A. FLÖGEL
H. KLEINE BÜNING
T. LETTMANN

On the restricted equivalence for subclasses of propositional logic

RAIRO. Informatique théorique et applications, tome 27, nº 4 (1993), p. 327-340
<http://www.numdam.org/item?id=ITA_1993__27_4_327_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 27, n° 4, 1993, p. 327 à 340)

ON THE RESTRICTED EQUIVALENCE FOR SUBCLASSES OF PROPOSITIONAL LOGIC (*)

by A. FLÖGEL (1), H. KLEINE BÜNING (1) and T. LETTMANN (1)

Communicated by P. V. EMDE BOAS

Abstract. – *In this paper we investigate the restricted equivalence problem and the restricted implication problem for classes of propositional formulas for which the satisfiability problem is solvable in polynomial time. The restricted equivalence problem of a class $\mathscr{C}$ can be expressed as equivalence problem of quantified Boolean formulas with matrix in $\mathscr{C}$. While for quantified definite* Horn *formulas the equivalence problem is shown to be* coNP-*complete, we give* DTIME(n^2) *algorithms for quantified* 2CNF *formulas.*

Résumé. – *Nous étudions le problème de l'équivalence restreinte et le problème de l'implication restreinte pour des classes des formules propositionnelles, dont le problème de la satisfaisabilité est décidable en temps polynômial. On peut exprimer le problème de l'équivalence restreinte pour une classe $\mathscr{C}$ par un problème d'équivalence pour les formules booléennes quantifiées dont la matrice est dans $\mathscr{C}$. Nous démontrons que le problème est* coNP-*complet pour des formules quantifiées dont la matrice est de* Horn *et nous présentons une solution en* DTIME(n^2) *pour les formules quantifiées dont la matrice est de* Krom *(conjonction de clauses de longueur* 2).

1. INTRODUCTION

In various contexts we make use of the expressive power of formal logic to give exact descriptions. But if several people work on one task, besides the question of consistency we also have to handle the problem whether two descriptions "mean" the same. This is due to the fact that everybody uses his own "private" abbreviations, that should be seen as "local" to his description. So it is useful to have algorithms to test, whether the knowledge bases are equivalent with respect to a given set of terms.

In first order logic we have two well-known notions of equivalence. *Logical equivalence* ($\approx$) means that formulas have identical sets of models, while

(*) Received February 1992, accepted September 1992.
(1) FB 17, Mathematik/Informatik, Universität Paderborn, Postfach 1621, D 4790 Paderborn (Germany).

0988-3754/93/04 327 14/\$3.40/© AFCET-Gauthier-Villars

equivalence with respect to satisfiability ($\approx_{\text{sat}}$) means, that one formula has a model, if and only if the other formula has one. Logical equivalence is very restrictive, but for many applications $\approx_{\text{sat}}$ is too weak, since for any two satisfiable formulas the relation $\approx_{\text{sat}}$ holds.

Often Tseitin's method [8] is used in order to transform a formula into conjunctive normal form (CNF). In that case for example each subformula $\alpha \vee (\beta \wedge \sigma)$ is replaced by $(x \vee \alpha) \wedge (\neg x \vee \beta) \wedge (\neg x \vee \sigma)$, where x is a new variable not occurring in α, β and σ. Obviously both formulas are equivalent with respect to satisfiability, but additionally both formulas have the same set of consequences containing variables of α, β and σ only. This also is the case by the transformation of CNF formulas into 3CNF, where each clause $(L_1 \vee \ldots \vee L_n)$ is replaced by $(L_1 \vee L_2 \vee x_1)$, $(\neg x_1 \vee L_3 \vee x_2)$, $\ldots$, $(\neg x_{n-3} \vee L_{n-1} \vee L_n)$ with new variables $x_1, \ldots, x_{n-3}$. Both formulas imply the same subset of the clauses, that contain only variables of the original formula.

We say two CNF-formulas are *restrictedly equivalent* with respect to a set of variables R, if each clause built with variables from R only, that is implied by one formula, is implied by the other formula as well.

A further example of restricted equivalence is the use of internal variables in order to reduce the length of formulas, *i.e.* $a_i \to b_j$ $(1 \leqq i, j \leqq n)$ is replaced by $a_i \to x$, $x \to b_j$. As in the example of different descriptions of some object we want to know whether both bases describe the same for a fixed set of "external" variables.

A slightly different problem is the drawing of all inferences, that contain only variables from R. This problem has been investigated as a Boolean projection problem in [4]. In [2] the concept of weak equivalence, a generalization of equivalence between constraint sets, is introduced, that allows to state the equivalence relative to a set of variables of the constraint sets.

It is a trivial observation that two formulas α and β are restrictedly equivalent with respect to a set of variables R, if and only if the quantified Boolean formulas $\exists x_1 \ldots \exists x_n \alpha$ and $\exists y_1 \ldots \exists y_m \beta$ are equivalent, with $\{x_1, \ldots, x_n\} = \operatorname{var}(\alpha) \setminus R$ and $\{y_1, \ldots, y_m\} = \operatorname{var}(\beta) \setminus R$. Thus the restricted equivalence problem is the equivalence problem for Boolean formulas with existential quantifiers.

In the next section, we give the definitions and an overview of known results used in this paper. Then we will investigate the restricted equivalence problem and the restricted implication problem for Horn formulas and the equivalence problem for quantified Horn formulas. We show that all of these problems are coNP-complete. For quantified 2CNF formulas we give algorithms to decide the equivalence problem and the implication problem in quadratic time, proving also that each Q2CNF formula can be tranferred in quadratic time to an equivalent Q2CNF formula without universal quantifiers.

