The 2-SAT Problem of Regular Signed CNF Formulas*

Bernhard Beckert
Univ. of Karlsruhe
Institute for Logic, Complexity and Deduction Systems
D-76128 Karlsruhe, Germany
beckert@ira.uka.de

Reiner Hähnle
Chalmers Univ. of Technology
Dept. of Computing Science
S-41296 Gothenburg, Sweden
reiner@cs.chalmers.se

Felip Manyà†
Univ. de Lleida
Dept. of Computer Science
Jaume II, 69
E-25001 Lleida, Spain
felip@eup.udl.es

Abstract

Signed conjunctive normal form (signed CNF) is a classical conjunctive clause form using a generalized notion of literal, called signed atom*. A signed atom is an expression of the form $S : p$, where p is a classical atom and S, its* sign*, is a subset of a domain N. The informal meaning is "p takes one of the values in S".*

Applications for deduction in signed logics derive from those of annotated logic programming (e.g., mediated deductive databases), constraint programming (e.g., scheduling), and many-valued logics (e.g., natural language processing). The central rôle of signed CNF justifies a detailed study of its subclasses, including algorithms for and complexities of associated SAT problems.

Continuing our previous work [1], in this paper we present new results on the complexity of the signed 2-SAT problem; i.e., the case in which all clauses of a signed CNF formula have at most two literals.

1 Introduction

Signed formulas are a logical language for knowledge representation that lies in the intersection of the areas *constraint programming* (CP), *many-valued logic* (MVL), and *annotated logic programming* (ALP).

Signed conjunctive normal form (signed CNF) is a classical conjunctive clause form using a generalized notion of literal, called *signed atom*. A signed atom is an expression of the form $S : p$, where p is a classical atom and S, its *sign*, is a subset of a domain N (in this paper, we only consider the case where N is finite). The informal meaning of $S : p$ is "p takes one of the values in S".

When N is considered to be a truth value set, signed CNF formulas turn out to be a generic representation for finite-valued logics: The problem of deciding the satisfiability of formulas (SAT problem) of any finite-valued logic is in a natural way polynomially reducible to the problem of deciding satisfiability of formulas in signed CNF (signed SAT) [11].

An important and well investigated subclass of signed CNF formulas are *regular* CNF formulas (Def. 4): If N is equipped with a partial order $\leq$, a sign is regular if it is either of the form $\{j \in N \mid j \geq i\}$ or $\{j \in N \mid j \leq i\}$ for some $i \in N$. Regular CNF formulas are those signed CNF formulas where all occurring are regular or are complements of regular signs. See Section 2 for a formal definition of the logic of signed (and regular) CNF formulas.

Regular literals can be assigned polarities, which gives rise to a generalized notion of Horn clauses (Def. 5). The particular case where N is lattice-ordered and S is an order filter is investigated in annotated logic programming [14] (there, S is called an *annotation*); therefore, annotated logic programs can be considered to be particular signed logic formulas.

Moreover, $S : p$ can be interpreted as "p is constrained to the values in S" and, hence, as an instance of finite-domain constraint programming [13, 4].

Applications for deduction in signed logics derive from those of annotated logic programming (e.g., mediated deductive databases), constraint programming (e.g., scheduling), and many-valued logics (e.g., natural language processing). The central rôle of signed CNF and, in particular, regular CNF in automated deduction justifies a detailed study of its subclasses, including algorithms for and complexities of associated SAT problems; the interested reader may consult our recent survey [2]. Previously known complexity results for subclasses of signed SAT, including those from [1], are summarized in Section 4.

* This work has been partially supported by EC COST Action 15 *Many-valued Logics for Computer Science Applications*.

† Partially supported by the project TIC96-1038-C04-03 funded by the CICYT. This work was carried out during a visit to the University of Karlsruhe with a postdoctoral fellowship of the "Comissionat per a Universitats i Recerca" (1997BEAI400138).

In this paper we present new results on the complexity of the signed 2-SAT problem; i.e., the case in which all clauses of a signed CNF formula have at most two literals.

It was known before that the 2-SAT problem of signed CNF formulas is NP-complete and that the 2-SAT problem of regular CNF formulas is polynomially solvable in case the truth value set N is totally ordered [16, 1], but there were no results for the general regular 2-SAT problem.

