Nested Boolean Functions as Models for Quantified Boolean Formulas

Uwe Bubeck and Hans Kleine Büning

Computer Science Institute, University of Paderborn, Germany
bubeck@upb.de (U. Bubeck), kbcsl@upb.de (H. Kleine Büning)

Abstract. Nested Boolean functions or Boolean programs are an alternative to the quantified Boolean formula (QBF) characterization of polynomial space. The idea is to start with a set of Boolean functions given as propositional formulas and to define new functions as compositions or instantiations of previously defined ones. We investigate the relationship between function instantiation and quantification and present a compact representation of models and countermodels of QBFs with and without free variables as nested Boolean functions. The representation is symmetric with respect to Skolem models and Herbrand countermodels. For a formula with free variables, it can describe both kinds of models simultaneously in one complete equivalence model which can be Skolem or Herbrand depending on actual assignments to the free variables.

1 Introduction

The satisfiability problem for quantified Boolean formulas (QBF) is the canonical PSPACE-complete problem. As an extension of propositional logic, QBF draws its expressive power from the ability to have quantifiers over propositional variables, where universal quantification $\forall x\, \Phi(x)$ for a variable x and a propositional or quantified Boolean formula Φ is defined to be true if and only if $\Phi(0)$ is true *and* $\Phi(1)$ is true, and $\exists y\, \Phi(y)$ means that $\Phi(0)$ *or* $\Phi(1)$ is true.

Nested Boolean functions (NBF) or Boolean programs have been introduced by Cook and Soltys [6] as an alternative characterization of polynomial space. They extend propositional logic with the ability to define Boolean functions as compositions or instantiations of previously defined functions, starting with a set of initial functions given as propositional formulas. For example, let

$$f_0(p_1, p_2) := (\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)$$

be an initial function which computes the parity of two binary variables. Then the parity of four variables can be computed by reusing f_0:

$$f_1(p_1, p_2, p_3, p_4) := f_0(f_0(p_1, p_2), f_0(p_3, p_4))$$

The parity of 16 variables can be expressed compactly by reusing f_1, and so on [6]. By replacing all occurrences of f_0 with its definition, we can expand f_1:

$$\begin{aligned} def(f_1)(p_1, ..., p_4) := {} & (\neg\, ((\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)) \wedge ((\neg p_3 \wedge p_4) \vee (p_3 \wedge \neg p_4))) \vee \\ & (((\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)) \wedge \neg\, ((\neg p_3 \wedge p_4) \vee (p_3 \wedge \neg p_4))) \end{aligned}$$

The semantics of nested Boolean functions can be defined by this fallback to propositional formulas. But in order to evaluate a NBF sequence $(f_0, ..., f_k)$, i.e. to check whether $f_k(a_1, ..., a_n) = 1$ for given arguments $a_1, ..., a_n \in \{0, 1\}$, it is not necessary to actually create the propositional expansion $def(f_k)$. By immediately replacing subterms whenever their values are known, the formula can be simplified on-the-fly, and polynomial space is sufficient. It can be shown that the evaluation and the satisfiability problem for NBF are PSPACE-complete [6]. Not surprisingly, there exist efficient transformations between QBF and NBF: any NBF can be transformed in linear time into an equivalent QBF [4], while the best known transformation in the other direction needs quadratic time [4].

In this paper, we want to further clarify the relationship between function instantiation and quantification. It is well known that in a closed prenex QBF, e.g. $\Phi = \forall x_n \exists y_n ... \forall x_1 \exists y_1 \ \phi(x_1, ..., x_n, y_1, ..., y_n)$, every existentially quantified variable y_i can be associated with a Boolean function f_i, called *Skolem function* [13], which depends on the values of those universally quantified variables whose quantifiers are outer to $\exists y_i$. For the given Φ, these are $x_i, ..., x_n$. Then Φ is true if and only if there exist $f_1, ..., f_n$ so that $\forall x_n ... \forall x_1 \ \phi(x_1, ..., x_n, f_1(x_1, ..., x_n), ..., f_n(x_n))$ is true. That means each occurrence of an existential variable is replaced with the corresponding Skolem function, and the resulting matrix must be true for all values of the universal variables. If that is the case, we call $f_1, ..., f_n$ a *Skolem model*. Analogously, the universally quantified variables can be mapped to *Herbrand functions* [8]. Φ is false if and only if there is a *Herbrand countermodel* $g_1, ..., g_n$ so that $\exists y_n ... \exists y_1 \ \phi(g_1(y_2, ..., y_n), ..., g_{n-1}(y_n), g_n(), y_1, .., y_n)$ is false. Recently, Balabanov and Jiang have shown how to extract a Skolem model of a true closed QBF from its cube-resolution proof and a Herbrand countermodel from the clause-resolution proof of a false QBF [1]. Earlier approaches use explicit skolemization techniques [2, 3, 9] and do not directly address Herbrand countermodels, or the generated strategies are not explicitly represented as functions [7]. Both Skolem and Herbrand (counter)models are of great practical importance when applications require solutions or explanations of unsatisfiability in addition to a mere decision of satisfiability. But the compact representation of these (counter)models remains a great problem for practical applications [12].