2. PRELIMINARIES

We assume that he reader is familiar with the basic notations and results of propositional logic. Therefore we will start with the definition of restricted equivalence and restricted implication. Then we will focus on the definitions and results concerning quantified Boolean formulas.

DEFINITION ($\overset{R}{\models}$, $\approx_R$): *Let be* α *and* β *formulas and* R *a set of variables. Restricted implication of* β *by* α *with respect to* R, $\alpha \overset{R}{\models} \beta$, *holds, if and only if for each clause* π *built only with variables of* R *holds, that* $\alpha \models \pi$ *implies* $\beta \models \pi$. *Restricted equivalence of* α *and* β *with respect to* R, $\alpha \approx_R \beta$, *holds, if and only if* $\alpha \overset{R}{\models} \beta$ *and* $\beta \overset{R}{\models} \alpha$ *hold.*

In the following a quantified Boolean formula (QBF) Φ is of the form $Q_1 v_1 \ldots Q_n v_n \alpha$, where each $Q_i \in \{\forall, \exists\}$, $\{v_1, \ldots, v_n\}$ is a set of n different propositional variables and α is a propositional formula built with the logical constants 0 (for "false") and 1 (for "true"), a set of propositional variables $\mathrm{var}(\alpha)$ and the logical connectives $\wedge$, $\vee$, and $\neg$. α is called the *matrix* of Φ. We will use $(\alpha_1 \leftarrow \alpha_2)$ as a shorthand for $(\alpha_1 \vee \neg \alpha_2)$ and $\mathrm{lit}(V)$ as a shorthand for the set of literals over the set of variables V. We will use $Q_i v_i v_{i+1} \ldots v_j$ to denote the prefix $Q_i v_i Q_{i+1} v_{i+1} \ldots Q_j v_j$ with $Q_i = Q_{i+1} = \ldots = Q_j$. We say Φ is *closed* if and only if $\{v_1, \ldots, v_n\} \supseteqq \mathrm{var}(\alpha)$. A variable z is called *free*, if and only if

$$z \in \mathrm{freevar}(\alpha) := (\mathrm{var}(\alpha) \setminus \{v_1, \ldots, v_n\}).$$

Note that each propositional formula α can be seen as a quantified formula with $\mathrm{var}(\alpha) = \mathrm{freevar}(\alpha)$.

A *truth assignment* $\mathfrak{J}$: freevar $(\Phi) \to \{0, 1\}$ associates a truth value to each free variable. The extension of $\mathfrak{J}$ to a truth assignment for QBF formulas is given by the well-known definition for propositional formulas together with $\mathfrak{J}(\forall x \Phi) := \mathfrak{J}(\Phi[x/1] \wedge \Phi[x/0])$ and $\mathfrak{J}(\exists y \Phi) := \mathfrak{J}(\Phi[y/1] \vee \Phi[y/0])$, where $\Phi[v/c]$ denotes the formula obtained by replacing all occurrences of the variable v in Φ by constant c.

Often it is useful to have an algorithmic notion of how to determine a truth value of QBF formulas under an initial truth assignment $\mathfrak{J}$ on freevar(Φ). $\mathfrak{J}(\Phi)=1$ holds, if and only if for each extension of $\mathfrak{J}$ on the $\forall$-variables of Φ we can find truth values for the $\exists$-variables depending only on the truth values of the $\forall$-variables that occur earlier in the prefix of Φ. We say a variable v_i *is before* a variable v_j, if the occurrence of v_i precedes the occurrence of v_j in the prefix of Φ. If v_i is universally quantified, we also say, that v_i *governs* v_j. E.g. in the formula $\Phi=\forall x_1 \exists y \forall x_2 (x_1 \wedge \neg y \vee x_2 \rightarrow z)$ the variable x_1 is before y and x_2. Therefore x_1 governs y, that means $\mathfrak{J}(\Phi)=1$, if and only if under the initial setting of $\mathfrak{J}(z)$ for each selection of truth values for x_1 and x_2 there is a truth value for y depending only on x_1 such that $(x_1 \wedge \neg y \vee x_2 \rightarrow z)$ evaluates to "true".

With the above definition of truth assignment the terms *satisfiability*, *tautology*, *inconsistency*, *equivalence* and *implication*, known from propositional logic, can be canonically extended to quantified Boolean formulas with free variables. Note that the truth value $\mathfrak{J}(\Phi)$ of a closed formula Φ can be determined with an empty initial truth assignment, *i.e.* closed quantified Boolean formulas are either true or false. The *evaluation problem* EVAL$(\mathscr{C})$ for a class $\mathscr{C}$ of closed quantified Boolean formulas is to decide whether a formula of this class is true. So to decide, whether a quantified Boolean formula Φ with free variables $\{z_1, \ldots, z_n\}$ is satisfiable, is equivalent to the evaluation problem for $\exists z_1, \ldots, z_n \Phi$.

A *quantified* CNF *formula* (QCNF) is of the form

$$Q_1 v_1 \ldots Q_n v_n (\varphi_1 \wedge \ldots \wedge \varphi_p)$$

where φ_i are propositional clauses.

It is well-known that EVAL (QCNF) is PSPACE-complete [6, 7]. But there are subclasses of QCNF for which the evaluation problem is solvable in polynomial time.

DEFINITION (QHorn, Q2CNF): *A* QCNF *formula is a quantified Horn formula* (QHorn), *if each clause of the matrix is a* Horn *clause*, i.e. *each clause contains at most one positive literal. A* QCNF *formula is a* quantified 2CNF formula (Q2CNF), *if each clause contains at most two literals.*

In [5] it is shown that EVAL(QHorn) is decidable in quadratic time and in [1] a linear time algorithm for EVAL(Q2CNF) is given.