In Section 5 we prove that:

- The 2-SAT problem for regular CNF formulas is NP-complete—even if the truth value set N is a complete, distributive or modular lattice.
- The 2-SAT problem for regular CNF formulas is polynomially solvable in case N is a lattice and all signs occurring in the formulas are regular (signs that are complements of regular signs but are not regular themselves are excluded).

2 Syntax and Semantics of Signed CNF Formulas

We assume that a signature, i.e., a denumerable set of propositional variables is given. To form signed atoms (literals), the propositional variables (atoms) are adorned with a sign that consists of a finite set of (truth) values.

Definition 1. A *truth value set* is a non-empty, finite set $N = \{i_1, i_2, \dots, i_n\}$ where $n \in \mathbb{N}$. The cardinality of N is denoted by $|N|$. A partial order $\leq$ is associated with N, which may be the empty order.

Definition 2. A *sign* is a set $S \subseteq N$ of truth values. A *signed atom* (a *literal*) is of the form $S : p$ where S is a sign and p is a propositional variable. The *complement* of a signed atom $S : p$, denoted by $\overline{S} : p$, is $(N \setminus S) : p$.

A *signed clause* is a finite set of signed atoms (literals). A signed clause containing exactly one literal is called a *signed unit clause*; and a signed clause containing exactly two literals is called a *signed binary clause*. The empty signed clause is denoted by $\square$.

A *signed CNF formula* is a finite set of signed clauses. A signed CNF formula whose clauses are binary is called a *signed 2-CNF formula*.

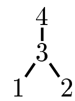
The clauses of a signed CNF formula are implicitly conjunctively connected; and the literals in a signed clause are implicitly disjunctively connected. In the following, we use $S_1 : p_1 \vee \cdots \vee S_k : p_k$ to represent a signed clause of the form $\{S_1 : p_1, \dots, S_k : p_k\}$.

Definition 3. The *size* of a signed clause C, denoted by $|C|$, is its cardinality. The size of a signed formula Γ, denoted by $|\Gamma|$, is the sum of the sizes of its signed clauses.

Definition 4. For each element i of the truth value set N, let $\uparrow i$ denote the sign $\{j \in N \mid j \geq i\}$ and let $\downarrow i$ denote the sign $\{j \in N \mid j \leq i\}$ where $\leq$ is the partial order associated with N. A sign S is *regular* if it is identical to $\uparrow i$ or to $\downarrow i$ for some $i \in N$.

A literal $S : p$ is a *regular literal* if (a) its sign S is regular or (b) its sign $S = \overline{S'}$ is the complement of a regular sign S'. A signed clause (a signed CNF formula) is a *regular clause* (a *regular CNF formula*) if all its literals are regular.

Example 1. Let the truth value set $N = \{1, 2, 3, 4\}$ be ordered as shown below, i.e., we use the standard order on natural numbers except that 1 and 2 are incomparable.



Then, the signs $\uparrow 1 = \{1, 3, 4\}$ and $\downarrow 1 = \{1\}$ are regular; and $\overline{\uparrow 1} = \{2\}$ and $\overline{\uparrow 3} = \{1, 2\}$ are complements of regular signs. The signs $\{3\}$ and $\{1, 4\}$ are neither regular nor complements of regular signs.

The complement $\overline{\uparrow 3}$ of the regular sign $\uparrow 3$ is *not* regular as it cannot be represented as $\uparrow i$ or $\downarrow i$ for any $i \in N$. Thus, a regular literal can have a sign that is not regular (but is the *complement* of a regular sign only).

Definition 5. A regular sign S is of *positive* (resp. *negative*) *polarity* if it can be represented as $\uparrow i$ (resp. $\downarrow i$) for some $i \in N$. A regular literal is of *positive* (*negative*) *polarity* if its sign is of positive (negative) polarity.

A regular clause is a *regular Horn clause* if it contains at most one literal of positive polarity and the signs of all its other literals are complements of signs with positive polarity. A regular CNF formula is a *regular Horn formula* if all its clauses are regular Horn clauses.