We consider Skolem and Herbrand (counter)models from a more theoretical viewpoint and show that we can compactly encode them by polynomial-size NBFs. More importantly, we further study the duality between Skolem models and Herbrand countermodels by considering QBFs with free variables. While a closed QBF is either true or false, the valuation of a QBF $\Phi(\mathbf{z})$ with free variables $\mathbf{z} = z_1, ..., z_r$ depends on the values of the free variables [10]. Consider the example $\Phi(z) = \forall x \exists y \ (x \vee y) \wedge (\neg x \vee \neg y) \wedge (\neg z \vee \neg y)$. If z is 0, we have $\Phi(0) = \forall x \exists y \ (x \vee y) \wedge (\neg x \vee \neg y) \wedge (\neg 0 \vee \neg y)$, which is a closed QBF and is true with Skolem model $f_y(x) = \neg x$. If z is 1, $\Phi(1) = \forall x \exists y \ (x \vee y) \wedge (\neg x \vee \neg y) \wedge (\neg 1 \vee \neg y)$ is false with Herbrand countermodel $f_x() = 0$. The interesting observation here is that QBFs with free variables can have Skolem models and Herbrand countermodels for different values of the free variables. How are both kinds related when the formula remains the same and only the free variables change? We propose a unified

complete equivalence model, in which all variables are mapped to functions and show how Skolem and Herbrand (counter)models are embedded in it and how these functions can be computed from the formula matrix by NBFs.

For space considerations, we rely only on the above informal introduction of NBF syntax and semantics and refer the reader to [4] for formal definitions. We furthermore assume in the following that all QBFs are in prenex normal form, i.e. all quantifiers appear in front of a purely propositional matrix. Two QBFs $\Phi(z_1, ..., z_r)$ and $\Psi(z_1, ..., z_r)$ with free variables $z_1, ..., z_r$ are *logically equivalent*, written $\Phi(z_1, ..., z_r) \approx \Psi(z_1, ..., z_r)$ or simply $\Phi \approx \Psi$, if and only if for every truth assignment τ to the free variables $z_1, ..., z_r$ both formulas evaluate to the same truth value [10]. We consider propositional formulas as QBFs in which all variables are free. Furthermore, given a NBF $(f_0, ..., f_k)$ where f_k has arguments $a_1, ..., a_n$, we say $f_k(a_1, ..., a_n)$ is logically equivalent to a QBF $\Phi(a_1, ..., a_n)$, written $f_k(a_1, ..., a_n) \approx \Phi(a_1, ..., a_n)$, if and only if the propositional expansion $def(f_k)(a_1, ..., a_n)$ is equivalent to $\Phi(a_1, ..., a_n)$. When we evaluate a NBF, we use "=" instead of "≈" for a more lightweight notation. The length of a QBF is the number of variable occurrences, including the prefix. The length of a NBF $(f_0, ..., f_k)$ is $|f_0| + ... + |f_k|$, where $|f_i|$ is the total number of occurrences of constants, variables and function symbols on the right-hand side of the defining equation of f_i . The parity example above has length $4 + 7$.