DEFINITION (EQUIV$(\mathscr{C})$, IMPL$(\mathscr{C})$): *Let* α, $\beta \in \mathscr{C}$ *with* $\mathscr{C}$ *some class of quantified Boolean formulas. The decision problem* EQUIV$(\mathscr{C})$ *is defined as*

$$\text{EQUIV}(\mathscr{C}) := \{ (\alpha, \beta) \in \mathscr{C}^2 \mid \mathfrak{J}(\alpha)=\mathfrak{J}(\beta) \text{ } \textit{for each } \mathfrak{J} \textit{ with domain } (\mathfrak{J}) \supseteq \text{freevar}(\alpha) \cup \text{freevar}(\beta) \}$$

The decision problem IMPL ($\mathscr{C}$) *is defined as*

$$\text{IMPL}(\mathscr{C}) := \{(\alpha, \beta) \in \mathscr{C}^2 \mid \mathfrak{J}(\alpha) = 1 \Rightarrow \mathfrak{J}(\beta) = 1$$
$$\textit{for each } \mathfrak{J} \textit{ with domain } (\mathfrak{J}) \supseteq \text{freevar}(\alpha) \cup \text{freevar}(\beta)\}$$

As mentioned earlier, the equivalence problem for quantified Boolean formulas includes the restricted equivalence problem for propositional formulas as a special case. Therefore a result concerning the complexity of restricted equivalence yields a lower bound for the complexity of the equivalence problem of quantified Boolean formulas of the appropriate class, while the complexity of EQUIV ($\mathscr{C}$) is an upper bound for the complexity of the corresponding restricted equivalence problem.

Since the complexity of EVAL ($\mathscr{C}$) also is a lower bound for the complexity of EQUIV ($\mathscr{C}$), we will restrict our investigations to classes for which the evaluation problem is decidable in polynomial time. So we are concerned with restricted equivalence and restricted implication for Horn and 2CNF as well as with the equivalence problem and implication problem for QHorn and Q2CNF.

3. RESTRICTED EQUIVALENCE AND IMPLICATION

In this section we investigate the restricted equivalence problem and the restricted implication problem for some subclasses of propositional logic for which the satisfiability problem is solvable in polynomial time. We make use of the connection between restricted equivalence in propositional calculus and equivalence of quantified Boolean formulas by using the more general case where appropriate.

3.1. Horn Formulas

A formula $\alpha \in$ Horn is a DHorn formula (definite Horn formula), if α does not contain negative clauses, *i.e.* clauses consisting of disjunctions of negative literals only. A formula $\alpha \in$ DHorn is *acyclic*, if and only if there is a topological ordering of the variables in α, such that for each clause the order of the positive literal is less than the order of all negative literals in this clause.

THEOREM 3.1: *The restricted implication problem* $\{(\alpha, \beta, R) \mid \alpha, \beta \in \text{DHorn},$ α, β *acyclic, R set of variables and* $\alpha \overset{R}{\models} \beta\}$ *and the restricted equivalence problem*

$\{(\alpha, \beta, R) \mid \alpha, \beta \in \text{DHorn}, \alpha, \beta$ *acyclic*, R *set of variables and* $\alpha \approx_R \beta\}$ *are* coNP-*complete*.

Proof: It is easy to see, that the restricted equivalence problem and the restricted implication problem for acyclic DHorn formulas are in coNP, by choosing some clause π with atoms in R and testing in polynomial time, whether π is the consequence of only one of the two formulas.

The completeness will be shown using the NP-completeness of the Monotone-3SAT-problem [3], that is $\{(\alpha \wedge \beta) \in 3\text{CNF} \mid \alpha$ contains only positive literals, β contains only negative literals and $(\alpha \wedge \beta) \in 3\text{SAT}\}$.

Let

$$\alpha = \{(L_{i,1} \vee L_{i,2} \vee L_{i,3}) \mid 1 \leqq i \leqq n\}$$

and

$$\beta = \{(\neg K_{j,1} \vee \neg K_{j,2} \vee \neg K_{j,3}) \mid 1 \leqq j \leqq m\}$$

with positive literals $L_{i,p}$ and $K_{j,q}$, $1 \leqq i \leqq n$, $1 \leqq j \leqq m$, $1 \leqq p, q \leqq 3$.

We associate the formula $(\alpha \wedge \beta)$ with two definite Horn formulas Φ_1 and Φ_2 and a set R, such that $(\alpha \wedge \beta) \notin \text{SAT}$, if and only if $\Phi_1 \approx_R \Phi_2$, and even such that also $(\alpha \wedge \beta) \notin \text{SAT}$, if and only if $\Phi_2 \overset{R}{\models} \Phi_1$.

Let be

$$\begin{aligned}
\Phi_1 &:= \{y_i \leftarrow L_{i,p} \mid 1 \leqq p \leqq 3,\ 1 \leqq i \leqq n\} \cup \{y \leftarrow y_1, \ldots, y_n\} \\
\Phi_2 &:= \{y_i \leftarrow L_{i,p} \mid 1 \leqq p \leqq 3,\ 1 \leqq i \leqq n\} \cup \{v \leftarrow y_1, \ldots, y_n\} \\
&\quad \cup \{u_j \leftarrow K_{j,1}, K_{j,2}, K_{j,3} \mid 1 \leqq j \leqq m\} \cup \{w \leftarrow u_j \mid 1 \leqq j \leqq m\} \\
&\quad \cup \{y \leftarrow v, w\}
\end{aligned}$$

with $y_1 \ldots y_n$, $u_1 \ldots u_m$, v, w and y new variables not occurring in $(\alpha \wedge \beta)$ and $R := \{y\} \cup \text{var}(\alpha \wedge \beta)$ (*fig.* 1).