Example 2. Using the truth value set N and the associated ordering from the previous example, the clause $\uparrow 1 : p$, the clause $\uparrow 2 : p \vee \overline{\uparrow 3} : q$, and the clause $\uparrow 4 : q$ are Horn clauses. The regular clause $\uparrow 1 : p \vee \uparrow 2 : q$ is *not* a Horn clause as it contains more than one literal of positive polarity. As $\downarrow 1 = \overline{\uparrow 2}$ but $\downarrow 4 \neq \overline{\uparrow i}$ for all $i \in N$, the clause $\downarrow 1 : p$ is Horn whereas $\downarrow 4 : p$ is *not* Horn (both clauses are regular).

Definition 6. An *interpretation* is a mapping that assigns to every propositional variable an element of the truth value set. An interpretation I *satisfies* a literal $S : p$ iff $I(p) \in S$. It *satisfies* a signed clause C iff it satisfies at least one of the literals in C; and it *satisfies* a signed CNF formula Γ iff it satisfies all clauses in Γ.

A signed CNF formula (a signed clause) is *satisfiable* iff it is satisfied by at least one interpretation; otherwise it is *unsatisfiable*.

By definition, the empty signed clause is unsatisfiable and the empty signed CNF formula is satisfiable.

3 Satisfiability Problems of Signed Logics

Similar to the classical case, the SAT problem of signed CNF formulas (*signed* SAT) is NP-complete, but some of its sub-classes are polynomially solvable.

Satisfiability problems of signed CNF formulas have the truth value set N as a second input parameter (besides the formula Γ to be tested for satisfiability). Thus, *signed SAT* is the problem of deciding for an arbitrary signed formula Γ over an arbitrary truth value set N, whether there is an interpretation over N satisfying Γ. One also considers decision problems where N is not an input parameter but fixed, which is denoted by attaching the fixed truth value set N as an index to the name of the decision problem. For example, given a fixed truth value set N, *regular 2-SAT*$_N$ is the problem of deciding, for a regular CNF formula Γ over N such that each clause in Γ has at most two literals, whether there is an interpretation over N satisfying Γ.

4 Previously Known Complexity Results

In this section we summarize previously know results on the complexity of subclasses of signed SAT.

It is well-known that the classical SAT problem is NP-complete [5]. It is, however, polynomially solvable under certain restrictions. For example, there are linear-time algorithms for solving the classical SAT problem in case all clauses of the formula have at most one positive literal (classical Horn SAT) [6] and in case all clauses of the formula have at most two literals (classical 2-SAT) [8]. In recent years, complexity results for the regular Horn SAT and signed 2-SAT problems have been established.

If N is totally ordered, regular Horn SAT$_N$ can be solved in time linear in the size n of the formula, and regular Horn SAT can be solved in time linear in $n \log n$ [12, 15] (an algorithm for deciding satisfiability of a particular subclass of this type of regular Horn formulas appeared before [7]).

If N is a finite lattice, regular Horn SAT is solvable in time linear in the size of the formula and polynomial in the cardinality of N [1]. For distributive lattices, the more precise bound $n \cdot |N|^2$ was found independently [17].

Signed 2-SAT$_N$ for $|N| \geq 3$ and, thus, signed 2-SAT are NP-complete [15, 1]. However, signed 2-SAT is polynomially solvable in case (a) the CNF formula is monosigned, i.e., all occurring signs are singletons, or (b) the CNF formula is regular and the ordering of N is total [16, 15]. The complexity of regular 2-SAT$_N$ for non-total orderings was previously unknown; we present new results for that class of problems in Section 5.

As it is of relevance for the following section, we provide the original proof for the NP-completeness of signed 2-SAT [16, 15].

Signed SAT is obviously in NP (NP-easy), as a non-deterministic algorithm can guess a satisfying interpretation and check that it satisfies the formula in polynomial time. Therefore, all sub-problems of signed SAT that are considered in the following are NP-easy as well.

Theorem 7. *Signed 2-SAT*$_{\{1,2,3\}}$ *is NP-complete, even with the restriction that all occurring signs are of cardinality 2.*

Proof. The NP-easyness of signed 2-SAT$_{\{1,2,3\}}$ is obvious (see the above remark).

To prove NP-hardness, we shown that the *3-colorability problem*, which is known to be NP-complete [9], is polynomially reducible to signed 2-SAT$_{\{1,2,3\}}$.

