) \cong \exists x x \equiv \exists x M$.
  - If $M \equiv \neg x$ then $search\_certif\_qbf(\exists x, M)$ returns $B = (\exists x, (\bot, \top))$ then $B^* = \exists x((\neg x \vee \bot) \wedge (x \vee \top)) \cong \exists x \neg x \equiv \exists x M$.

  Now Let suppose that $Q = \forall x$. If $M \equiv \top$ then $search\_certif\_qbf(\forall x, M)$ returns $B = (\forall x, (\top, \top))$ then $B^* = \forall x((\neg x \vee \top) \wedge (x \vee \top)) \cong \forall x \top \cong \forall x M$. Else $\forall x M$ is not valid and $search\_certif\_qbf(\forall x, M)$ returns *non_valid*.
- *Induction case:* Let suppose that $Q = \exists x Q'$.
  Let
  $$bl^+ = search\_certif\_qbf(Q', [x \leftarrow \top](M))$$
  and
  $$bl^- = search\_certif\_qbf(Q', [x \leftarrow \bot](M)).$$
  - We suppose that $bl^+ = non\_valid$ and $bl^- = non\_valid$ then $search\_certif\_qbf(\exists x Q', M)$ returns *non_valid*. By induction hypothesis both QBF $Q'[x \leftarrow \top](M)$ and $Q'[x \leftarrow \bot](M)$ are not valid then by definition $\exists x Q' M$ is not valid.
  - We suppose that $bl^+ \neq non\_valid$ (the case is similar for ($bl^+ = non\_valid$ and $bl^- \neq non\_valid$)) then by induction hypothesis $bl^+ = search\_certif\_qbf(Q', [x \leftarrow \top](M))$ is a **sat**-certificate for $Q'[x \leftarrow \top](M)$ and $(\exists x Q', (\top, \bot), grds(bl^+))$ is a **sat**-certificate for $\exists x Q' M$.

  Now let suppose that $Q = \forall x Q'$.
  If
  $$search\_certif\_qbf(Q', [x \leftarrow \top](M)) \text{ returns } non\_valid$$
  or
  $$search\_certif\_qbf(Q', [x \leftarrow \top](M)) \text{ returns } non\_valid$$

then

$$search\_certif\_qbf(\forall x Q', M) \text{ returns } non\_valid.$$

By induction hypothesis one of the two QBF $Q'[x \leftarrow \top](M)$ or $Q'[x \leftarrow \bot](M)$ is not valid then by definition $\forall x Q' M$ is not valid. Otherwise, by induction hypothesis $bl^+ = search\_certif\_qbf(Q', [x \leftarrow \top](M))$ is a **sat**-certificate for $Q'[x \leftarrow \top](M)$ and $bl^- = search\_certif\_qbf(Q', [x \leftarrow \bot](M))$ is a **sat**-certificate for $Q'[x \leftarrow \bot](M)$ then by Theorem 4 $(bl^+ \circ_x bl^-)$ is a **sat**-certificate for $\forall x Q' M$.

□

We need also two technical lemmas in order to prove Theorem 6.

Lemma 1.

$$\begin{array}{l}((A \vee ((P_1, N_1); \ldots; (P_n, N_n))^*) \wedge (B \vee ((P_1, N_1); \ldots; (P_n, N_n))^*)) \\ \equiv (((P_1 \vee (A \wedge B)), (N_1 \vee (A \wedge B))); \ldots; ((P_n \vee (A \wedge B)), (N_n \vee (A \wedge B))))^*\end{array}$$

Proof of Lemma 1.

$$\begin{array}{l}((A \vee ((P_1, N_1); \ldots; (P_n, N_n))^*) \wedge (B \vee ((P_1, N_1); \ldots; (P_n, N_n))^*)) \\ \equiv ((A \vee \bigwedge_{1 \leq i \leq n} ((\neg x_i \vee P_i) \wedge (x_i \vee N_i))) \wedge (B \vee \bigwedge_{1 \leq i \leq n} ((\neg x_i \vee P_i) \wedge (x_i \vee N_i)))) \\ \equiv ((A \wedge B) \vee \bigwedge_{1 \leq i \leq n} ((\neg x_i \vee P_i) \wedge (x_i \vee N_i))) \\ \equiv \bigwedge_{1 \leq i \leq n} (((\neg x_i \vee P_i) \vee (A \wedge B)) \wedge ((x_i \vee N_i) \vee (A \wedge B))) \\ \equiv \bigwedge_{1 \leq i \leq n} ((\neg x_i \vee (P_i \vee (A \wedge B))) \wedge (x_i \vee (N_i \vee (A \wedge B)))) \\ \equiv (((P_1 \vee (A \wedge B)), (N_1 \vee (A \wedge B))); \ldots; ((P_n \vee (A \wedge B)), (N_n \vee (A \wedge B))))^*\end{array}$$

□

Lemma 2.