Then the set of consequences with variables in R are

$$\Phi_1' = \{y \leftarrow L_{1,p_1}, \ldots, L_{n,p_n} \mid 1 \leqq p_1, \ldots, p_n \leqq 3\}$$

$$\Phi_2' = \{y \leftarrow L_{1,p_1}, \ldots, L_{n,p_n}, K_{r,1}, K_{r,2}, K_{r,3} \mid 1 \leqq p_1, \ldots, p_n \leqq 3, 1 \leqq r \leqq m\}$$

and obviously $\Phi_1' \models \Phi_2'$. Thus $\Phi_1 \overset{R}{\models} \Phi_2$ holds, yielding $\Phi_1 \approx_R \Phi_2$ if and only if $\Phi_2 \overset{R}{\models} \Phi_1$.

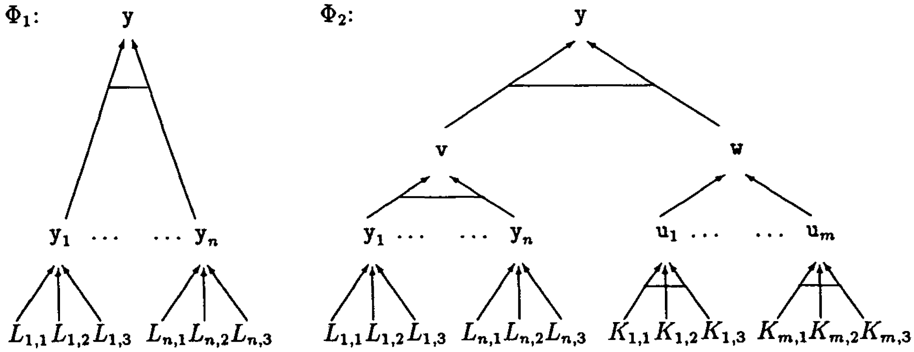

Figure 1

Further $\Phi_1 \approx_R \Phi_2$ if and only if $\Phi'_1 \approx \Phi'_2$. Let be $\mathfrak{J}(y)=1$. Then $\mathfrak{J}(\Phi'_i)=1$ and $\mathfrak{J}(\Phi'_i \cup \{\neg y\})=0$ hold for $i=1, 2$. But for the case $\mathfrak{J}(y)=0$ it holds, that $\Phi'_i[y/0]$ is the same formula as $(\Phi'_i \cup \{\neg y\})[y/0]$ for $i=1, 2$. Thus $\Phi'_1 \approx \Phi'_2$ if and only if $\Phi'_1 \cup \{\neg y\} \approx \Phi'_2 \cup \{\neg y\}$ if and only if $\Phi''_1 \approx \Phi''_2$ if and only if $\neg \Phi''_1 \approx \neg \Phi''_2$, where

$$\Phi''_1 := \{\neg L_{1,p_1} \vee \ldots \neg L_{n,p_n} \mid 1 \leqq p_1, \ldots, p_n \leqq 3\}$$

and

$$\Phi''_2 := \{\neg L_{1,p_1} \vee \ldots \vee \neg L_{n,p_n} \vee \neg K_{r,1} \vee \neg K_{r,2} \vee \neg K_{r,3} \mid 1 \leqq p_1, \ldots, p_n \leqq 3,\ 1 \leqq r \leqq m\}.$$

It remains to show $\neg \Phi''_1 \models \neg \Phi''_2$, if and only if $\alpha \models \neg \beta$ (*i.e.* $(\alpha \wedge \beta) \notin \mathrm{SAT}$).

Since Φ''_1 contains negative clauses only, there is a truth assignment $\mathfrak{J}'$ with $\mathfrak{J}'(\neg \Phi''_1)=1$.

Let $\neg \Phi''_1 \models \neg \Phi''_2$ be given. For $\mathfrak{J}(\alpha)=1$ in each clause $i=1, \ldots, n$ there is a literal L_{i,p_i}, $p_i \in \{1, 2, 3\}$ with $\mathfrak{J}(L_{i,p_i})=1$, such that $\mathfrak{J}(L_{1,p_1} \wedge \ldots \wedge L_{n,p_n})=1$. Hence $\mathfrak{J}(\neg \Phi''_1)=1$ and with $\neg \Phi''_1 \models \neg \Phi''_2$ we obtain $\mathfrak{J}(\neg \Phi''_2)=1$. Then there is a tuple $(p'_1, \ldots, p'_n) \in \{1, 2, 3\}^n$ and some $r \in \{1, \ldots, m\}$ with $\mathfrak{J}(L_{1,p'_1} \wedge \ldots \wedge L_{n,p'_n} \wedge K_{r,1} \wedge K_{r,2} \wedge K_{r,3})=1$ and therefore with $\mathfrak{J}(K_{r,1} \wedge K_{r,2} \wedge K_{r,3})=1$. Hence we get $\mathfrak{J}(\neg \beta)=1$.