The 3-colorability problem is to decide, for a given undirected graph $G = (V, E)$, whether there is a coloring of its nodes with three colors such that any two nodes connected by an edge are colored differently. Formally, the problem is to decide whether there is a function $c : V \to \{1, 2, 3\}$ such that $c(u) \neq c(v)$ for all edges $\langle u, v\rangle \in E$.

Given a graph $G = (V, E)$, we construct an instance Γ of signed 2-SAT$_{\{1,2,3\}}$ that is satisfiable if and only if G is 3-colorable. For each edge $\langle u, v\rangle \in E$, the formula Γ contains the following three binary signed clauses:

$$\{2,3\} : u \vee \{2,3\} : v$$
$$\{1,3\} : u \vee \{1,3\} : v$$
$$\{1,2\} : u \vee \{1,2\} : v$$

It is easy to check that these clauses are satisfied by an interpretation I if and only if $I(u) \neq I(v)$. Therefore, if the formula Γ is satisfied by an interpretation I, a 3-coloring c of G can be defined by setting $c(u) := I(u)$ for all $u \in V$; and, if c is a 3-coloring of G, then an interpretation I satisfying Γ can be defined by setting $I(u) := c(u)$ for all $u \in V$.

The formula Γ can obviously be computed from G in polynomial time. □

Corollary 8. *Signed 2-SAT and signed 2-SAT*$_N$ *for* $|N| \geq 3$ *are NP-complete.*

In the next section, Theorem 7 is used to prove that regular 2-SAT is NP-complete.

5 The Regular 2-SAT Problem

In this section, we first prove that regular 2-SAT is an NP-complete problem. NP-hardness is established in two different ways, but each time by a polynomial reduction from signed 2-SAT$_{\{1,2,3\}}$ to regular 2-SAT. By analyzing the proofs, we obtain NP-completeness of two different fragments of regular 2-SAT (Sections 5.1 and 5.2). On the other hand, we show that regular 2-SAT$_N$ is polynomially solvable if N is a lattice and only regular signs occur in literals, that is, all signs can be represented as $\uparrow i$ or $\downarrow i$ (Section 5.3).

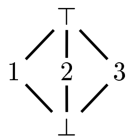
5.1 NP-completeness of Regular 2-SAT: First Version

Theorem 9. *Regular 2-SAT$_N$ is NP-complete, even if the truth value set N is a modular lattice and with the restriction that all occurring signs must be representable as complements of regular signs.*

Proof. NP-easyness of regular 2-SAT$_N$ is obvious (see the remark before Theorem 7).

For NP-hardness, by Theorem 7, it is sufficient to give a polynomial reduction of signed 2-CNF formulas Γ over $N = \{1, 2, 3\}$, where all signs in Γ have cardinality 2, to a regular 2-CNF formula Γ' over a truth value set N' fulfilling the restrictions expressed in the theorem, i.e., N' is a modular lattice and all signs occurring in Γ' are representable as complements of regular signs.

The idea is to complete the unordered truth value set N to the lattice N' shown below by adding a top element $\top$ above all $i \in N$ and a bottom element $\bot$ below all $i \in N$.



Now, each binary signed clause $C = S_1 : p_1 \vee S_2 : p_2$ in Γ, where $|S_1| = |S_2| = 2$, is expressed by the regular 2-CNF clause C' in Γ':

$$(C') \qquad \overline{\uparrow (N \setminus S_1)} : p_1 \vee \overline{\uparrow (N \setminus S_2)} : p_2 \ ,$$

where we use the convention $\uparrow \{i\} = \uparrow i$ and recall that $|S_i| = 2$.

It is easy to check that C' contains exactly the same literals as C up to $\bot \in N'$, which occurs additionally in its signs. Therefore, we must add clauses to Γ' expressing that $\bot$ cannot be taken on by satisfying interpretations. This is achieved by adding to Γ' the unit clause C'_p for each propositional variable p occurring in Γ:

$$(C'_p) \qquad \overline{\downarrow \bot} : p$$

Obviously, Γ' has the required form, and Γ is satisfiable (over N) iff Γ' is satisfiable (over N'). Note that $\top$ does not occur in Γ' as merely a semi-lattice is needed for the construction. □

Corollary 10. *Regular 2-SAT is NP-complete.*

5.2 NP-completeness of Regular 2-SAT: Second Version

In this section, we use a similar construction as in Theorem 9 to prove NP-hardness of the regular 2-SAT fragment where all signs are regular signs of positive polarity or complements of regular signs of positive polarity, that is, signs can be represented as $\uparrow i$ or $\overline{\uparrow i}$.

