Optimal Satisfiability for Propositional Calculi and Constraint Satisfaction Problems

Steffen Reith *and* *Heribert Vollmer*

Lehrstuhl für Theoretische Informatik
Universität Würzburg
Am Hubland
D-97074 Würzburg, Germany
[streit,vollmer]@informatik.uni-wuerzburg.de

Abstract. We consider the problems of finding the lexicographically minimal (or maximal) satisfying assignment of propositional formulas for different restricted classes of formulas. It turns out that for each class from our framework, these problems are either polynomial time solvable or complete for OptP. We also consider the problem of deciding if in the optimal assignment the largest variable gets value 1. We show that this problem is either in P or P^{NP} complete.

1 Introduction

In 1978 Thomas J. Schaefer proved a remarkable result. He examined satisfiability of propositional formulas for certain syntactically restricted formula classes. Each such class is given by a set S of Boolean functions allowed when constructing formulas. An *S-formula* in his sense is a conjunction of clauses, where each clause consists of a Boolean function from S applied to some propositional variables. Such a Boolean function can be interpreted as a constraint that has to be fulfilled by a given assignment; the satisfiability problem for S-formulas hence provides a mathematical model for the examination of the complexity of constraint satisfaction problems. Let $\mathbf{CSP}(S)$ denote the problem to decide for a given S-formula if it is satisfiable. Schaefer showed that, depending on S, the problem $\mathbf{CSP}(S)$ is either (1) efficiently (i.e., in polynomial time) solvable or (2) NP-complete; and he gave a simple criterion that, given S, allows one to determine whether (1) or (2) holds. Since the complexity of $\mathbf{CSP}(S)$ is either easy or hard (and not located in one of the – under the assumption $\mathrm{P} \neq \mathrm{NP}$ – infinitely many intermediate degrees between P and the NP-complete sets [Lad75]), Schaefer called this a "dichotomy theorem for satisfiability".

A somewhat more general kind of formulas was investigated by Lewis in 1979 (see [Lew79]). Here we are allowed to build propositional formulas using connectives taken from a finite set B of Boolean functions, instead of the usual connectives $\wedge, \vee, \neg, \ldots$. These formulas will be called B-formulas. (To distinguish, S-formulas in Schaefer's sense will henceforth be refered to as S-CSPs.) Classes of B-formulas are very closely related to closed classes of Boolean functions, which were fully characterized by Post in the twenties of this century (see [Pos41,JGK70]). Similar to Schaefer, Lewis obtained a dichotomy theorem which states that the satisfiability problem $\mathbf{SAT}(B)$ of such formulas either complete for NP or in P, depending on properties of the set B of allowed functions.

In the last few years, these results regained interest among complexity theorists. Constraint satisfaction problems were studied by Nadia Creignou and others [Cre95,CH96,CH97], see also the monograph [CKS00]. The complexity of problems related to B-circuits and B-formulas was studied in [RW00]. Considering different versions of satisfiability, optimization and counting problems, dichotomy theorems for classes as NP, MaxSNP, and #P were obtained. Also, the study of

Schaefer's formulas lead to remarkable results about approximability of optimization problems in the constraint satisfaction context [KST97,KSW97].

In this paper, we continue this line of research by considering the complexity of other optimization problems defined via formula satisfiability, namely the problems LexMax**SAT** and LexMin**SAT** of determining the lexicographically maximal (or minimal) satisfying assignment of a given propositional formula. In the case of unrestricted formulas, these problems are known to be complete for Krentel's class OptP [Kre88]. The main result of the present paper is a clarification of the complexity of the problem to determine maximal or minimal satisfying assignments of formulas given either in the Lewis context by a set of Boolean connectives, or in the Schaefer context, by a set of constraints. We will show that in all cases, the considered problem is either complete for OptP or solvable in polynomial time, depending on the set of allowed connectives or constraints.

We also examine the problem to determine if in the minimal or maximal satisfying assignment, the first variable is set to true. These problems are known as OddLexMin**SAT** and OddLexMax**SAT**, and were shown to be P^{NP}-complete for unrestricted propositional formulas by Wagner [Wag87]. Considering formulas given by a set of Boolean connectives or constraints, we obtain a dichotomy theorem for OddLexMin**SAT** and OddLexMax**SAT**, showing that the problem is either complete for P^{NP} or polynomial-time solvable.

After presenting some notations and preliminary results in Sect. 2, we turn to B-formulas in the Post/Lewis-context in Sect. 3. In Sect. 4 we consider constraint satisfaction problems. Finally, Sect. 5 concludes.