For the other direction, let be given $\alpha \models \neg \beta$ and $\mathfrak{J}(\neg \Phi''_1)=1$. Then $\mathfrak{J}(L_{1,p_1} \wedge \ldots \wedge L_{n,p_n})=1$ for some $(p_1, \ldots, p_n) \in \{1, 2, 3\}^n$ and therefore $\mathfrak{J}(\alpha)=1$. Since $\alpha \models \neg \beta$ we get $\mathfrak{J}(\neg \beta)=1$ and $\mathfrak{J}(K_{r,1} \wedge K_{r,2} \wedge K_{r,3})=1$ for some $r \in \{1, \ldots, m\}$. This implies

$$\mathfrak{I}(L_{1,p_1} \wedge \ldots \wedge L_{n,p_n} \wedge K_{r,1} \wedge K_{r,2} \wedge K_{r,3}) = 1 \quad \text{and} \quad \mathfrak{I}(\neg \Phi_2'') = 1.$$

Since $\alpha \models \neg \beta$ if and only if $(\alpha \wedge \beta) \notin \mathrm{SAT}$, the proof is complete. ■

Since the restricted equivalence for propositional logic can be expressed in terms of quantified formulas with existential quantifiers only, the equivalence problem for existentially quantified Horn formulas is coNP-hard. Again it is easy to see, that the equivalence problem for QHorn formulas is in coNP by choosing some arbitrary truth assignment and then evaluating the remaining closed quantified Horn formula using the polynomial time bounded evaluation algorithm in [5].

COROLLARY 3.2: EQUIV(QHorn) *and* IMPL(QHorn) *are* coNP-*complete.*

3.2. 2CNF Formulas

For two 2CNF formulas α and β and a set of variables R the restricted equivalence $\approx_R$ can trivially be tested in time cubic in the length of the two formulas using the linear evaluation test from [1]. It suffices to show $(\alpha \models L_1 \vee L_2) \Leftrightarrow (\beta \models L_1 \vee L_2)$ for all $4|R|^2$ possible clauses of length 2 built with variables from R. In the following we will show that this test can also be done in quadratic time. (Since the best known test for the equivalence of propositional 2CNF formulas requires quadratic time, this seems to be the best one might expect.)

As will be shown afterwards, this algorithm can also be used to test the equivalence of Q2CNF formulas, since each Q2CNF formula can be transformed in quadratic time into an equivalent Q2CNF formula with the same set of free variables and existential quantifiers only.

THEOREM 3.3: *For two* 2CNF *formulas* α *and* β *and a set of variables* R *the restricted equivalence* $\alpha \approx_R \beta$ *and the restricted implication* $\alpha \overset{R}{\models} \beta$ *can be decided in time quadratic in the length of* α *and* β.

Proof: Since the evaluation problem for Q2CNF formulas and therefore the satisfiability of 2CNF formulas, is decidable in linear time [1], we assume that α and β are satisfiable 2CNF formulas. Further we assume $R = \{z_1, \ldots, z_s\} \neq \emptyset$. The algorithm consists of two parts and returns *true*, if $\alpha \approx_R \beta$. In the first part all literals over R that are consequences of α and β are considered. In the second part for each of the remaining literals over R all clauses of length 2 containing this literal and implied by α and β are tested. The algorithm consists of the following steps:

1. For $\pi \in \{\alpha, \beta\}$ build

$$\mathcal{U}(\pi) := [L \mid \pi \models L, \ L \in \operatorname{lit}(\operatorname{var}(\pi))\}$$

2. If $(\mathcal{U}(\alpha) \cap \operatorname{lit}(R)) \neq (\mathcal{U}(\beta) \cap \operatorname{lit}(\mathrm{R}))$ then return *false* and stop.

3. Transform $\alpha(\beta)$ into $\alpha^*(\beta^*)$ by deleting all clauses containing a literal $L \in \mathcal{U}(\alpha)$ $(L \in \mathcal{U}(\beta))$.

4. Build the graphs $G(\alpha^*)$ and $G(\beta^*)$.

A *graph* $G(\pi)$ of a formula $\pi \in 2\mathrm{CNF}$ with exactly two literals per clause, consists of the set of vertices $V = \{L \mid L \in \operatorname{lit}(\operatorname{var}(\pi))\}$ and the edges $E = \{(\bar{L}_1, L_2), (\bar{L}_2, L_1) \mid (L_1 \vee L_2)$ is a clause in $\pi\}$.

5. For each literal $L \in (\operatorname{lit}(R) \cap \operatorname{lit}(\operatorname{var}(\alpha^*) \cup \operatorname{var}(\beta^*)))$ do

5 *a* Build $\mathcal{R}(\alpha^*, L) := \{L' \mid L'$ reachable from L in $G(\alpha^*)\}$ and $\mathcal{R}(\beta^*, L)$ analogously.

5 *b* If $(\mathcal{R}(\alpha^*, L) \cap \operatorname{lit}(R)) \neq (\mathcal{R}(\beta^*, L) \cap \operatorname{lit}(R))$ then return *false* and stop.

6. Return *true* and stop.

To show the equivalence restricted to the variables in R, we now have to test $(\alpha \models L_0 \vee L_1) \Leftrightarrow (\beta \models L_0 \vee L_1)$ for all clauses $(L_0 \vee L_1)$ built from literals with variables in R. Using $(\pi \models \neg L_0 \vee L_1) \Leftrightarrow (L_0 \wedge \pi \models L_1)$ it suffices to show, that for all literals $L \in \operatorname{lit}(R)$ holds

$$(\mathcal{U}(L \wedge \alpha) \cap \operatorname{lit}(R)) = (\mathcal{U}(L \wedge \beta) \cap \operatorname{lit}(R)),$$

with $\mathcal{U}(\pi)$ defined as above.

This is done in two parts. In the steps 1, 2 and 3 all directly implied unit clauses are processed and in the steps 4 and 5 all remaining clauses with two different literals are considered.

Using the linear time algorithm from [1] the sets $\mathcal{U}(\alpha)$ and $\mathcal{U}(\beta)$ can be computed in quadratic time. This allows us to compare the literals (*i.e.* unit clauses) over variables from R implied by α respectively β.