This subproblem of regular 2-SAT is considered separately for two reasons:

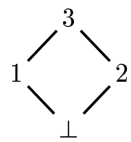
- If N is a lattice, this particular class of regular CNF formulas gives rise to clausal annotated logics [14];
- This kind of regular signs are those used to generalize the concept of Horn formulas (Def. 5). As already mentioned, the SAT problem of regular Horn formulas is polynomially solvable in case N is a lattice [1]. Here, we show that, in contrast to the Horn fragment, the 2-SAT fragment of that particular subclass of regular CNF formulas is NP-complete.

Theorem 11. *Regular 2-SAT$_N$ is NP-complete, even if the truth value set N is a distributive lattice and with the restriction that all occurring signs must be representable as regular signs of positive polarity or complements of regular signs of positive polarity.*

Proof. As before, NP-easyness is straightforward.

For NP-hardness, we use again a polynomial reduction of signed 2-CNF formulas Γ over $N = \{1, 2, 3\}$, where all signs in Γ have cardinality 2, to a regular 2-CNF formula Γ'' over a truth value set N'' fulfilling the restrictions expressed in the theorem, i.e., N'' is a distributive lattice and all signs in Γ'' can be represented as $\uparrow i$ or $\overline{\uparrow i}$.

This time, the idea is to designate an arbitrary element $i \in N$ as the top element and then complete N to a lattice N'' (shown below) with a bottom element $\bot$.



This construction avoids the need for signs that can only be represented as $\downarrow i$, as will shortly be seen.

Each binary signed clause $C = S_1 : p_1 \vee S_2 : p_2$ in Γ, where $|S_1| = |S_2| = 2$, is expressed by the regular 2-CNF clause C'' in Γ'':

$$(C'') \begin{cases} \overline{\uparrow 3} : p_1 \vee \overline{\uparrow 3} : p_2 & \text{if } S_1 = S_2 = \{1, 2\} \\ \overline{\uparrow 3} : p_1 \vee \uparrow (S_2 \setminus \{3\}) : p_2 & \text{if } S_1 = \{1, 2\} \neq S_2 \\ \overline{\uparrow 3} : p_2 \vee \uparrow (S_1 \setminus \{3\}) : p_1 & \text{if } S_2 = \{1, 2\} \neq S_1 \\ \uparrow (S_1 \setminus \{3\}) : p_1 \vee \uparrow (S_2 \setminus \{3\}) : p_2 & \text{otherwise} \end{cases}$$

again using the convention $\uparrow \{i\} = \uparrow i$ and recalling that $|S_i| = 2$.

As before, C'' contains exactly the same literals as C up to $\bot \in N''$, which occurs additionally in its signs. Therefore, we must add clauses to Γ'' which express that $\bot$ cannot be taken on by satisfying interpretations. This is clearly achieved by adding to Γ'' the clause C''_p for each propositional variable p occurring in Γ:

$$(C''_p) \qquad \uparrow 1 : p \vee \uparrow 2 : p$$

It is trivial to check that Γ'' has the required form and that Γ is satisfiable (over N) iff Γ'' is satisfiable (over N''). □

An analog result can be established for the case where all the signs are regular signs of negative polarity or complements of regular signs of negative polarity.

5.3 Regular 2-SAT with Signs of the Form $\uparrow i$ and $\downarrow i$

In this section, we prove that regular 2-SAT$_N$ is polynomially solvable if N is a lattice and only regular signs occur, that is, all signs can be represented as $\uparrow i$ or $\downarrow i$.