$$\begin{array}{l}(((P_1, N_1); \ldots; (P_n, N_n))^* \wedge ((P'_1, N'_1); \ldots; (P'_n, N'_n))^*) \\ \equiv (((P_1 \wedge P'_1), (N_1 \wedge N'_1)); \ldots; ((P_n \wedge P'_n), (N_n \wedge N'_n)))^*\end{array}$$

Proof of Lemma 2.

$$\begin{array}{l}(((P_1, N_1); \ldots; (P_n, N_n))^* \wedge ((P'_1, N'_1); \ldots; (P'_n, N'_n))^*) \\ \equiv (\bigwedge_{1 \leq i \leq n} ((\neg x_i \vee P_i) \wedge (x_i \vee N_i)) \wedge \bigwedge_{1 \leq i \leq n} ((\neg x_i \vee P'_i) \wedge (x_i \vee N'_i))) \\ \equiv \bigwedge_{1 \leq i \leq n} (((\neg x_i \vee P_i) \wedge (x_i \vee N_i)) \wedge ((\neg x_i \vee P'_i) \wedge (x_i \vee N'_i))) \\ \equiv \bigwedge_{1 \leq i \leq n} (((\neg x_i \vee P_i) \wedge (\neg x_i \vee P'_i)) \wedge ((x_i \vee N_i) \wedge (x_i \vee N'_i))) \\ \equiv \bigwedge_{1 \leq i \leq n} ((\neg x_i \vee (P_i \wedge P'_i)) \wedge (x_i \vee (N_i \wedge N'_i))) \\ \equiv (((P_1 \wedge P'_1), (N_1 \wedge N'_1)); \ldots; ((P_n \wedge P'_n), (N_n \wedge N'_n)))^*\end{array}$$

□

Theorem 6 *Let Q be a binder and $lb, lb' \in \mathcal{B}_Q$ such that $lb^* = QM$ and $lb'^* = QM'$. Then $(lb \oplus lb')^* = QM_\oplus$ with $M_\oplus \equiv (M \vee M')$.*

Proof of Theorem 6. The theorem holds by definition for the case of an empty binder. Theorem 6 is a direct consequence of the following lemma: Let Q be a non empty binder and $B, B' \in \mathcal{B}_Q$ such that $B = (Q, G)$ and $B' = (Q, G')$ then $(G \oplus G')^* \equiv (G^* \vee G'^*)$.

The proof of this lemma is by induction.

– *Base case:* $Q = q_1 x_1$.
Let $B = (q_1 x_1, (P_1, N_1))$ and $B' = (q_1 x_1, (P'_1, N'_1))$. Then

$$\begin{aligned}&((P_1, N_1)^* \vee (P'_1, N'_1)^*)\\ &= (((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1)) \vee ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1)))\\ &\equiv ((\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1)))\end{aligned} \tag{1}$$

By definition $((P_1, N_1) \oplus (P'_1, N'_1)) = ((P_1 \vee P'_1), (N_1 \vee N'_1))$ then

$$\begin{aligned}&((P_1, N_1) \oplus (P'_1, N'_1))^*\\ &= ((\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1)))\\ &= ((P_1, N_1)^* \vee (P'_1, N'_1)^*)\end{aligned}$$

– *Induction case:* $Q = q_1 x_1 \dots q_n x_n$.
Let

$$B = (q_1 x_1 \dots q_n x_n, (P_1, N_1); \dots; (P_n, N_n))$$

and

$$B' = (q_1 x_1 \dots q_n x_n, (P'_1, N'_1); \dots; (P'_n, N'_n))$$

Then by Definition 2

$$\begin{aligned}&((P_1, N_1) \dots (P_n, N_n))^*\\ &= \textstyle\bigwedge_{1 \le k \le n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k))\\ &= ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1)) \wedge [(P_2, N_2); \dots; (P_n, N_n)]^*\end{aligned} \tag{2}$$

and

$$\begin{aligned}&((P'_1, N'_1) \dots (P'_n, N'_n))^*\\ &= \textstyle\bigwedge_{1 \le k \le n} ((\neg x_k \vee P'_k) \wedge (x_k \vee N'_k))\\ &= ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1)) \wedge [(P'_2, N'_2); \dots; (P'_n, N'_n)]^*\end{aligned} \tag{3}$$

Then from (2) and (3)

$$\begin{aligned}&(((P_1, N_1); \dots; (P_n, N_n))^* \vee ((P'_1, N'_1); \dots; (P'_n, N'_n))^*)\\ &\equiv ((\neg x_1 \vee (P_1 \vee P'_1)) \wedge (x_1 \vee (N_1 \vee N'_1))) \wedge\\ &\quad ((\neg x_1 \vee P_1) \vee [(P'_2, N'_2); \dots; (P'_n, N'_n)]^*) \wedge\\ &\quad ((x_1 \vee N_1) \vee [(P'_2, N'_2); \dots; (P'_n, N'_n)]^*) \wedge\\ &\quad ((\neg x_1 \vee P'_1) \vee [(P_2, N_2); \dots; (P_n, N_n)]^*) \wedge\\ &\quad ((x_1 \vee N'_1) \vee [(P_2, N_2); \dots; (P_n, N_n)]^*) \wedge\\ &\quad ([(P_2, N_2); \dots; (P_n, N_n)]^* \vee [(P'_2, N'_2); \dots; (P'_n, N'_n)]^*)\end{aligned} \tag{4}$$

