数理解析研究所講究録 1252 巻 2002 年 218-225

SOME CLASSES OF BOOLEAN FORMULAE WITH POLYNOMIAL TIME SATISFIABILITY TESTING*

Ondřej Čepek
Department of Theoretical Informatics and Mathematical Logic
Charles University
cepek@ksi.ms.mff.cuni.cz

November 9, 2001

Abstract

This short note deals with several classes of Boolean formulae which have the property, that satisfiability can be tested for them in polynomial time with respect to the length of the formula. The studied classes known from the literature build up on the well-known classes of quadratic on Horn formulae. We prove several interesting properties of these classes and show their mutual positions with respect to inclusion, a problem which was not previously studied.

1 Introduction

The class of Horn formulae is a very important and extensively studied subclass of general Boolean formulae. The principal reason for their importance is the fact, that the satisfiability problem (SAT), which is well-known to be NP-complete for general Boolean formulae, can be solved efficiently (in linear time with respect to the length of the formula) for Horn formulae [11, 15, 17]. This has significant practical implications. Many real-life problems require for their solution to solve SAT as a subproblem, and hence are in general intractable; however, they become tractable if the underlying Boolean formula in the problem is Horn. Such problems arise in several application areas, among others in artificial intelligence [9, 13, 14] and database design [10, 16]. The limiting factor in using Horn formulae is their expressing power. Not every real-life problem can be formulated in such a way, that the underlying formula is Horn.

For the above reasons it is obvious, that finding broader classes of formulae, which preserve the property that satisfiability is decidable for them in polynomial time, is highly desirable. Several attempts in this direction were successfully made. The first natural generalization that was considered is the class of ***hidden Horn*** formulae, which are in literature sometimes also called ***renameable*** or ***disguised*** Horn formulae. This class consists of formulae, which can be obtained from Horn formulae by so called "variable complementing" (also known as "variable renaming" or "variable switching"), i.e. by replacing some Boolean variables by their complements. Aspvall showed in [1] that recognizing whether a given Boolean formula is hidden Horn can be done in linear time. Moreover, the recognition algorithm combined with the linear time SAT algorithm for Horn formulae [11, 15, 17] directly yields a linear time SAT algorithm for the class of hidden Horn formulae.

Yamasaki and Doshita [19] defined a different generalization of Horn formulae, called their class S_0, and developed a cubic time SAT algorithm for formulae in S_0. This was later improved to quadratic time by Arvind and Biswas [3]. Moreover, recognizing whether a given formula belongs to S_0 can be decided also in quadratic time by a straightforward algorithm which uses in a simple way the definition of the class. The class S_0 was further generalized by Gallo and Scutellà [12] who came up with a recursively defined infinite hierarchy of classes of Boolean formulae $\Gamma_0 \subseteq \Gamma_1 \subseteq \Gamma_2 \subseteq \cdots$ such that Γ_0 consists of all

*This research was supported by the Grant Agency of the Czech Republic - Grant 201/01/0942

Horn formulae, Γ_1 equals to S_0, and the union $\cup_{k=0}^{\infty}\Gamma_k$ is the set of all Boolean formulae. For every fixed k, recognizing whether a given formula belongs to Γ_k can be done in polynomial time (in $O(\ell n^k)$ time where ℓ is the length of the formula and n is the number of Boolean variables). If a formula is in the class Γ_k, then the same time bound holds for testing its satisfiability.

Another generalization of Horn formulae was defined by Boros, Crama, and Hammer in [4], where the class of *q-Horn* formulae was introduced. This class properly contains not only all Horn formulae, but also all quadratic formulae and hidden Horn formulae. In [4] it was shown, that satisfiability can be tested in linear time for q-Horn formulae, and recognizing whether a given formula is q-Horn can be done in polynomial time by an algorithm based on linear programming. The complexity of recognition was later improved by Boros, Hammer, and Sun [5] to linear time by means of a purely combinatorial algorithm.

Yet another generalization of Horn formulae, so called *extended Horn (EH)* and *hidden extended Horn (HEH)* formulae, were defined by Chandru and Hooker in [8]. The definition of the class of EH formulae uses nontrivial polyhedral techniques and is quite complicated. HEH formulae then originate from EH formulae in the same way as hidden Horn formulae do from Horn formulae. The main property of HEH formulae is that satisfiability can be tested for them in linear time by unit resolution. On the other hand, the biggest drawback of the class is that no polynomial time recognition procedure is known for it, with the exception of a small subclass of EH formulae for which recognition can be solved in almost linear time by the algorithm of Swaminathan and Wagner [18].