2 Quantification as Iterated Function Composition

Definition 1. *Let $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$ with $Q_i \in \{\forall, \exists\}$ be a QBF with bound variables $v_1, ..., v_n$ ($n \geq 1$ w.l.o.g), free variables $\mathbf{z} = z_1, ..., z_r$ and propositional matrix ϕ.*
Then we iteratively define Boolean functions $F_0, ..., F_n$ as follows:

1. $F_0(v_1, ..., v_n, \mathbf{z}) := \phi(v_1, ..., v_n, \mathbf{z})$
2. *For $i = 1, ..., n$:*

$$F_i(v_{i+1}, ..., v_n, \mathbf{z}) := \begin{cases} F_{i-1}(F_{i-1}(0, v_{i+1}, ..., v_n, \mathbf{z}), v_{i+1}, ..., v_n, \mathbf{z}) & , \text{if } Q_i = \forall \\ F_{i-1}(F_{i-1}(1, v_{i+1}, ..., v_n, \mathbf{z}), v_{i+1}, ..., v_n, \mathbf{z}) & , \text{if } Q_i = \exists \end{cases}$$

For the example $\Phi(z) = \forall v_2 \exists v_1 \ (v_1 \vee v_2) \wedge (\neg v_1 \vee \neg v_2) \wedge (\neg z \vee \neg v_1)$ we obtain:

$$\begin{aligned} F_0(v_1, v_2, z) &:= (v_1 \vee v_2) \wedge (\neg v_1 \vee \neg v_2) \wedge (\neg z \vee \neg v_1) \\ F_1(v_2, z) &:= F_0(F_0(1, v_2, z), v_2, z) = F_0((\neg v_2 \wedge \neg z), v_2, z) = \neg z \vee v_2 \\ F_2(z) &:= F_1(F_1(0, z), z) = F_1(\neg z, z) = \neg z \end{aligned}$$

The main idea behind this definition is that we try to assign values to the quantified variables, going from the outermost to the innermost quantifier (NBFs are evaluated by recursion from the last function in the sequence back to the initial functions). Similar to the DPLL algorithm for QBF (QDPLL) [5], we might have to branch in the worst case for $x = 0$ and $x = 1$ on each variable x. If x is universally quantified and the formula is false for $x = 0$, there is no need to try $x = 1$. Similarly, if x is existential and the formula is true for $x = 1$, we do not

try $x = 0$. In our NBF encoding, the result of the first branch determines where the second branch is going. If the formula is false for $x = 0$, we stay with $x = 0$, and if it is true for $x = 1$, we stay with $x = 1$. In either case, the arguments to the inner call of F_{i-1} are the same as for the outer call of F_{i-1}, which suggests that it would be an important optimization for a real NBF solver to recognize duplicate instantiations for the same arguments.

Lemma 1. *For $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$ with associated functions $F_0, ..., F_n$ as in Definition 1, it holds for all $i = 1, ..., n$ that*

$$F_i(v_{i+1}, ..., v_n, \mathbf{z}) \approx Q_i v_i ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z}) \ .$$

Proof. Let $i = 1$, and consider the case that $Q_1 = \forall$. Then we must show that $\phi(\phi(0, v_2, ..., v_n, \mathbf{z}), v_2, ..., v_n, \mathbf{z}) \approx \forall v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$. Assume that the left-hand side is true for some truth value assignment τ to $v_2, ..., v_n$ and $\mathbf{z}$, that means $\phi(\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})), \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$. Then it is not possible that the inner instantiation of ϕ is false. Because if we did substitute 0 for the inner instantiation $\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z}))$, the outer instantiation of ϕ would become the same and would thus also be false, which would contradict our assumption that the whole left-hand side is true. If, on the other hand, $\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ for the inner instantiation, the outer becomes $\phi(1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z}))$. With this being true by the initial assumption, we know that $\phi(v_1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z}))$ is true for $v_1 = 0$ and $v_1 = 1$, and thus also $\forall v_1 \ \phi(v_1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$.
From right to left, $\forall v_1 \ \phi(v_1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ for some τ implies that $\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ for the inner instantiation of ϕ on the left-hand side, so the outer instantiation on the left becomes $\phi(1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z}))$, which is also true by the universal quantification on v_1.
If $Q_1 = \exists$, we must show $\phi(\phi(1, v_2, ..., v_n, \mathbf{z}), v_2, ..., v_n, \mathbf{z}) \approx \exists v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$. For a truth assignment τ which satisfies the left-hand side, the inner instantiation of ϕ on the left is either false or true. Accordingly, $\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ or $\phi(1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ on the left, and that implies the right-hand side. In the other direction, let $\exists v_1 \ \phi(v_1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ for some τ. Then $\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$ or $\phi(1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 1$. If the latter holds, the inner instantiation on the left-hand side is also true, and the left-hand side becomes $\phi(1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z}))$ and is thus true. On the other hand, if $\phi(1, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z})) = 0$, the left-hand side becomes $\phi(0, \tau(v_2), ..., \tau(v_n), \tau(\mathbf{z}))$, which is true in this case.