Let

$$\mathcal{X} = ((\neg x_1 \vee P_1) \wedge (x_1 \vee N_1)) \tag{5}$$

and

$$\mathcal{X}' = ((\neg x_1 \vee P'_1) \wedge (x_1 \vee N'_1)) \tag{6}$$

Then from (4), (5), (6) and Lemma 1

$$
\begin{array}{l}
(((P_1,N_1);\ldots;(P_n,N_n))^*\vee((P'_1,N'_1);\ldots;(P'_n,N'_n))^*)\\
\equiv((\neg x_1\vee(P_1\vee P'_1))\wedge(x_1\vee(N_1\vee N'_1)))\wedge\\
\quad[((P'_2\vee\mathcal{X}),(N'_2\vee\mathcal{X}));\ldots;((P'_n\vee\mathcal{X}),(N'_n\vee\mathcal{X}))]^*\wedge\\
\quad[((P_2\vee\mathcal{X}'),(N_2\vee\mathcal{X}'));\ldots;((P_n\vee\mathcal{X}'),(N_n\vee\mathcal{X}'))]^*\wedge\\
\quad([(P_2,N_2);\ldots;(P_n,N_n)]^*\vee[(P'_2,N'_2);\ldots;(P'_n,N'_n)]^*)
\end{array}
\tag{7}
$$

Let

$$
\begin{array}{l}
(\mathcal{P}_2,\mathcal{N}_2);\ldots;(\mathcal{P}_n,\mathcal{N}_n)\\
=((P_2,N_2);\ldots;(P_n,N_n))\oplus((P'_2,N'_2);\ldots;(P'_n,N'_n))
\end{array}
$$

By induction hypothesis

$$
\begin{array}{l}
[(\mathcal{P}_2,\mathcal{N}_2);\ldots;(\mathcal{P}_n,\mathcal{N}_n)]^*\\
=([(P_2,N_2);\ldots;(P_n,N_n)]^*\vee[(P'_2,N'_2);\ldots;(P'_n,N'_n)]^*)
\end{array}
\tag{8}
$$

Then from (7), (8) and Lemma 2

$$
\begin{array}{l}
(((P_1,N_1);\ldots;(P_n,N_n))^*\vee((P'_1,N'_1);\ldots;(P'_n,N'_n))^*)\\
\equiv((\neg x_1\vee(P_1\vee P'_1))\wedge(x_1\vee(N_1\vee N'_1)))\wedge\\
\quad[((P'_2\vee\mathcal{X}),(N'_2\vee\mathcal{X}));\ldots;((P'_n\vee\mathcal{X}),(N'_n\vee\mathcal{X}))]^*\wedge\\
\quad[((P_2\vee\mathcal{X}'),(N_2\vee\mathcal{X}'));\ldots;((P_n\vee\mathcal{X}'),(N_n\vee\mathcal{X}'))]^*\wedge\\
\quad[(\mathcal{P}_2,\mathcal{N}_2);\ldots;(\mathcal{P}_n,\mathcal{N}_n)]^*\\
\equiv((\neg x_1\vee(P_1\vee P'_1))\wedge(x_1\vee(N_1\vee N'_1)))\wedge\\
\quad[((((P'_2\vee\mathcal{X})\wedge(P_2\vee\mathcal{X}'))\wedge\mathcal{P}_2),(((N'_2\vee\mathcal{X})\wedge(N_2\vee\mathcal{X}'))\wedge\mathcal{N}_2));\ldots;\\
\quad((((P'_n\vee\mathcal{X})\wedge(P_n\vee\mathcal{X}'))\wedge\mathcal{P}_n),(((N'_n\vee\mathcal{X})\wedge(N_n\vee\mathcal{X}'))\wedge\mathcal{N}_n))]^*\\
\equiv(((P_1,N_1);\ldots;(P_n,N_n))\oplus((P'_1,N'_1);\ldots;(P'_n,N'_n)))^*
\end{array}
$$

□

Theorem 7 *Let QM be a QBF.*
$search\_comp\_qbf(Q,M)$ returns a literal base lb such that $lb^*\cong QM$ if QM is valid and returns non_valid otherwise.

Proof of Theorem 7.