Very little is known about the mutual relationships (with respect to inclusion) among the above described classes (the exception being that both q-Horn formulae and HEH formulae are known to contain all hidden Horn formulae). In this paper we shall address this question and show, that all of the above defined classes (S_0, q-Horn, HEH) are indeed different, i.e. none of them contains any other. Furtermore we shall show how do the classes of q-Horn formulae and HEH formulae relate to the infinite hierarchy $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ defined by Gallo and Scutellà [12].

2 Classes with polynomial time satisfiability testing

Throughout this paper we shall work with the set $x_1, x_2, \ldots, x_n$ of *Boolean variables* (proposition letters). A *literal* is either a variable or its negation. The set of all *positive* literals $x_1, x_2, \ldots, x_n$ and all *negative* literals $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$ will be denoted by I, i.e.

$$I = \{x_1, x_2, \ldots, x_n, \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n\}.$$

For every i, the pair $x_i, \overline{x}_i$ is called a *complementary pair* of literals. A *clause* is a disjunction of literals which contains no complementary pair. A clause is called a *positive clause* if it contains only positive literals and it is called a *negative clause* if it contains only negative literals. A *length* of a clause is the number of literals in it. In the subsequent text we shall frequently treat clauses as sets, e.g the expression $a \in C$ will denote that clause C contains literal a. Similarly $C \subseteq D$ will mean that all literals of clause C are contained also in clause D, in which case C is called a *subclause* of D. A *conjunctive normal form (CNF)* is a conjunction of clauses. It is well-known, that every Boolean formula (of propositional logic) can be transformed into a logically equivalent CNF. Thus, in the remainder of this paper we shall work only with CNFs, and the word "formula" will always mean a CNF. A *length* of a formula is then defined as the sum of lengths of its clauses.

A *model* is an assignment of truth values to variables which extends in an obvious way to an assignment of truth values to literals (complementary literals receive complementary values). A model *satisfies* a clause if it makes at least one literal in the clause true. A formula is *satisfiable* if there exists a model which simultaneously satisfies all clauses in the formula. The *satisfiability problem (SAT)* is defined as follows:

Instance: A formula ϕ.
Question: Is ϕ satisfiable?

SAT is known to be NP-complete, however it is solvable in polynomial time for certain classes of formulae. Perhaps the simplest such class is the class of *quadratic* formulae. A formula is quadratic if every clause in it has length at most two. It was proved e.g. in [2] that in such a case SAT can be solved in linear time with respect to the length of the formula. Beside that, the class of quadratic formulae (let us denote it by Q) has additional nice properties, namely it is closed under the following five operations:

- *Literal deletion:* Let $\phi \in Q$ and let ϕ' originate from ϕ by deleting a literal from some clause. Then $\phi' \in Q$.
- *Clause deletion:* Let $\phi \in Q$ and let ϕ' originate from ϕ by deleting an entire clause. Then $\phi' \in Q$.
- *Partial assignment:* Let $\phi \in Q$ and let ϕ' originate from ϕ by substituting a truth value for a variable. Obviously, this simply amounts to literal deletion of all occurences of the selected variable which evaluate to zero, and to clause deletion of all clauses in which the selected variable evaluates to 1. Hence any class of formulae which is closed under both literal and clause deletion is closed also under partial assignment. Thus, $\phi' \in Q$.
- *Variable complementation:* Let $\phi \in Q$ and let $S \subseteq \{1, \ldots, n\}$ be an index set indexing a subset of variables. Define a formula ϕ^S as follows: ϕ^S is produced from ϕ by replacing all occurences of x_i by $\overline{x}_i$ and all occurences of $\overline{x}_i$ by x_i for every $i \in S$, and by leaving all other literals (corresponding to variables x_i, $i \notin S$) unchanged. Then $\phi^S \in Q$.
- *Disjoint union:* Let $\phi_1, \phi_2 \in Q$ be two formulae on disjoint sets of variables, and let $\phi = \phi_1 \wedge \phi_2$. Then $\phi \in Q$ (the class Q is closed even under a "general" union, where the two sets of variables are not required to be disjoint, however).