To do so, we first prove that the resolution rule (1) below is refutation complete for lattice-based regular CNF formulas that contain only regular signs. Then we prove that, whenever such a formula contains at most two literals per clause, the number of its possible resolvents is polynomial in the number of distinct regular literals occurring in the formula.

$$\frac{\uparrow i : p \vee D_1 \quad \downarrow j : p \vee D_2}{D_1 \vee D_2} \quad \text{if } j \not\geq i \tag{1}$$

regular binary resolution

Lemma 12. *Let N be a lattice. Every unsatisfiable set Γ of regular unit clauses whose signs are all regular contains clauses $\uparrow i : p$ and $\downarrow j : p$ such that $j \not\geq i$.*

Proof. By way of contradiction, assume that for all propositional variables p, for all $\uparrow i_k : p \in \Gamma$ $(1 \leq k \leq r)$ and all $\downarrow j_l : p \in \Gamma$ $(1 \leq l \leq m)$, it is the case that $j_l \geq i_k$.

Since N is a lattice, for each j_l $(1 \leq l \leq m)$, we have that $j_l \geq i^*(p) = (i_1 \sqcup \cdots \sqcup i_r)$; and, therefore, the interpretation I defined as $I(p) = i^*(p)$ for every propositional variable p satisfies Γ. This contradicts the unsatisfiability of Γ, so Γ must contain two regular unit clauses $\uparrow i : p$ and $\downarrow j : p$ such that $j \not\geq i$. □

Soundness of rule (1) is straightforward, we only prove completeness.

Theorem 13. *Regular binary resolution is refutation complete for regular CNF formulas Γ over N if N is a lattice and only regular signs occur in Γ.*

Proof. Let $\Gamma = \{C_1, C_2, \ldots, C_m\}$ be an unsatisfiable regular CNF formula as stated. Let r be the number of excess literals in Γ, that is $r = (\Sigma_{i=1}^m |C_i|) - |\Gamma|$, and proceed by induction on r.

Without loss of generality, we assume that $\Box \notin \Gamma$; hence, if $r = 0$, then every clause is a unit clause. By Lemma 12, there are clauses $\uparrow i : p$ and $\downarrow j : p$ in Γ such that $j \not\geq i$. Since they are unit clauses, their regular binary resolvent is the empty clause.

Suppose now that there is a regular binary resolution refutation of every unsatisfiable regular CNF formula with at most r excess literals, and assume that Γ has $r+1$ excess literals.

At least one clause in Γ, say C, contains at least two literals. Let C' be the result of removing one literal, say L, from C, and let Γ' be the result of replacing C in Γ with C'.

Since any satisfying interpretation for Γ' would satisfy Γ as well, Γ' must be unsatisfiable. In addition, Γ' has only r excess literals, so there exists a regular binary resolution derivation of the empty clause from Γ'. Thus, if the same derivation is used on Γ, it will either end with the empty clause or with the unit clause containing L.

We now have a regular binary resolution derivation of the unsatisfiable regular CNF formula $\Gamma'' = (\Gamma - \{C\}) \cup \{L\}$ from Γ. The formula Γ'' is unsatisfiable because every satisfying interpretation would satisfy Γ as well (since $L \in C$). As Γ'' has fewer excess literals than Γ, there is a regular binary resolution derivation of the empty clause from Γ''; the two derivations together provide a refutation of Γ. □

Theorem 14. *Regular 2-SAT is polynomially solvable when restricted to instances where the truth value set is a lattice and all occurring signs are regular.*

Proof. Using rule (1), regular binary resolvents of clauses of a regular 2-SAT instance have at most two literals. Therefore, the number of different resolvents that can be derived is polynomial in the number of distinct regular literals occurring in the formula (rule (1) does not introduce any new literals). Since rule (1) is refutation complete for regular 2-SAT instances whose signs are regular (Theorem 13), it suffices to generate all (polynomially many) possible resolvents and check whether the empty clause is among them. □

6 Conclusion & Future Work

In this paper we shed some new light on the complexity of the 2-SAT problem of regular CNF formulas. When aiming at implementation and application of signed logic [10, 3, 15], it is important to identify which fragments are easy and which are hard (in the sense of P versus NP). Our main findings were: in contrast to classical logic, N-valued regular 2-SAT becomes NP-complete relatively soon—if N is restricted to finite modular lattices and only complements of regular signs may occur in the input (Theorem 9) or if N is restricted to finite distributive lattices and only regular signs of uniform polarity and their complements may occur in the input (Theorem 11). Interestingly, the latter regular CNF fragment has polynomial complexity under the Horn restriction [1]. On the other hand, the regular 2-SAT problem is polynomially solvable if N is a finite lattice and only regular signs occur in the input (Theorem 13). Further polynomial results were obtained elsewhere for the totally ordered case [16]. Our findings indicate that the complexity of the 2-SAT problem of regular CNF formulas depends crucially on the form of the signs occurring in the input, once the ordering of N is non-linear. Specifically, any negative information (that is: complemented signs) make the problem hard.