For the induction step, assume $F_i(v_{i+1}, ..., v_n, \mathbf{z}) \approx Q_i v_i ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$ for $i \geq 1$. Then $Q_{i+1} v_{i+1} ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z}) \approx Q_{i+1} v_{i+1} F_i(v_{i+1}, ..., v_n, \mathbf{z})$, and we must show $F_i(F_i(0, v_{i+2}, ..., v_n, \mathbf{z}), v_{i+2}, ..., v_n, \mathbf{z}) \approx \forall v_{i+1} F_i(v_{i+1}, ..., v_n, \mathbf{z})$ resp. $F_i(F_i(1, v_{i+2}, ..., v_n, \mathbf{z}), v_{i+2}, ..., v_n, \mathbf{z}) \approx \exists v_{i+1} F_i(v_{i+1}, ..., v_n, \mathbf{z})$. When we let $\phi'(v_{i+1}, ..., v_n, \mathbf{z}) := F_i(v_{i+1}, ..., v_n, \mathbf{z})$, the proof can be obtained in complete analogy to the induction base when substituting ϕ' for ϕ. □

Corollary 1. *$F_n(\mathbf{z}) \approx Q_n v_n ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$*
for $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$ with $F_0, ..., F_n$ as in Definition 1.

The length of the NBF $(F_0, ..., F_n)$ is $|\phi|+\Sigma_{i=1...n}(2i+1+2|\phi|)$, which is quadratic in $|\Phi|$. This is the same as for the existing transformation from QBF to NBF in [4] which simulates quantifier expansion, while we now simulate QDPLL. An important difference is that the expansion approach leads to definitions of the form $f_{i+1}(...) := f_i(f_{j_1}(...), ..., f_{j_r}(...))$ and needs multiple initial functions, while we now have definitions $f_{i+1}(...) := f_i(f_i(...), x_2, ..., x_r)$ where recursion occurs only in the first argument of the outer f_i, and we now only need one initial function, which is exactly the matrix of the QBF. This is important for our equivalence models. The following small technical lemma will be helpful later.

Lemma 2. *For $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z})$ with $F_0, ..., F_n$ as in Def. 1, we have $F_i(v_{i+1}, ..., v_n, \mathbf{z}) \approx Q_i v_i F_{i-1}(v_i, ..., v_n, \mathbf{z})$ for all $i = 1, ..., n$.*

Proof. If $i = 1$, $F_0(v_1, ..., v_n, \mathbf{z})$ is $\phi(v_1, ..., v_n, \mathbf{z})$, and the claim follows immediately from Lemma 1.
For $i > 1$, $Q_i v_i(F_{i-1}(v_i, ..., v_n, \mathbf{z})) \approx Q_i v_i(Q_{i-1} v_{i-1} ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z}))$ by Lemma 1. Again by Lemma 1, the latter is equivalent to $F_i(v_{i+1}, ..., v_n, \mathbf{z})$. □

3 Equivalence Models for Quantified Boolean Formulas

Equivalence models for QBFs $\Phi(\mathbf{z}) = \forall x_n \exists y_n ... \forall x_1 \exists y_1\ \phi(x_1, ..., x_n, y_1, ..., y_n, \mathbf{z})$ with free variables $\mathbf{z}$ have been defined in [11] by an equivalence-preserving mapping of existential variables $y_1, ..., y_n$ to functions $h_1(x_1, ..., x_n, \mathbf{z}), ..., h_n(x_n, \mathbf{z})$ with $\Phi(\mathbf{z}) \approx \forall x_n ... \forall x_1\ \phi(x_1, ..., x_n, h_1(x_1, ..., x_n, \mathbf{z}), ..., h_n(x_n, \mathbf{z}), \mathbf{z})$. For a symmetric treatment of the quantifiers, we will now define the notion of *complete* equivalence models where *all* quantified variables, including the universal ones, are mapped to functions over the free variables.