The above five operations are very useful in working with examples of formulae which belong to a given class. Therefore we shall study these operations for all classes which we shall work with in the subsequent text. However, as we shall see, not all of these classes will behave as "nicely" as the class of quadratic functions, which is closed under all five operations. The biggest drawback of quadratic formulae is their low "expressing power", i.e. few "real world" problems can be formulated in terms of quadratic formulae. A widely studied class of formulae with a considerably higher expressing power is the class of *Horn* formulae. A clause is Horn if it contains at most one positive literal. A formula is Horn if it consists only of Horn clauses. Again, SAT can be solved in linear time for Horn formulae, as was shown e.g. in [11, 15, 17]. The class of Horn formulae is clearly closed under both literal and clause deletion and hence also under partial assignment. It is also closed under disjoint union (even "general" union). However, it is not closed under variable complementation. This feature is rather unfortunate. A more or less random choice of which "real world" phenomenon is associated with a positive literal and which with the corresponding negative literal (e.g. for dual pairs like light/dark, switch-on/switch-off, etc.) may influence whether the resulting Boolean formulation of the underlying "real-world" problem yields a Horn formula or not. That in turn determines whether the obtained formulation is practically usable or not. This drawback of Horn formulae is eliminated in the following class, which can be thought of as the "complementation closure" of the class of Horn formulae.

2.1 Hidden Horn formulae

A formula ϕ is *hidden Horn* if there exists an index set S for which ϕ^S is Horn. The following easy observation follows immediately from the definition and the properties of Horn formulae and hence is left without a proof.

Proposition 2.1 *The class of hidden Horn formulae is closed under literal deletion, clause deletion, partial assignment, variable complementation, and disjoint union.*

Note however, that unlike in the Horn case, the class of hidden Horn formulae is not closed under "general" union, as two hidden Horn formulae may require conflicting sets of variables to be complemented (no class introduced from now on will be closed under "general" union, so we will stop referring to it). In [1] a linear time algorithm was designed, which for any given formula ϕ tests whether it is hidden Horn, and in the positive case outputs the appropriate index set S, such that ϕ^S is Horn. Since there is an obvious one-to-one correspondence between satisfying models of ϕ and satisfying models of ϕ^S, this linear time recognition algorithm combined with any linear time SAT algorithm for Horn formulae immediately yields a linear time algorithm for SAT on hidden Horn formulae. Now let us define another three different generalizations of Horn formulae: q-Horn, (hidden) extended Horn, and S_0 formulae.

2.2 Q-Horn formulae

An assignment of truth values to variables, which can be simply defined as a function $t : \{x_1, \ldots, x_n\} \longrightarrow \{0,1\}$ can be generalized to a so called *valuation* $\alpha : \{x_1, \ldots, x_n\} \longrightarrow [0,1]$ by relaxing the integrality requirement. Similarly as in the case of truth value assignments, a valuation extends to all literals by requiring that $\forall i : \alpha(x_i) + \alpha(\overline{x}_i) = 1$. For a clause C we define $\alpha(C) = \sum_{a \in C} \alpha(a)$, and call valuation α to be *feasible* for C if $\alpha(C) \leq 1$. A formula ϕ is *q-Horn* if there exists a valuation α which is feasible for all clauses in ϕ (such a valuation is then called feasible for ϕ).

It is easy to observe, that every quadratic formula is q-Horn (the valuation $\alpha(x_i) = \frac{1}{2}\ \forall i$ is feasible for every quadratic formula), every Horn formula is q-Horn (the valuation $\alpha(x_i) = 1\ \forall i$ is feasible), and every hidden Horn formula is q-Horn (take the valuation $\alpha(x_i) = 0\quad \forall i \in S$ and $\alpha(x_i) = 1\quad \forall i \notin S$, where S is the index set for which ϕ^S is Horn). Moreover, the class of q-Horn formulas has the following properties.