- *Base cases:*
  - We suppose that $q=\forall$.
    - If $M\equiv\top$ then $search\_comp\_qbf(\forall x,M)=(\forall x(\top,\top))$ then

      $$search\_comp\_qbf(\forall x,M)^*=\forall x((\neg x\vee\top)\wedge(x\vee\top))\cong\forall x\top\cong\forall xM$$

    - If $M\equiv x$ ($M\equiv\neg x$ and $M\equiv\bot$ are similar) then $\forall xM$ is not valid and

      $$search\_comp\_qbf(\forall x,M)=non\_valid$$

  - We suppose that $q=\exists$.
    - If $M\equiv\top$ then

      $$search\_comp\_qbf(\exists x,M)=(\exists x(\top,\top))$$

      then

      $$search\_comp\_qbf(\exists x,M)^*=\exists x((\neg x\vee\top)\wedge(x\vee\top))\cong\exists x\top\cong\exists xM$$

* If $M \equiv x$ then

$$search\_comp\_qbf(\exists x, M) = (\exists x(\top, \bot))$$

then

$$search\_comp\_qbf(\exists x, M)^* = \exists x((\neg x \vee \top) \wedge (x \vee \bot)) \cong \exists x x \cong \exists x M$$

* If $M \equiv \neg x$ then

$$search\_comp\_qbf(\exists x, M) = (\exists x(\bot, \top))$$

then

$$search\_comp\_qbf(\exists x, M)^* = \exists x((\neg x \vee \bot) \wedge (x \vee \top)) \cong \exists x \neg x \cong \exists x M$$

* If $M \equiv \bot$ then $\exists x M$ is not valid and

$$search\_comp\_qbf(\exists x, M) = non\_valid$$

– *Induction cases:* Let $Q = qxQ'$ and

$$bl^+ = search\_comp\_qbf(Q', [x \leftarrow \top](M))$$

and

$$bl^- = search\_comp\_qbf(Q', [x \leftarrow \bot](M))$$

- We suppose that ($bl^+ = non\_valid$) and ($bl^- = non\_valid$). Then by induction hypothesis $Q'[x \leftarrow \top](M)$ and $Q'[x \leftarrow \bot](M)$ are not valid then QM is not valid and $search\_comp\_qbf(Q, M)$ returns non_valid.
- We suppose that ($bl^+ \neq non\_valid$) and ($bl^- = non\_valid$). By induction hypothesis,

$$(bl^+)^* \cong Q'[x \leftarrow \top](M) \tag{1}$$

If $q = \forall$ it is similar to the case "($bl^+ = non\_valid$) and ($bl^- = non\_valid$)". Otherwise $q = \exists$ and by induction hypothesis

$$Q'[x \leftarrow \bot](M) \text{ is not valid.} \tag{2}$$

Now let

$$\begin{aligned} &QM_+ \\ &= (Q, (\top, \bot)\ ;\ grds(bl^+))^* \\ &= search\_comp\_qbf(Q, M)^* \end{aligned} \tag{3}$$

and

$$Q'M^+ = (bl^+)^* \tag{4}$$

Since by Definition 2

$$M_+ = (\neg x \vee \top) \wedge (x \vee \bot) \wedge M^+$$

then

$$[x \leftarrow \top](M_+) \equiv M^+ \text{ and } [x \leftarrow \bot](M_+) \equiv \bot$$

then

$$Q'[x \leftarrow \top](M_+) \cong Q'M^+ \tag{5}$$

and

$$Q'[x \leftarrow \bot](M_+) \text{ is not valid} \tag{6}$$

then with (5), (4) and (1)

$$Q'[x \leftarrow \top](M_+) \cong Q'[x \leftarrow \top](M)$$

then with (6) and (2)

$$QM_+ \cong QM$$

then with (3)

$$search\_comp\_qbf(Q, M)^* \cong QM.$$

- We suppose that $(bl^+ = non\_valid)$ and $(bl^- \neq non\_valid)$.
  The case is similar to the previous case.
- We suppose that $(bl^+ \neq non\_valid)$ and $(bl^- \neq non\_valid)$. By induction hypothesis,

$$(bl^+)^* \cong Q'[x \leftarrow \top](M) \tag{7}$$

and

$$(bl^-)^* \cong Q'[x \leftarrow \bot](M) \tag{8}$$

Let

$$QM_+ = (Q, (\top, \bot)\ ;\ grds(bl^+))^* \tag{9}$$

$$QM_- = (Q, (\bot, \top)\ ;\ grds(bl^-))^* \tag{10}$$

and

$$\begin{array}{l} QM_\oplus \\ = ((Q, (\top, \bot)\ ;\ grds(bl^+)) \oplus (Q, (\bot, \top)\ ;\ grds(bl^-)))^* \\ = search\_comp\_qbf(Q, M)^* \end{array} \tag{11}$$

By Theorem 6,

$$M_\oplus \equiv (M_+ \vee M_-) \tag{12}$$

Let

$$Q'M^+ = (bl^+)^* \tag{13}$$

and

$$Q'M^- = (bl^-)^* \tag{14}$$

By Definition 2, (9) and (13)

$$M_+ = (\neg x \vee \top) \wedge (x \vee \bot) \wedge M^+$$

then

$$[x \leftarrow \top](M_+) \equiv M^+ \tag{15}$$

and

$$[x \leftarrow \bot](M_+) \equiv \bot \tag{16}$$

By Definition 2, (10) and (14)

$$M_- = (\neg x \vee \bot) \wedge (x \vee \top) \wedge M^-$$

then

$$[x \leftarrow \top](M_-) \equiv \bot \tag{17}$$

and

$$[x \leftarrow \bot](M_-) \equiv M^- \tag{18}$$

Then from (12), (15) and (17)