2 Preliminaries

Propositional Logic

Let Φ be a propositional formula. By $\mathrm{Var}(\Phi)$ we denote the set of those variables associated with Φ. Hence, $\mathrm{Var}(\Phi)$ must contain those variables with actual occurrences in Φ, but we also allow $\mathrm{Var}(\Phi)$ to contain additionally so called *fictive* variables. We say that Φ is a *formula over* $\mathrm{Var}(\Phi)$. To denote that Φ is a formula over $\{x_1, \ldots, x_k\}$, we also write $\Phi(x_1, \ldots, x_k)$. By $\Phi\left[\begin{smallmatrix}x\\y\end{smallmatrix}\right]$ we indicate the formula created by simultaneously replacing each occurrence of x in Φ by y, where x, y are either variables or constants.

Let V be a finite set of propositional variables. Since we want to compare propositional truth assignments lexicographically, we have to talk about the first, second, etc. variable. Thus we will always assume an ordering on V without further mention.

An *assignment with respect to* V is a function $I\colon V \to \{0,1\}$. When the set V of variables is clear from the context, we will simply speak of an *assignment*. In order for an assignment w.r.t. V to be compatible with a formula Φ, we must have $\mathrm{Var}(\Phi) = V$. That an assignment I satisfies Φ will be denoted by $I \models \Phi$. If I is an assignment w.r.t. V, $y \notin V$, and $a \in \{0,1\}$, then $I \cup \{y := a\}$ denotes the assignment I' w.r.t. $V \cup \{y\}$, defined by $I'(y) = a$ and $I'(x) = I(x)$ for all $x \neq y$. On the other hand, for $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\} \subseteq V$, we denote by $I' = I/\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ the assignment I' w.r.t. $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$, given by $I'(x) = I(x)$ iff $x \in \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$.

If $V = \{x_1, \ldots, x_k\}$, $x_1 < \cdots < x_k$, then an assignment I with $I(x_i) = a_i$ will also be denoted by $(a_1, \ldots, a_k)$. An ordering on the variables induces an ordering on assignments as follows: $(a_1, \ldots, a_k) < (b_1, \ldots, b_k)$ if and only if there is an $i \leq k$ such that for all $j < i$ we have $a_j = b_j$ and $a_i < b_i$. We refer to this ordering as the *lexicographical ordering*. We write $(a_1, \ldots, a_k) \models_{\min} \Phi$ ($(a_1, \ldots, a_k) \models_{\max} \Phi$, resp.) iff $(a_1, \ldots, a_k) \models \Phi$ and there exists no lexicographically smaller (larger, resp.) $(a'_1, \ldots, a'_k) \in \{0,1\}^k$ such that $(a'_1, \ldots, a'_k) \models \Phi$.

Maximization and Minimization Problems

The study of optimization problems in computational complexity theory started with the work of Krentel [Kre88,Kre92]. He defined the class OptP and an oracle hierarchy built on this class using so called metric Turing machines. We do not need this machine model here; therefore we proceed by defining the class OptP and its subclasses MinP and MaxP (explicitly introduced the first time in [Köb89]) using a characterization given in [VW95].

We fix the alphabet $\Sigma = \{0,1\}$. Let FP denote the class of all functions $f\colon \Sigma^* \to \Sigma^*$ computable deterministically in polynomial time. Say that a function h belongs to the class MinP if there is a function $f \in \mathrm{FP}$ and a polynomial p such that, for all x, $h(x) = \min_{|y|=p(|x|)} f(x,y)$, where minimization is taken with respect to lexicographical order. The class MaxP is defined by taking the maximum of these values. Finally, let $\mathrm{OptP} = \mathrm{MinP} \cup \mathrm{MaxP}$.

Krentel considered the following reducibility in connection with these classes: A function f is *metric reducible* to h ($f \leq^p_{\mathrm{met}} h$) if there exist two functions $g_1, g_2 \in FP$ such that for all x we have: $f(x) = g_1(h(g_2(x)), x)$. We say that g_1, g_2 establish the metric reduction from f to h. When we claim a metric reducibility below, we will always witness this by constructing suitable functions g_1, g_2.

OptP is a subclass of $\mathrm{FP}^{\mathrm{NP}}$, and it is well known that the closure of all three classes MinP, MaxP, and OptP under metric reductions coincides with the class $\mathrm{FP}^{\mathrm{NP}}$; which means that showing completeness of a problem for MinP generally implies hardness of the same problem for MaxP and completeness for OptP, see [Kre88,VW95,Vol94].

Krentel in [Kre88] presented a number of problems complete for OptP under metric reducibility. The for us most important one is the problem of finding the lexicographically minimal satisfying assignment of a given formula, defined as follows:

Problem:	LexMin3-**SAT**
Instance:	a propositional formula Φ in 3-CNF
Output:	the lexicographically smallest satisfying assignment of Φ, or "$\perp$" if Φ is unsatisfiable

The problem LexMax3-**SAT** is defined analogously.