Proposition 2.2 *The class of q-Horn formulae is closed under literal deletion, clause deletion, partial assignment, variable complementation, and disjoint union.*

Proof: The fact that the class of q-Horn formulae is closed under both literal and clause deletion and hence also under partial assignment follows directly from the definition of q-Horn formulae. The same is true for disjoint union. To see that the class is closed under variable complementation note, that if valuation α is feasible for ϕ then valuation α' defined by $\alpha'(x_i) = 1 - \alpha(x_i)\quad \forall i \in S$ and $\alpha'(x_i) = \alpha(x_i)\quad \forall i \notin S$ is feasible for ϕ^S. Hence, if ϕ is q-Horn then ϕ^S is also q-Horn for all index sets S. ■

In [4], where the class of q-Horn formulae was introduced, it was shown, that for a q-Horn formula SAT can be tested in linear time (with respect to the length of the formula), and recognizing whether a given formula is q-Horn can be done in polynomial time by an algorithm based on linear programming. The complexity of recognition was later improved in [5] to linear time by means of a purely combinatorial algorithm.

2.3 (Hidden) Extended Horn formulae

The definition of this class of formulae utilizes polyhedral techniques, namely the results from [7], and is quite complicated. Because of the required brevity of this note, it is not possible to give the full details of the definition here. The interested reader shall look into the original paper [8] or into [6] for a summary. To put the class of hidden extended Horn (HEH) formulae in perspective with other classes discussed in this note, let us state two propositions from [6].

Proposition 2.3 *The class of hidden extended Horn formulae is closed under clause deletion, partial assignment, variable complementation, and disjoint union, however, it is not closed under literal deletion.*

Proposition 2.4 *The class of hidden extended Horn formulae properly contains the class of hidden Horn formulae.*

Although the definition of HEH formulae is rather complicated and difficult to grasp, the class behaves remarkably nice with respect to solving SAT. It was discovered in [8] that SAT can be solved for HEH formulae in linear time by an algorithm based on unit resolution. On the other hand, the biggest drawback of the class of HEH formulae is that no polynomial time recognition procedure is known for it, with the exception of a small subclass of extended Horn formulae for which recognition can be solved in almost linear time by the algorithm of Swaminathan and Wagner [18].

2.4 Class S_0 and infinite hierarchy $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$

The definition of class S_0 is quite simple. A formula ϕ is in the class S_0 if there exists an ordering $\{C_1, \ldots, C_m\}$ of the clauses of ϕ such that each C_i can be written in the form $C_i = P_i \vee H_i$ where

1. $\forall i = 1, \ldots, m\ :\ H_i$ is a Horn clause,

2. $\forall i = 1, \ldots, m \ : \ P_i$ is a positive clause, and

3. $\forall i = 1, \ldots, m-1 \ : \ P_{i+1} \subseteq P_i$.

Quite clearly, S_0 contains all Horn formulae. On the other hand, unlike in the case of q-Horn and HEH formulae, the set of hidden Horn formulae is not contained in S_0. To see this it is enough to consider e.g. the formula

$$\phi = (x_1 \vee x_2) \wedge (x_3 \vee x_4) \tag{1}$$

which is certainly hidden Horn (complementing e.g. x_1 and x_3 suffices to get a Horn formula), but is not in S_0 (the sets P_1 and P_2 must contain at least one literal each, and hence can never fulfil the required inclusion). This observation immediately implies the following easy statement.

Proposition 2.5 *The class S_0 is closed under clause deletion and partial assignment. It is not closed under literal deletion, variable complementation, and disjoint union.*

Proof: Let us start with the negative results. S_0 is not closed under literal deletion because the formula $\phi' = (x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4)$ is in S_0 (setting e.g. $P_1 = x_2 \vee x_3$ and $P_2 = x_3$ proves it), while formula ϕ which is obtained from ϕ' by deleting literal x_3 from the first clause is not in S_0. Similarly, S_0 is not closed under variable complementation since the above formula ϕ can be obtained by variable complementation e.g. from $(\overline{x}_1 \vee x_2) \wedge (\overline{x}_3 \vee x_4)$ which is Horn and thus also in S_0. Finally, S_0 is not closed under disjoint union because $(x_1 \vee x_2)$ as well as $(x_3 \vee x_4)$ (as well as every formula consisting of a single clause) is in S_0 while ϕ is not.