$$[x \leftarrow \top](M_\oplus) \equiv M^+ \tag{19}$$

and from (12), (16) and (18)

$$[x \leftarrow \bot](M_\oplus) \equiv M^- \tag{20}$$

Then from (19)

$$Q'[x \leftarrow \top](M_\oplus) \cong Q'M^+ \tag{21}$$

and from (20)

$$Q'[x \leftarrow \bot](M_\oplus) \cong Q'M^- \tag{22}$$

Then from (21), (13) and (7)

$$Q'[x \leftarrow \top](M_\oplus) \cong Q'[x \leftarrow \top](M)$$

and from (22), (14) and (8)

$$Q'[x \leftarrow \bot](M_\oplus) \cong Q'[x \leftarrow \bot](M)$$

then

$$QM_\oplus \cong QM$$

then from (11)

$$search\_comp\_qbf(Q, M)^* \cong QM.$$

□

We need X lemmas in order to prove Theorem 8. In what follows $Q^i = q_1 x_1 \ldots q_i x_i$ and $Q_i = q_i x_i \ldots q_n x_n$.

Lemma 3. *Let QM and QM' be two QBF such that $QM \cong QM'$, i be an integer, $1 \leq i \leq n$, $\{y_1, \ldots, y_p\}$ be the existentially quantified variables of $\{x_1, \ldots, x_{i-1}\}$, $v_{i-1} = [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}]$ be an interpretation such that there exists a satisfying Skolem function sequence $\hat{y_1}; \ldots; \hat{y_p}$; s for $Q^{i-1}Q_iM$ and v_{i-1} is in $\hat{y_1}; \ldots; \hat{y_p}$. Then $Q_i v_{i-1}(M) \cong Q_i v_{i-1}(M')$.*

Proof of Lemma 3.

– Base case: Obvious for $i = 1$.
– Induction case:
  • $q_i = \exists$. Let $\hat{y_1}; \ldots; \hat{y_p}; \hat{x_i}; s$ be a satisfying Skolem function sequence for $Q^{i-1}\exists x_i Q_{i+1} M$ then by induction hypothesis

$$\exists x_i Q_{i+1} v_{i-1}(M) \cong \exists x_i Q_{i+1} v_{i-1}(M')$$

  We suppose that $\hat{x_i}$ for v_{i-1} is equal to **true** (the proof is similar with **false**). Then, $v_i = [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}][x_i \leftarrow \top]$ is such that $\hat{y_1}; \ldots; \hat{y_p}; \hat{x_i}; s$ is a satisfying Skolem function sequence for $Q^{i-1}\exists x_i Q_{i+1} M$ and v_i is in $\hat{y_1}; \ldots; \hat{y_p}; \hat{x_i}$ and

$$Q_{i+1} v_{i-1}[x_i \leftarrow \top](M) \cong Q_{i+1} v_{i-1}[x_i \leftarrow \top](M')$$

  • $q_i = \forall$. Let $\hat{y_1}; \ldots; \hat{y_p}; s$ be such a satisfying Skolem function sequence for $Q^{i-1}\forall x_i Q_{i+1} M$ then by induction hypothesis

$$\forall x_i Q_{i+1} v_{i-1}(M) \cong \forall x_i Q_{i+1} v_{i-1}(M')$$

  Then, $\hat{y_1}; \ldots; \hat{y_p}; s$ is a satisfying Skolem function sequence for $Q^{i-1}\forall x_i Q_{i+1} M$,

$$Q_{i+1} v_{i-1}[x_i \leftarrow \top](M) \cong Q_{i+1} v_{i-1}[x_i \leftarrow \top](M')$$

  and

$$Q_{i+1} v_{i-1}[x_i \leftarrow \bot](M) \cong Q_{i+1} v_{i-1}[x_i \leftarrow \bot](M').$$

□

Lemma 4. *Let QM be a valid QBF and*

$$(Q, (P_1, N_1); \ldots; (P_n, N_n)) = search\_comp\_qbf(Q, M).$$

Let i be an integer, $1 \leq i \leq n$, $\{y_1, \ldots, y_p\}$ be the existentially quantified variables of $\{x_1, \ldots, x_{i-1}\}$ and $v_{i-1} = [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}]$ be an interpretation such that there exists a satisfying Skolem function sequence $\hat{y_1}; \ldots; \hat{y_p}; s$ for $Q^{i-1}Q_i M$ and v_{i-1} is in $\hat{y_1}; \ldots; \hat{y_p}$ and for all j, $1 \leq j < i$, if $C_j = \top$ then $\models [x_1 \leftarrow C_1] \ldots [x_{j-1} \leftarrow C_{j-1}](P_j)$ else $\models [x_1 \leftarrow C_1] \ldots [x_{j-1} \leftarrow C_{j-1}](N_j)$.
Then

$$search\_comp\_qbf(Q_i, v_{i-1}(M))^* \cong Q_i(\textstyle\bigwedge_{i \leq k \leq n} ((\neg x_k \vee v_{i-1}(P_k)) \wedge (x_k \vee v_{i-1}(N_k))))$$