Then the formulas are reduced by deleting all clauses subsumed by these unit clauses. The resulting formulas α^* and β^* will only contain clauses with exactly two literals. It remains to test

$$(\mathcal{U}(L \wedge \alpha^*) \cap \operatorname{lit}(R)) = (\mathcal{U}(L \wedge \beta^*) \cap \operatorname{lit}(R))$$

for all literals $L \in \operatorname{lit}(R)$ occurring in one of the two formulas.

$\mathcal{U}(L \wedge \alpha^*)$ can be computed in linear time. Since α^* does not imply any unit clause, each resolvent from α^* is a clause with two literals. Since $(L \wedge \alpha^*)$

is satisfiable, $(L \wedge \alpha^*) \models L'$ can be shown by linear resolution with start clause $\bar{L}'$ and last resolution step $L, \bar{L} \vdash \square$. So vertex L' is reachable from vertex L in the graph $G(\alpha^*)$, *i.e.* $\mathcal{U}(L \wedge \alpha^*) = \mathcal{R}(\alpha^*, L)$. Since all vertices reachable from a certain vertex L can be computed in linear time by a simple marking algorithm, step 5 requires quadratic time.

For the implication problem one only has to test

$$(\mathcal{U}(L \wedge \alpha) \cap \operatorname{lit}(R)) \supseteq (\mathcal{U}(L \wedge \beta) \cap \operatorname{lit}(R))$$

for all literals $L \in \operatorname{lit}(R)$. So one has to replace $\neq$ by $\not\supseteq$ in step 2 and to replace step 5 *b* by

5 *b'* If $((\mathcal{R}(\alpha^*, L) \cup \mathcal{U}(\alpha)) \cap \operatorname{lit}(R)) \not\supseteq (\mathcal{R}(\beta^*, L) \cap \operatorname{lit}(R))$ then return *false* and stop.

Since each step of the above algorithm requires at most quadratic time, the algorithm requires quadratic time. ■

Obviously this algorithm can also be used to test the equivalence (implication) of two satisfiable Q2CNF formulas Φ and Ψ with free variables and existential quantifiers only. One only has to omit the existential quantifiers and define $R := \operatorname{freevar}(\Phi) \cup \operatorname{freevar}(\Psi)$. Using the algorithm given in the proof of the following lemma we can also handle Q2CNF formulas with universal quantifiers.

In order to proof this lemma, we need the following definition and some results fromm [5].

Definition (Q-resolution): *Let* $\Phi = \Pi\alpha$ *with matrix* α *and prefix* Π *be a normalized* QCNF *formula*, i. e. Φ *contains no tautological clauses, there are no multiple occurrences of the same literal in a clause, pure* $\forall$*-clauses are replaced by the empty clause and in each other clause all* $\forall$*-literals are deleted, that are not before any* $\exists$*-literal occurring in this clause.*

Q-resolution step:

Let α_1 *be a clause with* $\exists$*-literal* y_l *and* α_2 *be a clause with* $\exists$*-literal* $\neg y_l$. *Then the Q*-resolvent σ *of* α_1 *and* α_2 *is obtained as follows:*

1. *Remove all occurrences of* y_l *and* $\neg y_l$ *in* α_1 *and* α_2 *obtaining the clauses* α_1' *and* α_2'.

2. *If the clause* $\alpha_1' \vee \alpha_2'$ *contains complementary literals, then no Q-resolvent exists. Otherwise remove all occurrences of* $\forall$*-literals, that are not before any* $\exists$*-literal occurring in* α_i' *to obtain* α_i'' $(i=1, 2)$. *Then the Q-resolvent is* $\sigma = \sigma_1'' \vee \alpha_2''$.

We write $\Phi \mathrel{\underset{Q-\mathrm{Res}}{\overset{1}{\vdash\!\!\!-\!\!\!-}}} \Pi(\alpha \wedge \sigma)$ *or* $\Phi \mathrel{\underset{Q-\mathrm{Res}}{\overset{1}{\vdash\!\!\!-\!\!\!-}}} \sigma$ *for short. By* $\mathrel{\underset{Q-\mathrm{Res}}{\vdash\!\!\!-\!\!\!-}}$ *we will denote the reflexive and transitive closure via Q-resolution.*

Note that only literals bound by existential quantifiers can be matched and universal variables not before an $\exists$-literal occurring in the same clause will be eliminated. The resulting clauses are normalized in the sense of the above definition and can be used for further Q-resolution steps.

In [5] it is shown that if $\Pi\alpha \mathrel{\underset{Q-\mathrm{Res}}{\overset{1}{\vdash\!\!\!-\!\!\!-}}} \Pi(\alpha \wedge \sigma)$ holds, then $\mathfrak{I}(\Pi\alpha)=1$ implies $\mathfrak{I}(\Pi(\alpha \wedge \sigma))=1$ and that for a closed QCNF formula holds, that Φ is false if and only if $\Phi \mathrel{\underset{Q-\mathrm{Res}}{\vdash\!\!\!-\!\!\!-}} \square$, *i.e.* a *refutation* of Φ via Q-resolution exists.

LEMMA 3.4: *For each satisfiable formula* $\Phi \in$ Q2CNF *a formula* $\Phi' \in$ Q2CNF *with* freevar$(\Phi)=$ freevar(Φ') *and* $\Phi \approx \Phi'$ *can be constructed in time quadratic in the length of* Φ, *such that* Φ' *does not contain universally quantified variables and length* $(\Phi')=O$ *(length* (Φ)).

Proof: Let be freevar$(\Phi)=\{z_1, \ldots, z_r\}$, let $\{y_1, \ldots, y_s\}$ be the existential variables of Φ, $\{x_1, \ldots, x_t\}$ the universal variables of Φ and $\Phi=Q_1 v_1 \ldots Q_{s+t} v_{s+t}(\varphi_1 \ldots \varphi_m)$.