On the other hand, it easily follows from the definition that the class S_0 is closed under clause deletion, and it is not hard to see that it is closed under partial assignment. Indeed, removing a literal from all clauses in which it appears preserves the required nesting of the positive clauses. ■

In [19] a $O(n^3)$ SAT algorithm (where n is the number of variables) was developed for formulae in S_0, which was later improved to $O(n^2)$ in [3]. Although the recognition problem was not addressed in [19] or [3], recognizing whether a given formula belongs to S_0 can be decided also in quadratic time by a straightforward algorithm which uses in a simple way the definition of the class. However, we shall not present this algorithm here, because we shall see later, that the recognition problem for S_0 is just a special case of a more general recognition problem (solved in [12]), which we shall deal with below.

The definition of class S_0, which is based on the idea of nested positive clauses, was further generalized in [12] in the following way. Let ϕ be a formula consisting of clauses $\{C_1, \ldots, C_m\}$ on the set of variables $X = \{x_1, \ldots, x_n\}$. Let us write each clause C_i in the form $C_i = P_i \vee N_i$, where P_i is a positive clause and N_i is a negative clause. Furthermore, let us denote $P(\phi) = \{P_1, \ldots, P_m\}$ and let J be an arbitrary subset of variables, i.e. $J \subseteq X$. Then we define two set operations, which use J to restrict $P(\phi)$ in two different ways to obtain "smaller" sets of positive clauses, by

- $P(\phi)_J = P(\phi) \setminus \{P_i \in P(\phi) \mid J \subseteq P_i\}$, and
- $P(\phi)\Theta J = \{P_i \setminus J \mid P_i \in P(\phi)\}$.

The above two operations enable us to recursively define an infinite hierarchy of sets of positive clauses $\Sigma_0, \Sigma_1, \Sigma_2, \ldots$ as follows

- $P(\phi) \in \Sigma_0$ iff $\forall P_i \in P(\phi) \ : \ |P_i| \leq 1$,
- $\forall k > 0 : P(\phi) \in \Sigma_k$ iff $\exists x_j \in X \ : \ P(\phi)_{\{x_j\}} \in \Sigma_{k-1}$ and $P(\phi)\Theta\{x_j\} \in \Sigma_k$

The above hierarchy of sets of positive clauses can be extended in a natural way to an infinite hierarchy of classes of Boolean formulae $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ by

$$\phi \in \Gamma_k \equiv P(\phi) \in \Sigma_k.$$

Obviously, Γ_0 consists exactly of all Horn formulae, and it is not hard to see that Γ_1 equals to S_0 (see [12] for details). Moreover, the following claims were proved in [12]. If ϕ is a formula of length ℓ on n variables then

- $\phi \in \Gamma_n$ and hence $\cup_{k=0}^{\infty}\Gamma_k$ contains all formulae,
- recognizing whether $\phi \in \Gamma_k$ can be done in $O(\ell n^k)$ time, and
- if $\phi \in \Gamma_k$ then SAT for ϕ can be tested in $O(\ell n^k)$ time as well.

To simplify notation and to avoid the need to switch back and forth between ϕ and $P(\phi)$ we shall for a set J of positive literals denote by

- ϕ_J the formula originating from ϕ by removing every clause which contains all literals in J,
- $\phi \Theta J$ the formula originating from ϕ by removing all occurences of all literals in J.

3 Mutual relationships with respect to inclusion

Let us denote by HH, QH, and HEH the classes of all hidden Horn, q-Horn, and hidden extended Horn formulae respectively. The mutual relationships of these classes as well as of the class S_0 with respect to inclusion were completely established in [6], where it was proved, that the classes QH, HEH, and S_0 are "in a general position", i.e. all eight sets defined by the partitioning of the set of all formulae by QH, HEH, and S_0 are nonempty.

In this note let us turn our attention to the relationship of HH and the infinite hierarchy $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ A natural question is whether there exists an index k such that $HH \subseteq \Gamma_k$. We shall provide a negative answer to this question by showing that for every index k, there exists a hidden Horn formula ψ_k, such that $\psi_k \in (\Gamma_{k+1} \setminus \Gamma_k)$. Let us define these formulae by

$$\forall k = 0,1,2,\ldots : \psi_k = \bigwedge_{i=0}^{k} (x_1^i \vee x_2^i) \tag{2}$$

Lemma 3.1 $\forall k = 0,1,2,\ldots : \psi_k \in HH \cap (\Gamma_{k+1} \setminus \Gamma_k)$.