Proof of Lemma 4.
By Theorem 7

$$search\_comp\_qbf(Q, M)^* = Q \bigwedge_{i \leq k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k))) \cong QM$$

Then by Lemma 3

$$Q_i v_{i-1}(\bigwedge_{i \leq k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k))) \cong Q_i v_{i-1}(M) \quad (1)$$

But, by definition of v_{i-1}

$$\begin{array}{l} v_{i-1}(\bigwedge_{1\le k\le n}((\neg x_k \vee P_k)\wedge(x_k\vee N_k))) \\ \equiv \bigwedge_{i\le k\le n}((\neg x_k\vee v_{i-1}(P_k))\wedge(x_k\vee v_{i-1}(N_k))) \end{array}$$

Then

$$\begin{array}{l} Q_i v_{i-1}(\bigwedge_{1\le k\le n}((\neg x_k \vee P_k)\wedge(x_k\vee N_k))) \\ \cong Q_i \bigwedge_{i\le k\le n}((\neg x_k\vee v_{i-1}(P_k))\wedge(x_k\vee v_{i-1}(N_k))) \end{array} \quad (2)$$

From (1) and (2)

$$\begin{array}{l} search\_comp\_qbf(Q_i, v_{i-1}(M))^* \\ \cong Q_1 v_{i-1}(M) \\ \cong Q_i \bigwedge_{i\le k\le n}((\neg x_k\vee v_{i-1}(P_k))\wedge(x_k\vee v_{i-1}(N_k))) \end{array}$$

□

Lemma 5. *Let QM be a valid QBF and*

$$(Q, (P_1, N_1); \ldots; (P_n, N_n)) = search\_comp\_qbf(Q, M).$$

Let i be an integer, $1 \le i \le n$, $\{y_1, \ldots, y_p\}$ be the existentially quantified variables of $\{x_1, \ldots, x_{i-1}\}$, $v_{i-1} = [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}]$ be an interpretation such that there exists a satisfying Skolem function sequence $\hat{y_1}; \ldots; \hat{y_p}; s$ for $Q^{i-1}Q_iM$, v_{i-1} is in $\hat{y_1}; \ldots; \hat{y_p}$ and for all k, $1 \le k < i$, if $C_k = \top$ then $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](P_k)$ else $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](N_k)$. Let

$$(Q_i, (P_i^i, N_i^i); \ldots; (P_n^i, N_n^i)) = search\_comp\_qbf(Q_i, v_{i-1}(M)).$$

Then for all j, $i \le j \le n$, $v_{i-1}(P_j) \equiv P_j^i$ and $v_{i-1}(N_j) \equiv N_j^i$.

Proof of Lemma 5. By Lemma 4,

$$\begin{array}{l} search\_comp\_qbf(Q_i, v_{i-1}(M))^* \\ \cong Q_i(\bigwedge_{i\le k\le n}((\neg x_k\vee v_{i-1}(P_k))\wedge(x_k\vee v_{i-1}(N_k)))) \end{array}$$

By Theorem 7,

$$\begin{array}{l} search\_comp\_qbf(Q_i, v_{i-1}(M))^* \\ = Q_i(\bigwedge_{i\le k\le n}((\neg x_k\vee P_k^i)\wedge(x_k\vee N_k^i))) \end{array}$$

Then, by induction, for all j, $i \le j \le n$, $v_{i-1}(P_j) \equiv P_j^i$ and $v_{i-1}(N_j) \equiv N_j^i$.
□

Theorem 8 *Let QM be a valid QBF. Then $search\_comp\_qbf(Q, M)$ is optimal.*

Proof of Theorem 8. We have to prove: Let $Q_1 M$ be a QBF and

$$(Q_1, (P_1, N_1); \ldots; (P_n, N_n)) = search\_comp\_qbf(Q_1, M).$$

For all i, $1 \leq i \leq n$, let $v_{i-1} = [x_1 \leftarrow C_1] \ldots [x_{i-1} \leftarrow C_{i-1}]$ be an interpretation such that for all k, $1 \leq k < i$ if $C_k = \top$ then $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](P_k)$ else $\models [x_1 \leftarrow C_1] \ldots [x_{k-1} \leftarrow C_{k-1}](N_k)$.

Then

$\models v_{i-1}(P_i)$
if and only if there exists a sequence of satisfying Skolem functions for $Q_{i+1} v_{i-1}[x_i \leftarrow \top](\bigwedge_{1 \leq k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k)))$

and

$\models v_{i-1}(N_i)$
if and only if there exists a sequence of satisfying Skolem functions for $Q_{i+1} v_{i-1}[x_i \leftarrow \bot](\bigwedge_{1 \leq k \leq n} ((\neg x_k \vee P_k) \wedge (x_k \vee N_k)))$.