Definition 2. *Let $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z})$ with $Q_i \in \{\forall, \exists\}$ be a QBF with bound variables $v_1, ..., v_n$, free variables $\mathbf{z} = z_1, ..., z_r$ and propositional matrix ϕ. A sequence of Boolean functions $h_1(\mathbf{z}), ..., h_n(\mathbf{z})$ is called a* complete equivalence model *if and only if $\Phi(\mathbf{z}) \approx \phi(h_1(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$.*

It is easy to see that every QBF has a complete equivalence model: For formulas without free variables, the complete equivalence model consists of constants $h_1, ..., h_n \in \{0, 1\}$ such that Φ is true if and only if $\phi(h_1, ..., h_n)$ is true. Clearly, every true (false, respectively) closed QBF has a satisfying (falsifying, respectively) truth assignment to the matrix, and has thus a complete equivalence model. For a QBF with free variables, a complete equivalence model could always be constructed in a naive way by considering all assignments $\tau(\mathbf{z})$ to the free variables and choosing $h_i(\tau(\mathbf{z})) := \epsilon_i \in \{0, 1\}$ for all $i = 1, ..., n$ such that $\phi(\epsilon_1, ..., \epsilon_n, \tau(\mathbf{z}))$ is true if and only if $\Phi(\tau(\mathbf{z}))$ is true.

The problem of deciding whether a sequence $h_1, ..., h_n \in \{0, 1\}$ is a complete equivalence model for a closed or open QBF is PSPACE-complete. It is in PSPACE, since the equivalence problem for QBF is in PSPACE, and the hardness follows from a similar reduction as in [11]: given a closed QBF $\Phi = Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n)$, let $\Phi' = \exists v_{n+1} Q_n v_n ... Q_1 v_1\ (\phi(v_1, ..., v_n) \wedge v_{n+1})$. Then $h_1 = ... = h_{n+1} = 0$ is a complete equivalence model for Φ' iff Φ is false.

Lemma 3. *If $\Sigma_2^P \neq \Pi_2^P$ in the polynomial-time hierarchy, there must exist quantified Boolean formulas with free variables for which every propositional representation of the complete equivalence model requires super-polynomial length.*

Proof. Consider a closed QBF $\Phi = \forall x_n ... \forall x_1 \exists y_m ... \exists y_1 \ \phi(x_1, ..., x_n, y_1, ..., y_m)$ for which the satisfiability problem is Π_2^P-complete. Then let $\Phi'(x_1, ..., x_n) := \exists y_m ... \exists y_1 \ \phi(x_1, ..., x_n, y_1, ..., y_m)$. If Φ' had a complete equivalence model with polynomial-size propositional encoding, we could guess in polynomial time propositional formulas $h_1(x_1, ..., x_n), ..., h_m(x_1, ..., x_n)$ and insert them into Φ. If a Π_1^P-oracle accepts $\forall x_n ... \forall x_1 \ \phi(x_1, ..., x_n, h_1(x_1, ..., x_n), ..., h_m(x_1, ..., x_n))$, we know that Φ is true and that $h_1, ..., h_m$ have been guessed correctly. In total, we would be able to solve the formula in Σ_2^P, and thus $\Sigma_2^P = \Pi_2^P$. □

Lemma 3 holds analogously for the non-complete equivalence models from [11] and for Skolem/Herbrand (counter)models, even for closed QBFs with only two levels of quantification. Also in practical QBF applications, Skolem/Herbrand (counter)models are often infeasibly large when represented as propositional formulas. We are now going to represent complete equivalence models as NBFs instead, allowing us to place a polynomial bound on the size of these models.

Definition 3. *Let $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1 \ \phi(v_1, ..., v_n, \mathbf{z})$ with $Q_i \in \{\forall, \exists\}$ be a QBF with bound variables $v_1, ..., v_n$ ($n \geq 1$ w.l.o.g), free variables $\mathbf{z} = z_1, ..., z_r$ and matrix ϕ. Using the function representation from Definition 1, we map each variable v_k to a model function h_k as follows:*

1. $h_n(\mathbf{z}) := \begin{cases} F_{n-1}(0, \mathbf{z}) & \text{, if } Q_n = \forall \\ F_{n-1}(1, \mathbf{z}) & \text{, if } Q_n = \exists \end{cases}$
2. *For $i = n-1, ..., 1$:*
   $h_i(\mathbf{z}) := \begin{cases} F_{i-1}(0, h_{i+1}(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z}) & \text{, if } Q_i = \forall \\ F_{i-1}(1, h_{i+1}(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z}) & \text{, if } Q_i = \exists \end{cases}$