Since each clause of the form $(Z_i \vee X_j)$ can be replaced by Z_i and each clause of the form $(Y_i \vee X_j)$, where Y_i is before X_j in the prefix, can be replaced by Y_i, we assume that Φ only contains $\forall$-literals in clauses where the other literal is an $\exists$-literal whose variable is governed by the variable of the $\forall$-literal. Since Φ is satisfiable, clauses of the form $(X_i \vee X_j)$ may not occur.

Then the rest of the algorithm consists of the following steps:

1. Build the set $\mathscr{U}_Z := \{ Z \mid \Phi \models Z,\ Z \in \mathrm{lit}(\mathrm{freevar}(\Phi)) \}$

2. Transform Φ into Φ^* by deleting all clauses containing a literal $Z \in \mathscr{U}_Z$ and all literals $\bar{Z}$ with $Z \in \mathscr{U}_Z$ of the remaining clauses.

3. Transform Φ^* into Φ^{**} by deleting all clauses containing a universal variable and deleting all universal variables in the prefix.

4. Transform Φ^{**} into Φ' by adding all literals in $\mathscr{U}_Z$ as unit-clauses.

Using the linear time evaluation test [1] for Q2CNF formulas the set $\mathscr{U}_Z$ can be constructed in quadratic time. Further the satisfiability of Φ guarantees the satisfiability of $\mathscr{U}_Z$ and Φ^* built in step 2.

Then $\Phi^* \not\models Z$ for all unit-clauses $Z \in \mathrm{lit}(\mathrm{freevar}(\Phi))$. So $(\Phi^*, \{z_i\})$ and $(\Phi^*, \{\neg z_i\})$ are satisfiable for $i=1, \ldots, r$. (Here $(\Phi, \{L_1, \ldots, L_p\})$ with

$\Phi = Q_1 v_1 \ldots Q_n v_n (\varphi_1 \wedge \ldots \wedge \varphi_m)$ denotes the formula

$$Q_1 v_1 \ldots Q_n v_n (\varphi_1 \wedge \ldots \wedge \varphi_m \wedge L_1 \wedge \ldots \wedge L_p)).$$

We now show that $\Phi^* \approx \Phi^{**}$ holds, where Φ^{**} is obtained from Φ^* by deleting all clauses that contain universal variables.

Let be $\mathfrak{J}$ an arbitrary truth assignment. Then we denote the closed Q2CNF formula $\Phi^* [z_1/\mathfrak{J}(z_1), \ldots, z_r/\mathfrak{J}(z_r)]$ by $\Phi^*_{\mathfrak{J}}$. $\Phi^{**}_{\mathfrak{J}}$ is used analogously.

It is obvious that $\mathfrak{J}(\Phi^*) = 1$ implies $\mathfrak{J}(\Phi^{**}) = 1$, because Φ^{**} is obtained from Φ^* by clause deletion. Thus, if no refutation of $\Phi^*_{\mathfrak{J}}$ via Q-resolution exists, there is also no refutation of $\Phi^{**}_{\mathfrak{J}}$.

To show the other direction, we assume that $\mathfrak{J}$ is a truth assignment with $\mathfrak{J}(\Phi^{**}) = 1$ and $\mathfrak{J}(\Phi^*) = 0$. Therefore $\Phi^*_{\mathfrak{J}}$ is false and $\Phi^{**}_{\mathfrak{J}}$ is true, *i.e.* $\Phi^*_{\mathfrak{J}} \vdash_{Q-\text{Res}} \square$ and $\Phi^{**}_{\mathfrak{J}} \not\vdash_{Q-\text{Res}} \square$.

The input clauses, *i.e.* the clauses from the matrix of $\Phi^*_{\mathfrak{J}}$, occurring in the refutation of $\Phi^*_{\mathfrak{J}}$, may not all occur in the matrix of $\Phi^{**}_{\mathfrak{J}}$. Otherwise the refutation of $\Phi^*_{\mathfrak{J}}$ would also be a refutation of $\Phi^{**}_{\mathfrak{J}}$. Thus, the refutation of $\Phi^*_{\mathfrak{J}}$ contains a clause of the form $(X_i \vee Y_j)$ from the matrix of $\Phi^*_{\mathfrak{J}}$ as input clause.

The satisfiability of Φ^* guarantees also, that not all of the input clauses in the refutation of $\Phi^*_{\mathfrak{J}}$ may occur in the matrix of Φ^*. Otherwise, the refutation of $\Phi^*_{\mathfrak{J}}$ would also show the unsatisfiability of Φ^*. Thus, the refutation of $\Phi^*_{\mathfrak{J}}$ contains a unit-clause (Y_k) as input clause, which was obtained from a clause $(Z_p \vee Y_k)$ in the matrix of Φ^* by replacing z_p with $\mathfrak{J}(z_p)$.