Proof : First of all, for every k the formula ψ_k consists of only positive literals, and thus it is hidden Horn (by complementing all variables we get a Horn formula). Let us proceed by induction on k to show that $\psi_k \in (\Gamma_{k+1} \setminus \Gamma_k)$.

- Case $k = 0$. Clearly, $\psi_0 = x_1^0 \vee x_2^0$ is in S_0 but is not Horn. Thus $\psi_0 \in (\Gamma_1 \setminus \Gamma_0)$.
- Case $k = 1$. Note that $\psi_1 = (x_1^0 \vee x_2^0) \wedge (x_1^1 \vee x_2^1)$ is just (up to a renaming of variables) formula (1) from Section 2.4. Thus $\psi_1 \notin \Gamma_1$. To see that $\psi_1 \in \Gamma_2$ it suffices to verify that there exists a variable x such that $(\psi_1)_{\{x\}} \in \Gamma_1$ and $(\psi_1)\Theta\{x\} \in \Gamma_2$. Let us take $x = x_2^1$. Then $(\psi_1)_{\{x\}} = (x_1^0 \vee x_2^0) = \psi_0 \in \Gamma_1$ and $(\psi_1)\Theta\{x\} = (x_1^0 \vee x_2^0) \wedge (x_1^1)$. This formula is clearly in Γ_2 since by taking $y = x_1^1$ we get $((\psi_1)\Theta\{x\})_{\{y\}} = \psi_0 \in \Gamma_1$ and $((\psi_1)\Theta\{x\})\Theta\{y\} = \psi_0 \in \Gamma_1 \subseteq \Gamma_2$.
- Let the statement be true for $0, 1, \ldots, k-1$ and let us assume by contradiction that $\psi_k \in \Gamma_k$. That means that there exists a variable x such that $(\psi_k)_{\{x\}} \in \Gamma_{k-1}$. However, regardless of the choice of x, the formula $(\psi_k)_{\{x\}}$ is just (up to a renaming of variables) the formula ψ_{k-1}, which is a contradiction to the induction hypothesis. Thus $\psi_k \notin \Gamma_k$. To show that $\psi_k \in \Gamma_{k+1}$ we can simply repeat step by step the proof that $\psi_1 \in \Gamma_2$, only with $x = x_2^k$, $y = x_1^k$, and Γ_k and Γ_{k+1} taking the roles of Γ_1 and Γ_2.

■

Corollary 3.2 $\forall k = 0,1,2,\ldots$:

1. $(QH \cap HEH) \cap (\Gamma_{k+1} \setminus \Gamma_k) \neq \emptyset$

2. $(HEH \setminus QH) \cap (\Gamma_{k+1} \setminus \Gamma_k) \neq \emptyset$

3. $(QH \setminus HEH) \cap (\Gamma_{k+1} \setminus \Gamma_k) \neq \emptyset$

Proof : Since $HH \subseteq (QH \cap HEH)$ the first part of the statement follows directly from Lemma 3.1. Note that formulae ψ_k, $k = 1, 2, \ldots$ are obtained by succesively adding "copies" of the formula $\phi = \psi_0 = x_1 \vee x_2$ for which $\phi \in HH \cap (\Gamma_1 \setminus \Gamma_0)$ holds. The second and third parts of the statement can be proved in a similar manner as Lemma 3.1 where the role of ϕ is taken by the formulae $\phi_2 = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3)$ and $\phi_3 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2)$. We have already shown that $\phi_2 \in \overline{QH} \cap HEH \cap S_O$ and hence also $\phi_2 \in (HEH \setminus QH) \cap (\Gamma_1 \setminus \Gamma_0)$ (since $\phi_2 \notin QH$ it also cannot be Horn, i.e. in Γ_0). Similarly $\phi_3 \in (QH \setminus HEH) \cap (\Gamma_1 \setminus \Gamma_0)$. The rest of the proof is similar to the proof of Lemma 3.1 (although it is more technical because formulae ϕ_2 and ϕ_3 are more complicated than formula ϕ) and is left to the reader as an excercise. ■

We have remarked in the proof of Lemma 3.1 that formulae ψ_k, $k = 1, 2, \ldots$, play a similar role for classes Γ_k as formula (1) from Section 2.4 did for S_0 (indeed $S_0 = \Gamma_1$ and (1) after a proper renaming of variables is just ψ_1). Thus, Lemma 3.1 gives us enough material to prove the properties of class Γ_k, which, as we shall see, are the same as the properties of S_0 specified in Proposition 2.5.