In the future, one might look at the following questions: In the proof of Theorem 9, a modular but non-distributive lattice was used—is it possible to prove the theorem with the help of a distributive lattice? It would be good to have more polynomial results like Theorem 13. We also would like to generalize our findings to the infinite-valued case as much as possible. Finally, our theoretical findings need to be implemented and evaluated in decision procedures for signed logic [3].

References

[1] B. Beckert, R. Hähnle, and F. Manyà. Transformations between signed and classical clause logic. In *Proc. 29th Int. Symposium on Multiple-Valued Logics, Freiburg, Germany*, pages 248–255. IEEE Computer Society Press, Los Alamitos, 1999.

[2] B. Beckert, R. Hähnle, and F. Manyá. The SAT problem of signed CNF formulas. In D. Basin, M. D'Agostino, D. Gabbay, S. Matthews, and L. Viganò, editors, *Labelled Deduction*, volume 17 of *Applied Logic Series*, pages 61–82. Kluwer, Dordrecht, May 2000.

[3] R. Béjar and F. Manyà. Phase transitions in the regular random 3-SAT problem. In Z. W. Raś and A. Skowron, editors, *Proc. Int. Symposium on Methodologies for Intelligent Systems, Warsaw, Poland*, number 1609 in LNCS, pages 292–300. Springer-Verlag, 1999.

[4] T. Castell and H. Fargier. Between SAT and CSP: Propositional satisfaction problems and clausal CSPs. In H. Prade, editor, *Proc. European Conference on Artificial Intelligence, Brighton, UK*, pages 214–218. John Wiley & Sons, 1998.

[5] S. Cook. The complexity of theorem-proving procedures. In *Proc. Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[6] W. Dowling and J. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulæ. *J. of Logic Programming*, 3:267–284, 1984.

[7] G. Escalada-Imaz and F. Manyà. The satisfiability problem for multiple-valued Horn formulæ. In *Proc. Int. Symposium on Multiple-Valued Logics, Boston, USA*, pages 250–256. IEEE Computer Society Press, Los Alamitos, 1994.

[8] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. of Computing*, 5:691–703, 1976.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.

[10] R. Haenni and N. Lehmann. Assumption-based reasoning with finite set constraints. In *Proc. Int. Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, France*, pages 1289–1295, 1998.

[11] R. Hähnle. Short conjunctive normal forms in finitely-valued logics. *J. of Logic and Computation*, 4(6):905–927, 1994.

[12] R. Hähnle. Exploiting data dependencies in many-valued logics. *J. of Applied Non-Classical Logics*, 6:49–69, 1996.

[13] R. Hähnle, R. Hasegawa, and Y. Shirai. Model generation theorem proving with interval constraints. In F. Benhamou, W. J. Older, M. van Emden, and P. van Hentenryck, editors, *Proc. ILPS Post-Conference Workshop on Interval Constraints, Portland, USA*, 1995.

[14] M. Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *J. of Logic Programming*, 12:335–367, 1992.

[15] F. Manyà. *Proof Procedures for Multiple-Valued Propositional Logics*, volume 9 of *Monografies de l'Institut d'Investigació en Intel·ligència Artificial*. IIIA-CSIC, Bellaterra (Barcelona), 1999.

[16] F. Manyà. The 2-SAT problem in signed CNF formulas. *Multiple-Valued Logic, An International Journal*, 2000. To appear.

[17] V. Sofronie-Stokkermans. On translation of finitely-valued logics to classical first-order logic. In H. Prade, editor, *Proc. European Conference on Artificial Intelligence, Brighton, UK*, pages 410–411. John Wiley & Sons, 1998.