It is easy to see, that a refutation of $\Phi^*_{\mathfrak{J}}$, containing $(X_i \vee Y_j)$ and (Y_k) as input clauses, is of the form $(Y_k), (X_i \vee Y_j) \vdash^{1}_{Q-\text{Res}} \square$ with $Y_k = \bar{Y}_j$, or can be written as

$$(Y_k), (\bar{Y}_k \vee Y_{l_1}) \vdash^{1}_{Q-\text{Res}} (Y_{l_1}); \quad (Y_{l_1}), (\bar{Y}_{l_1} \vee Y_{l_2}) \vdash^{1}_{Q-\text{Res}} (Y_{l_2}); \ \ldots; (Y_{l_q}),$$

$$(\bar{Y}_{l_q} \vee \bar{Y}_j) \vdash^{1}_{Q-\text{Res}} (\bar{Y}_j); \quad (\bar{Y}_j), (X_i \vee Y_j) \vdash^{1}_{Q-\text{Res}} \square.$$

In both cases the refutation of $\Phi^*_{\mathfrak{J}}$ can be transformed into a derivation of the unit-clause (Z_p) in Φ^* by replacing the unit-clause (Y_k) by the clause

$(Z_p \vee Y_k)$ from Φ^*. In the first case $(Y_k = \bar{Y}_j)$ this yields

$$(Z_p \vee Y_k), \qquad (X_i \vee Y_j) \mathrel{\vdash\!\!\frac{\;1\;}{Q-\mathrm{Res}}} (Z_p)$$

and in the second case

$$(Z_p \vee Y_k), \qquad (\bar{Y}_k \vee Y_{l_1}) \mathrel{\vdash\!\!\frac{\;1\;}{Q-\mathrm{Res}}} (Z_p \vee Y_{l_1});$$

$$(Z_p \vee Y_{l_1}), (\bar{Y}_{l_1} \vee Y_{l_2}) \mathrel{\vdash\!\!\frac{\;1\;}{Q-\mathrm{Res}}} (Z_p \vee Y_{l_2}); \ldots; (Z_p \vee Y_{l_q}),$$

$$(\bar{Y}_{l_q} \vee \bar{Y}_j) \mathrel{\vdash\!\!\frac{\;1\;}{Q-\mathrm{Res}}} (Z_p \vee \bar{Y}_j); (Z_p \vee \bar{Y}_j), (X_i \vee Y_j) \mathrel{\vdash\!\!\frac{\;1\;}{Q-\mathrm{Res}}} (Z_p)).$$

But this is a contradiction to the fact, that $\Phi^* \nvDash Z$ for all unit-clauses $Z \in \mathrm{lit}(\mathrm{freevar}(\Phi))$ holds.

From this contradiction we get that $\mathfrak{J}(\Phi^{**}) = 1$ implies $\mathfrak{J}(\Phi^*) = 1$. Since Φ is equivalent to $(\Phi^*, \mathcal{U}_Z)$ and $\Phi^* \approx \Phi^{**}$, for $\Phi' := (\Phi^{**}, \mathcal{U}_Z)$ it holds that $\Phi \approx \Phi'$. By construction Φ' does not contain universal variables and $\mathrm{length}(\Phi') = O(\mathrm{length}(\Phi))$. ■

Using the construction given in the proof of the above lemma, we can show the following theorem.

THEOREM 3.5: EQUIV (Q2CNF) *and* IMPL (Q2CNF) *can be decided in time quadratic in the length of the two formulas.*

Proof: Let be given two Q2CNF formulas Φ and Ψ. The satisfiability of these formulas can be tested using the linear evaluation test from [1]. Using lemma 3.4 both formulas can be transformed into equivalent Q2CNF formulas Φ' and Ψ' with existential quantifiers only. By defining $R = \mathrm{freevar}(\Phi') \cup \mathrm{freevar}(\Psi')$ and setting the matrix of Φ' as α and the matrix of Ψ' as β, the algorithm for the restricted equivalence (restricted implication) of theorem 3.3 yields true if $(\Phi, \Psi) \in \mathrm{EQUIV}(\mathrm{Q2CNF})$ $((\Phi, \Psi) \in \mathrm{IMPL}$ $(\mathrm{Q2CNF}))$. Since each step requires at most quadratic time the test can be done in quadratic time. ■

4. CONCLUSION

Motivated by the problem of comparing logical descriptions with respect to a fixed set of names, we investigated the restricted equivalence problem and the restricted implication problem for Horn formulas and the equivalence problem for QHorn formulas. While these problems are shown to be intractable they are decidable in quadratic time for Q2CNF formulas. Our restriction to these classes was motivated by the fact, that the evaluation problem of these classes of quantified Boolean formulas is decidable in polynomial time. An important part in the algorithm for Q2CNF formulas is the transformation into an equivalent Q2CNF formula with existential quantifiers only. Since the best known algorithm for the equivalence of propositional 2CNF formulas requires quadratic time, the quadratic time algorithm for EQUIV (Q2CNF) seems to be the best one might expect.

REFERENCES

1. B. ASPVALL, M. F. PLASS and R. E. TARJAN, A linear-time algorithm for testing the truth of certain quantified boolean formulas, *Information Processing letters, 8*, 1979, pp. 121-123.
2. S. van DENNEHEUVEL and K. L. KWAST, Weak equivalence for constraint sets, *Proc. IJCAI*-91, *Sidney, Australia*, Vol. 2, 1991, pp. 851-856.
3. E. M. GOLD, Complexity of automaton identification from given data, *unpublished manuscript*, 1974.
4. J. N. HOOKER, Logical inference and polyhedral projection, to appear in *Proc. CSL'*91, Springer LNCS, 1991.
5. H. KLEINE BÜNING, M. KARPINSKI and A. FLÖGEL, Resolution for quantified boolean formulas, to appear in *Information and Computation*.
6. L. J. STOCKMEYER and A. R. MEYER, World problems requiring exponential time, *Proc. 5th Ann. ACM Symp. Theory of Computing*, 1973, pp. 1-9.
7. L. J. STOCKMEYER, The polynomial-time hierarchy, *Theoretical computer Science, 3*, 1977, pp. 1-22.
8. G. S. TSEITIN, On the complexity of derivations in propositional calculus, in A. O. Slisenko (ed.): *Studies in constructive mathematics and mathematical logic*, Consultants bureau, New York, 1970, part II, pp. 115-125.