By the properties of v_{i-1}, these consequences are equivalent to:

$\models v_{i-1}(P_i)$
if and only if there exists a sequence of satisfying Skolem functions for $Q_{i+1} \bigwedge_{i \leq k \leq n} ((\neg x_k \vee v_{i-1}[x_i \leftarrow \top](P_k)) \wedge (x_k \vee v_{i-1}[x_i \leftarrow \top](N_k)))$

and

$\models v_{i-1}(N_i)$
if and only if there exists a sequence of satisfying Skolem functions for $Q_{i+1} \bigwedge_{i \leq k \leq n} ((\neg x_k \vee v_{i-1}[x_i \leftarrow \bot](P_k)) \wedge (x_k \vee v_{i-1}[x_i \leftarrow \bot](N_k)))$.

And, by Lemma 4, these consequences are equivalent to

$\models v_{i-1}(P_i)$
if and only if there exists a sequence of satisfying Skolem functions for $search\_comp\_qbf(Q_{i+1}, v_{i-1}[x_i \leftarrow \top](M))$

and

$\models v_{i-1}(N_i)$
if and only if there exists a sequence of satisfying Skolem functions for $search\_comp\_qbf(Q_{i+1}, v_{i-1}[x_i \leftarrow \bot](M))$.

We prove for $[x_i \leftarrow \bot]$, the proof is similar for $\top$.
By Theorem 7,

$$Q_1 M \cong Q_1 \bigwedge_{1 \leq j \leq n} ((\neg x_j \vee P_j) \wedge (x_j \vee N_j))$$

then

$$Q_i v_{i-1}(M) \cong Q_i v_{i-1}(\bigwedge_{1 \leq j \leq n} ((\neg x_j \vee P_j) \wedge (x_j \vee N_j)))$$

then, by definition of v_{i-1},

$$\begin{array}{l} Q_i v_{i-1}(M) \\ \cong Q_1((\neg x_i \vee v_{i-1}(P_i)) \wedge (x_i \vee v_{i-1}(N_i))) \\ \wedge \bigwedge_{i+1 \leq j \leq n} ((\neg x_j \vee v_{i-1}(P_j)) \wedge (x_j \vee v_{i-1}(N_j))) \end{array}$$

then there exists a sequence of satisfying Skolem functions for $Q_1 v_{i-1}[x_i \leftarrow \bot](M)$ if and only if there exists a sequence of satisfying Skolem functions for

$$Q_i(v_{i-1}(N_i) \wedge \bigwedge_{i+1 \leq j \leq n} ((\neg x_j \vee v_{i-1}[x_i \leftarrow \bot](P_j)) \wedge (x_j \vee v_{i-1}[x_i \leftarrow \bot](N_j))))$$

if and only if $\models v_{i-1}(N_i)$ and there exists a sequence of satisfying Skolem functions for

$$Q_{i+1} \bigwedge_{i+1 \leq j \leq n} ((\neg x_j \vee v_{i-1}[x_i \leftarrow \bot](P_j)) \wedge (x_j \vee v_{i-1}[x_i \leftarrow \bot](N_j)))$$

then if $\not\models v_{i-1}(N_i)$ then there is no sequence of satisfying Skolem functions for $Q_{i+1} v_{i-1}[x_i \leftarrow \bot](M)$ then if there exists a sequence of satisfying Skolem functions for $Q_{i+1} v_{i-1}[x_i \leftarrow \bot](M)$ then $\models v_{i-1}(N_i)$.

Now if there is no sequence of satisfying Skolem functions for $Q_{i+1} v_{i-1}[x_i \leftarrow \bot](M)$ then, by Lemma 5 and definition of the algorithm *search_comp_qbf*, $v_{i-1}(N_i) = \bot$. □

Theorem 9 *Let QM be a valid QBF. Then search_comp_qbf$(Q, M)^*$ is minimal.*

Proof of Theorem **??**.

- *Base cases:*
  - We suppose that $q = \forall$.
    - If $M \equiv \top$ then *search_comp_qbf*$(\forall x, M) = (\forall x(\top, \top))$ then

      $$search\_comp\_qbf(\forall x, M)^* = \forall x((\neg x \vee \top) \wedge (x \vee \top)) \cong \forall x \top$$

      is minimal.
    - If $M \equiv x$ ($M \equiv \neg x$ and $M \equiv \bot$ are similar) then $\forall x M$ is not valid.
  - We suppose that $q = \exists$.
    - If $M \equiv \top$ then

      $$search\_comp\_qbf(\exists x, M) = (\exists x(\top, \top))$$

      then

      $$search\_comp\_qbf(\exists x, M)^* = \exists x((\neg x \vee \top) \wedge (x \vee \top)) \cong \exists x \top$$

      is minimal.

* If $M \equiv x$ then

$$search\_comp\_qbf(\exists x, M) = (\exists x(\top, \bot))$$

then

$$search\_comp\_qbf(\exists x, M)^* = \exists x((\neg x \vee \top) \wedge (x \vee \bot)) \cong \exists x x$$

is minimal.

* If $M \equiv \neg x$ then

$$search\_comp\_qbf(\exists x, M) = (\exists x(\bot, \top))$$

then

$$search\_comp\_qbf(\exists x, M)^* = \exists x((\neg x \vee \bot) \wedge (x \vee \top)) \cong \exists x \neg x$$

is minimal.