The intuition here is as follows: according to Definition 1, $F_1, ..., F_n$ are defined by $F_i(v_{i+1}, ..., v_n, \mathbf{z}) := F_{i-1}(F_{i-1}(\sigma_i, v_{i+1}, ..., v_n, \mathbf{z}), v_{i+1}, ..., v_n, \mathbf{z})$ with $\sigma_i \in \{0, 1\}$ according to the quantifier type of Q_i. If we had already found model functions $h_{i+1}, ..., h_n$ (we omit their arguments $\mathbf{z}$ for simplicity), we could substitute them for $v_{i+1}, ..., v_n$: $F_i(h_{i+1}, ..., h_n, \mathbf{z}) \approx F_{i-1}(F_{i-1}(\sigma_i, h_{i+1}, ..., h_n, \mathbf{z}), h_{i+1}, ..., h_n, \mathbf{z})$. To write the latter as $F_{i-1}(h_i, ..., h_n, \mathbf{z})$, we choose $h_i := F_{i-1}(\sigma_i, h_{i+1}, ..., h_n, \mathbf{z})$.

Consider again the example $\Phi(z) = \forall v_2 \exists v_1 \ (v_1 \vee v_2) \wedge (\neg v_1 \vee \neg v_2) \wedge (\neg z \vee \neg v_1)$ with $F_1(v_2, z) = \neg z \vee v_2$ and $F_2(z) = \neg z$ (Section 2, p. 3). Then $h_2(z) = F_1(0, z) = \neg z$ and $h_1(z) = F_0(1, h_2(z), z) = \neg h_2(z) \wedge \neg z = z \wedge \neg z = 0$.

Lemma 4. *For $i = 1, ..., n$:*

$$F_n(\mathbf{z}) \approx F_{i-1}(h_i(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$$

Proof. The proof is by backward induction on i. For $i = n$, the right-hand side is $F_{n-1}(h_n(\mathbf{z}), \mathbf{z})$, which is $F_{n-1}(F_{n-1}(0, \mathbf{z}), \mathbf{z})$ if $Q_n = \forall$ and $F_{n-1}(F_{n-1}(1, \mathbf{z}), \mathbf{z})$ otherwise. By Definition 1 (from right to left), this is $F_n(\mathbf{z})$.

For the induction step, let $i \in \{1, ..., n-1\}$. Assume the above equivalence holds for $i+1$, that means $F_n(\mathbf{z}) \approx F_i(h_{i+1}(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$. By Definition 1, the right-hand side is $F_{i-1}(F_{i-1}(\sigma_i, h_{i+1}(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z}), h_{i+1}(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$ with $\sigma_i \in \{0, 1\}$ according to the quantifier type of Q_i. By the above Definition 3, $h_i(\mathbf{z}) := F_{i-1}(\sigma_i, h_{i+1}(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$, so the last expression can be written as $F_{i-1}(h_i(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$, i.e. the right-hand side of the statement to be proven. □

Theorem 1. *For $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z})$, model functions $h_1, .., h_n$ constructed according to Definition 3 are a complete equivalence model.*

Proof. We have to show that $\phi(h_1(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z}) \approx Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z})$. With $F_0 := \phi$, $\phi(h_1(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$ is $F_0(h_1(\mathbf{z}), ..., h_n(\mathbf{z}), \mathbf{z})$. Using Lemma 4 (for $i = 1$), the latter is equivalent to $F_n(\mathbf{z})$, which in turn is equivalent to $Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z})$ by Corollary 1. □

The size of the complete equivalence model constructed according to Definition 3 is $|F_0| + ... + |F_{n-1}| + |h_1| + ... + |h_n|$. The first part is quadratic in $|\Phi|$ as observed in Section 2, and $|h_1| + ... + |h_n| \leq \Sigma_{i=1...n}(i(1 + |\phi|) + 1 + |\phi|)$ (notice that $\mathbf{z}$ stands for at most $|\phi|$ free variable symbols), so the whole model has cubic size when it is written as a sequence of nested Boolean functions.

In [4], a linear-time transformation from NBF to QBF is presented. By applying this transformation to the above NBF representation of the complete equivalence model, we obtain the following corollary:

Corollary 2. *Every QBF $\Phi(\mathbf{z}) = Q_n v_n ... Q_1 v_1\ \phi(v_1, ..., v_n, \mathbf{z})$ with $Q_i \in \{\forall, \exists\}$ has a complete equivalence model $h_1(\mathbf{z}), ..., h_n(\mathbf{z})$ where each $h_i(\mathbf{z})$ can again be represented as a QBF with free variables $\mathbf{z}$ of size cubic in $|\Phi|$.*

By Lemma 3, if $h_1(\mathbf{z}), ..., h_n(\mathbf{z})$ are represented as propositional formulas, their size cannot be bounded by a polynomial in $|\Phi|$ if $\Sigma_2^P \neq \Pi_2^P$. Since we do have short QBF representations, a further consequence is that there must exist QBFs $\Psi(\mathbf{z})$ for which there are no logically equivalent propositional formulas $\psi(\mathbf{z})$ of length polynomial in $|\Psi(\mathbf{z})|$, unless $\Sigma_2^P = \Pi_2^P$.