Proposition 3.3 *The class Γ_k, $k = 1, 2, \ldots$, is closed under clause deletion and partial assignment. It is not closed under literal deletion, variable complementation, and disjoint union.*

Proof: Let us again start with the negative results. First let us show that Γ_k is not closed under literal deletion. Let $\psi' = \psi_{k-1} \wedge (x_2^{k-1} \vee x_1^k \vee x_2^k)$. We shall prove that ψ' is in Γ_k while ψ_k which is obtained from ψ' by deleting literal x_2^{k-1} from the last clause is not in Γ_k (the latter follows from Lemma 3.1). To prove that $\psi' \in \Gamma_k$ we have to find x such that $\psi'_{\{x\}} \in \Gamma_{k-1}$ and $\psi'\Theta\{x\} \in \Gamma_k$. Let us take $x = x_2^{k-1}$. Then $\psi'_{\{x\}} = \psi_{k-2} \in \Gamma_{k-1}$ (by Lemma 3.1) and $\psi'\Theta\{x\} = \psi_{k-2} \wedge (x_1^{k-1}) \wedge (x_1^k \vee x_2^k)$. Let us denote this last formula by ψ''. To prove that it is indeed in Γ_k as required, it suffices to take $y = x_1^{k-1}$. Then $\psi''_{\{x\}} = \psi''\Theta\{x\} = \psi_{k-1}$ (up to a relabeling of variables) which is in Γ_k by Lemma 3.1.

Proving that Γ_k is not closed under variable complementation and disjoint union is easy. It is enough to observe that formula ψ_k can be obtained by variable complementation e.g. from a Horn formula $\bigwedge_{i=0}^k (\overline{x}_1^i \vee x_2^i)$, as well as by a disjoint union of ψ_{k-1} and $(x_1^k \vee x_2^k)$ (which are both in Γ_k).

To show that Γ_k is closed under clause deletion, we shall proceed by a double induction on k and on the number of variables which appear as positive literals in the formula. For the basic step notice, that the statement is true for $\Gamma_1 = S_0$ as well as for all formulas with only one variable appearing as positive literals (those are all Horn). Now let $\phi \wedge C \in \Gamma_k$ where ϕ is a formula and C is a clause. We want to prove that $\phi \in \Gamma_k$. Let x be such that $(\phi \wedge C)_{\{x\}} \in \Gamma_{k-1}$ and $(\phi \wedge C)\Theta\{x\} \in \Gamma_k$. We shall show that also $\phi_{\{x\}} \in \Gamma_{k-1}$ and $\phi\Theta\{x\} \in \Gamma_k$. The first claim follows from the fact that $\phi_{\{x\}}$ consists of a subset of clauses of the formula $(\phi \wedge C)_{\{x\}}$ and Γ_{k-1} is closed under clause deletion by the induction hypothesis. The second claim follows similarly. The formula $\phi\Theta\{x\}$ consists of a subset of clauses of the formula $(\phi \wedge C)\Theta\{x\}$, which is a formula in Γ_k with a smaller number of variables appearing as positive literals than $\phi \wedge C$. Thus by the induction hypothesis $\phi\Theta\{x\} \in \Gamma_k$.