* If $M \equiv \bot$ then $\exists x M$ is not valid.

– *Induction cases:* Let $Q = qxQ'$ and

$$bl^+ = search\_comp\_qbf(Q', [x \leftarrow \top](M))$$

and

$$bl^- = search\_comp\_qbf(Q', [x \leftarrow \bot](M))$$

- We suppose that ($bl^+ = non\_valid$) and ($bl^- = non\_valid$). Then by induction hypothesis $Q'[x \leftarrow \top](M)$ and $Q'[x \leftarrow \bot](M)$ are not valid then QM is not valid.
- We suppose that ($bl^+ \neq non\_valid$) and ($bl^- = non\_valid$). By induction hypothesis,

$$(bl^+)^* \text{ is minimal.} \tag{1}$$

If $q = \forall$ it is similar to the case "($bl^+ = non\_valid$) and ($bl^- = non\_valid$)". Otherwise $q = \exists$ and by Theorem 7

$$Q'[x \leftarrow \bot](M) \text{ is not valid.} \tag{2}$$

Now let

$$\begin{aligned} & QM_+ \\ & = (Q, (\top, \bot)\ ;\ grds(bl^+))^* \\ & = search\_comp\_qbf(Q, M)^* \end{aligned} \tag{3}$$

and

$$Q'M^+ = (bl^+)^* \tag{4}$$

Since by Definition 2

$$M_+ = (\neg x \vee \top) \wedge (x \vee \bot) \wedge M^+$$

then

$$[x \leftarrow \top](M_+) \equiv M^+ \text{ and } [x \leftarrow \bot](M_+) \equiv \bot$$

then

$$Q'[x \leftarrow \top](M_+) \text{ is minimal if and only if } Q'M^+ \text{ is also minimal} \quad (5)$$

and

$$Q'[x \leftarrow \bot](M_+) \text{ is not valid.} \quad (6)$$

then from (5), (4) and (1) $Q'[x \leftarrow \top](M_+)$ is minimal then with (6) and (2) QM_+ is minimal then with (3) $search\_comp\_qbf(Q, M)^*$ is minimal.

- We suppose that $(bl^+ = non\_valid)$ and $(bl^- \neq non\_valid)$.
  The case is similar to the previous case.
- We suppose that $(bl^+ \neq non\_valid)$ and $(bl^- \neq non\_valid)$. By induction hypothesis,

$$(bl^+)^* \text{ is minimal} \quad (7)$$

and

$$(bl^-)^* \text{ is minimal.} \quad (8)$$

Let

$$QM_+ = (Q, (\top, \bot)\ ;\ grds(bl^+))^* \quad (9)$$

$$QM_- = (Q, (\bot, \top)\ ;\ grds(bl^-))^* \quad (10)$$

and

$$\begin{aligned} &QM_\oplus \\ &= ((Q, (\top, \bot)\ ;\ grds(bl^+)) \oplus (Q, (\bot, \top)\ ;\ grds(bl^-)))^* \\ &= search\_comp\_qbf(Q, M)^* \end{aligned} \quad (11)$$

By Theorem 6,

$$M_\oplus \equiv (M_+ \vee M_-) \quad (12)$$

Let

$$Q'M^+ = (bl^+)^* \quad (13)$$

and

$$Q'M^- = (bl^-)^* \quad (14)$$

By Definition 2, (9) and (13)

$$M_+ = (\neg x \vee \top) \wedge (x \vee \bot) \wedge M^+$$

then

$$[x \leftarrow \top](M_+) \equiv M^+ \quad (15)$$

and

$$[x \leftarrow \bot](M_+) \equiv \bot \quad (16)$$

By Definition 2, (10) and (14)

$$M_- = (\neg x \vee \bot) \wedge (x \vee \top) \wedge M^-$$

then

$$[x \leftarrow \top](M_-) \equiv \bot \quad (17)$$

and

$$[x \leftarrow \bot](M_{-}) \equiv M^{-} \tag{18}$$

Then from (12), (15) and (17)

$$[x \leftarrow \top](M_{\oplus}) \equiv M^{+} \tag{19}$$

and from (12), (16) and (18)

$$[x \leftarrow \bot](M_{\oplus}) \equiv M^{-} \tag{20}$$

Then from (19)

$$Q'[x \leftarrow \top](M_{\oplus}) \text{ is minimal if and only if } Q'M^{+} \text{ is minimal} \tag{21}$$

and from (20)

$$Q'[x \leftarrow \bot](M_{\oplus}) \text{ is minimal if and only if } Q'M^{-} \text{ is minimal.} \tag{22}$$

Then from (21), (13) and (7) $Q'[x \leftarrow \top](M_{\oplus})$ is minimal and from (22), (14) and (8) $Q'[x \leftarrow \bot](M_{\oplus})$ is also minimal then $QM_{\oplus}$ is minimal then from (11) $search\_comp\_qbf(Q, M)^{*} \cong QM$ is minimal.

□