Consider again the example $\Phi(z) = \forall v_2 \exists v_1\ (v_1 \vee v_2) \wedge (\neg v_1 \vee \neg v_2) \wedge (\neg z \vee \neg v_1)$ with $F_1(v_2, z) = \neg z \vee v_2$, $F_2(z) = \neg z$, $h_2(z) = \neg z$ and $h_1(z) = F_0(1, h_2(z), z) = \neg h_2(z) \wedge \neg z = 0$. If $z = 0$, $\Phi(0)$ is true and has a Skolem model. How is this Skolem model embedded in the complete equivalence model (h_1, h_2)? Notice that $h_1(0) = \neg h_2(0)$. Assume we had not mapped the universal variable v_2 (whose quantifier is outer to that of v_1) to h_2 and instead kept v_2 as a parameter to h_1. Then we would have $h_1(v_2, 0) = \neg v_2$. Indeed, the Skolem model function for v_1 is $f_1(v_2) = \neg v_2$. This suggests that we obtain a Skolem model if we apply the mapping from variables v_i to functions h_i only to existential variables and leave the universals as parameters to the functions of existential variables that are quantified further inside. So we modify Definition 3 to leave universal variables untouched. W.l.o.g., we consider only QBFs with alternating quantifiers.

Definition 4. *Let $\Phi(\mathbf{z}) = \exists v_n \forall v_{n-1} ... \exists v_2 \forall v_1\ \phi(v_1, ..., v_n, \mathbf{z})$ with even $n \geq 2$ be a QBF with alternating quantifiers and free variables $\mathbf{z}$. Using the function*

representation from Definition 1, we map each existentially quantified variable $v_n, v_{n-2}, ..., v_2$ *to a Skolem function as follows:*

1. $f_n(\mathbf{z}) := F_{n-1}(1, \mathbf{z})$
2. *For* $i = n-2, ..., 2$ *(if* $n > 2$*):*
   $f_i(v_{i+1}, ..., v_{n-1}, \mathbf{z}) := F_{i-1}(1, v_{i+1}, f_{i+2}(v_{i+3}, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z})$

Lemma 5. *For all even* i *with* $0 \leq i \leq n-2$*:*

$$F_n(\mathbf{z}) \approx \forall v_{n-1} ... \forall v_{i+1}\ F_i(v_{i+1}, f_{i+2}(v_{i+3}, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z})$$

Proof. The proof is by backward induction on i. For $i = n-2$, the right-hand side is $\forall v_{n-1}\ F_{n-2}(v_{n-1}, f_n(\mathbf{z}), \mathbf{z})$. By Lemma 2, this is equivalent to $F_{n-1}(f_n(\mathbf{z}), \mathbf{z}) := F_{n-1}(F_{n-1}(1, \mathbf{z}))$, and that is $F_n(\mathbf{z})$ by Definition 1.
For the induction step, let i be even with $i \in \{0, ..., n-4\}$ and assume the above equivalence holds for $i+2$. We must show:

$$F_n(\mathbf{z}) \approx \forall v_{n-1} ... \forall v_{i+1}\ F_i(v_{i+1}, f_{i+2}(v_{i+3}, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z})$$

By Lemma 2:

$$\begin{aligned} &\forall v_{i+1} F_i(v_{i+1}, f_{i+2}(v_{i+3}, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z}) \\ \approx\ & F_{i+1}(f_{i+2}(v_{i+3}, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z}) \end{aligned}$$

The first argument of F_{i+1} is

$$f_{i+2}(v_{i+3}, ..., v_{n-1}, \mathbf{z}) := F_{i+1}(1, v_{i+3}, f_{i+4}(v_{i+5}, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z})$$

by Definition 4. If we substitute this into the right-hand side of the previous equivalence, we obtain $F_{i+1}(F_{i+1}(1, v_{i+3}, ..., f_n(\mathbf{z}), \mathbf{z}), v_{i+3}, ..., f_n(\mathbf{z}), \mathbf{z})$, and that is $F_{i+2}(v_{i+3}, ..., f_n(\mathbf{z}), \mathbf{z})$ by Definition 1, because v_{i+2} is existentially quantified. By the induction hypothesis, $\forall v_{n-1} ... \forall v_{i+3}\ F_{i+2}(v_{i+3}, ..., f_n(\mathbf{z}), \mathbf{z}) \approx F_n(\mathbf{z})$. □