Finally, let us show that Γ_k is closed under partial assignment. Let $\phi \in \Gamma_k$ and let x be a variable in ϕ. Setting $x = 1$ amounts to deleting all clauses containing the literal x (and that leaves the formula in Γ_k as shown above) and to deleting all occurences of literal $\overline{x}$ (negative literals have no effect on belonging to Γ_k). Setting $x = 0$ amounts to deleting all clauses containing the literal $\overline{x}$ (again, that leaves the formula in Γ_k as shown above) and to deleting all occurences of literal x. We have to show that the last operation also leaves the formula in Γ_k, i.e. in our notation we have to prove that for every variable x, $\phi\Theta\{x\} \in \Gamma_k$. Once more, this claim will be proven by a double induction on k and on the number of variables which appear as positive literals in the formula. The basic step is again trivial. For the induction step note, that since $\phi \in \Gamma_k$, there exists y such that $\phi_{\{y\}} \in \Gamma_{k-1}$ and $\phi\Theta\{y\} \in \Gamma_k$. If $x = y$ we are done and if $x \neq y$ it is enough to show that $(\phi\Theta\{x\})_{\{y\}} \in \Gamma_{k-1}$ and $(\phi\Theta\{x\})\Theta\{y\} \in \Gamma_k$. The first claim follows from the fact that $(\phi\Theta\{x\})_{\{y\}} = (\phi_{\{y\}})\Theta\{x\}$ (it does not matter whether we first delete clauses containing y and then all literals x or vice versa) and the induction hypothesis for Γ_{k-1}. The second claim follows from a similar observation that $(\phi\Theta\{x\})\Theta\{y\} = (\phi\Theta\{y\})\Theta\{x\}$ and the induction hypothesis for Γ_k and formulae with a smaller number of variables appearing as positive literals. ■

References

[1] B. Aspvall. Recognizing disguised NR(1) instances of the satisfiability problem. *Journal of Algorithms*, **1**(1980), 97-103.

[2] B. Aspvall, M.F. Plass, and R.E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formula. *Information Processing Letters*, **8**(1979), 121-123.

[3] V. Arvind and S. Biswas. An $O(n^2)$ algorithm for the satisfiability problem of a subset of propositional sentences in CNF that includes all Horn sentences. *Information Processing Letters*, **24**(1987), 67-69.

[4] E. Boros, Y. Crama, and P.L. Hammer. Polynomial-time inference of all valid implications for Horn and related formulae. *Annals of Mathematics and Artificial Intelligence*, **1**(1990), 21-32.

[5] E. Boros, P.L. Hammer, and X. Sun. Recognition of q-Horn formulae in linear time. *Discrete Applied mathematics*, **55**(1994), 1-13.

[6] O. Čepek. Generalizations of Horn formulae and their mutual relationships. *in Proceedings of the 4th Czech-Japan seminar on data analysis and decision making under uncertainty*, (2001), 3-21.

[7] R. Chandrasekaran. Integer programming problems for which a simple rounding type of algorithm works. In W.R.Pulleyblank, ed. *Progress in Combinatorial Optimization*, Academic Press Canada, Toronto, Ontario, Canada (1984), 101-106.

[8] V. Chandru and J.N. Hooker. Extended Horn sets in propositional logic. *Journal of the ACM*, **38**(1991), 205-221.

[9] R. Dechter and J. Pearl. Structure identification in relational data. *Artificial Intelligence*, **58**(1992), 237-270.

[10] C. Delobel and R.G. Casey. Decomposition of a Data Base and the Theory of Boolean Switching Functions. *IBM Journal of Research and Development*, **17**(1973), 374-386.

[11] W.F. Dowling and J.H. Gallier. Linear time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, **3**(1984), 267-284.

[12] G. Gallo and M.G. Scutellà. Polynomially solvable satisfiability problems. *Information Processing Letters*, **29**(1988), 221-227.

[13] P.L. Hammer and A. Kogan. Optimal compression of propositional Horn knowledge bases: Complexity and approximation. *Artificial Intelligence*, **64**(1993), 131-145.

[14] P.L. Hammer and A. Kogan. Knowledge compression-logic minimization for expert systems. *Computers As Our Better Partners. Proceedings of the IISF/ACM Japan Int. Symp.* World Scientific, Singapore, 1994, 306-312.

[15] A. Itai and J.A. Makowsky. Unification as a complexity measure for logic programming. *Journal of Logic Programming*, **4**(1987), 105-117.

[16] D. Maier. Minimal covers in the relational database model. *Journal of the ACM*, **27**(1980), 664-674.

[17] M. Minoux. LTUR: A simplified linear time unit resolution algorithm for Horn formulae and computer implementation. *Information Processing Letters*, **29**(1988), 1-12.

[18] R. Swaminathan and D.K. Wagner. The arborescence-realization problem. *working paper University of Cincinnati* (1993).

[19] S. Yamasaki and S. Doshita. The satisfiability problem for a class consisting of Horn sentences and some non-Horn sentences in propositional logic. *Information and Control*, **59**(1983), 1-12.