Corollary 3. *Let* $\Phi(\mathbf{z}) = \exists v_n \forall v_{n-1} ... \exists v_2 \forall v_1\ \phi(v_1, ..., v_n, \mathbf{z})$ *with even* $n \geq 2$ *be a QBF with alternating quantifiers and free variables* $\mathbf{z}$*. Then*

$$\Phi(\mathbf{z}) \approx \forall v_{n-1} ... \forall v_1\ \phi(v_1, f_2(v_3, ..., v_{n-1}, \mathbf{z}), ..., v_{n-1}, f_n(\mathbf{z}), \mathbf{z})$$

for functions $f_2, .., f_n$ *as in Definition 4. That means* $f_2, ..., f_n$ *are a non-complete equivalence model in the sense of* [11] *and a Skolem model if* Φ *is closed and true.*

Analogously, it is possible to show that Herbrand countermodels can be obtained when omitting the existential variables from the complete equivalence models.

4 Conclusion and Future Work

We have introduced complete equivalence models for QBFs as a generalization of Skolem and Herbrand (counter)models by mapping all quantified variables to Boolean functions, which we can compactly encode by NBFs. These NBFs are essentially recursive instantiations of the propositional matrix of the QBF, which raises the question for future work how restrictions on the matrix, e.g. 2-CNF or Horn, affect the structure of the complete equivalence models. It would also be interesting to investigate whether this recursive computation can be related to the resolution-based (counter)model construction in [1].

References

[1] V. Balabanov and J.-H. Jiang. Unified QBF Certification and its Applications. *Formal Methods in System Design*, 41:45–65, 2012.

[2] M. Benedetti. Evaluating QBFs via Symbolic Skolemization. In *Proc. 11th Intl. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2004)*, volume 3452 of *LNCS*, pages 285–300. Springer, 2005.

[3] M. Benedetti. Extracting Certificates from Quantified Boolean Formulas. In *Proc. 19th Intl. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 47–53. Morgan Kaufmann Publishers, 2005.

[4] U. Bubeck and H. Kleine Büning. Encoding Nested Boolean Functions as Quantified Boolean Formulas. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 8:101–116, 2012.

[5] 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.

[6] S. Cook and M. Soltys. Boolean Programs and Quantified Propositional Proof Systems. *The Bulletin of the Section of Logic*, 28(3):119–129, 1999.

[7] A. Goultiaeva, A. Van Gelder, and F. Bacchus. A Uniform Approach for Generating Proofs and Strategies for Both True and False QBF Formulas. In *Proc. 22th Intl. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, pages 546–553. AAAI Press, 2011.

[8] J. Herbrand. *Recherches sur la Théorie de la Demonstration.* PhD Thesis, Université de Paris, 1930.

[9] T. Jussila, A. Biere, C. Sinz, D. Kröning, and C. Wintersteiger. A First Step Towards a Unified Proof Checker for QBF. In *Proc. 10th Intl. Conf. on Theory and Applications of Satisfiability Testing (SAT 2007)*, volume 4501 of *LNCS*, pages 201–214. Springer, 2007.

[10] H. Kleine Büning and U. Bubeck. Theory of Quantified Boolean Formulas. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, pages 735–760. IOS Press, 2009.

[11] H. Kleine Büning and X. Zhao. Equivalence Models for Quantified Boolean Formulas. In *Proc. 7th Intl. Conf. on Theory and Applications of Satisfiability Testing (SAT 2004), Revised Selected Papers*, volume 3542 of *LNCS*, pages 224–234. Springer, 2005.

[12] A. Niemetz, M. Preiner, F. Lonsing, M. Seidl, and A. Biere. Resolution-Based Certificate Extraction for QBF (Tool Presentation). In *Proc. 15th Intl. Conf. on Theory and Applications of Satisfiability Testing (SAT 2012)*, volume 7317 of *LNCS*, pages 430–435. Springer, 2012.

[13] T. Skolem. Über die Mathematische Logik. *Norsk matematisk Tidsskrift*, 10:125–142, 1928.