Proposition 1 ([Kre88]). LexMin3-**SAT** *and* LexMax3-**SAT** *are complete for* OptP *under metric reductions.*

3 Dichotomy Theorems for B-formulas

3.1 Formulas Given by a Set of Boolean Functions

Any function of the kind $f\colon \{0,1\}^k \to \{0,1\}$ will be called *(k-ary) Boolean function.* By BF we denote the set of all Boolean functions. For simplicity we often use propositional formulas to represent Boolean functions, formally: Let Φ be a formula over a set $V = \mathrm{Var}(\Phi)$ of k variables. Then f_Φ, the Boolean function defined (or, *represented*) by Φ, is given by $f_\Phi(a_1,\ldots,a_k) = 1$ iff $(a_1,\ldots,a_k) \models \Phi$. For example, the functions $\mathrm{id}(x)$, $\mathrm{et}(x,y)$, $\mathrm{vel}(x,y)$, $\mathrm{non}(x)$, $\mathrm{aut}(x,y)$ are represented by the formulas x, $x \wedge y$, $x \vee y$, $\neg x$ and $x \oplus y$. We will use 0 and 1 for the constant 0-ary boolean functions. The variable x_i is called *fictive* in f if $f(a_1,\ldots,a_{i-1},0,a_{i+1},\ldots,a_n) = f(a_1,\ldots,a_{i-1},1,a_{i+1},\ldots a_n)$ for all $a_1,\ldots a_{i-1},a_{i+1},\ldots,a_n$ and $1 \leq i \leq n$.

Let B be a set of Boolean functions. By $[B]$ we denote the smallest set of Boolean functions, which contains B and is closed under superposition, i.e., under substitution (that is, composition of functions), permutation and identification of variables, and introduction of fictive variables. A set F of Boolean functions is a *base* for B if $[F] = B$, and F is called *closed* if $[F] = F$. A base B is called *complete* if $[B] = \mathrm{BF}$. As an example, it is well known that $\{\mathrm{vel}, \mathrm{et}, \mathrm{non}\}$ and $\{\neg(x \wedge y)\}$ are complete bases.

The following closed sets of functions turn out to be particularly interesting. Let f be an n-ary boolean function. Following [Pos41], f is said to be

- *a-reproducing*, if $f(a, a, \ldots, a) = a$, where $a \in \{0, 1\}$;
- *linear*, if there exists $a_0, a_1, \ldots, a_n \in \{0, 1\}$ such that $f(b_1, \ldots, b_n) = a_0 \oplus (a_1 \wedge b_1) \oplus \ldots \oplus (a_n \wedge b_n)$, for all $b_1, \ldots, b_n \in \{0, 1\}$;
- *self-dual*, if $f(a_1, \ldots, a_n) = \overline{f(\overline{a}_1, \ldots, \overline{a}_n)}$ for all $a_1, \ldots, a_n \in \{0, 1\}$;
- *monotonic*, if $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$ for all $a_1, \ldots, a_n, b_1, \ldots, b_n \in \{0, 1\}$ such that $a_1 \leq b_1, a_2 \leq b_2, \ldots, a_n \leq b_n$;
- *a-separating*, if there exists an $i \in \{1, 2, \ldots, n\}$ such that $f^{-1}(a) \subseteq \{0, 1\}^{i-1} \times \{a\} \times \{0, 1\}^{n-i}$ ($a \in \{0, 1\}$);
- *a-separating of degree m* if, for every $U \subseteq f^{-1}(a)$ such that $|U| = m$, there exists an $i \in \{1, 2, \ldots, n\}$ such that $U \subseteq \{0, 1\}^{i-1} \times \{a\} \times \{0, 1\}^{n-i}$ ($a \in \{0, 1\}$, $m \geq 2$).

The class of Boolean functions that are 0-reproducing (1-reproducing, linear, self-dual, monotone, 0-separating, 1-separating, 0-separating of degree m, 1-separating of degree m, resp.) is denoted by R_0 (R_1, L, D, M, S_0, S_1, S_0^m, S_1^m, resp.).

Emil Post in [Pos41] gave a complete list of all classes of Boolean functions being closed under superposition. Fig. 1 represents the inclusions among these. Besides those classes introduced above, we find other classes in the diagram, which are always obtained as the intersection of their super-classes in the diagram, e.g., R is the intersection of the classes R_1 and R_0. The well known five Post-classes, i.e., the maximal non-complete closed sets of Boolean functions, are marked by bold circles.

Moreover Post showed that for each closed class there exists a finite base. A complete list can be found in [Pos41,JGK70]. Important for us are the following cases:

Proposition 2 ([Pos41,JGK70]). *Every closed class of Boolean functions has a finite base. In particular:*

1. $\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\}$ *is a base of* BF.
2. $\{\mathrm{et}, \mathrm{aut}\}$ *is a base of* R_0.
3. $\{\mathrm{vel}, x \oplus y \oplus 1\}$ *is a base of* R_1.
4. $\{\mathrm{et}, \mathrm{vel}, 0, 1\}$ *is a base of* M.
5. $\{(x \wedge \overline{y}) \vee (x \wedge \overline{z}) \vee (\overline{y} \wedge \overline{z})\}$ *is a base of* D.
6. $\{(x \wedge y) \vee (x \wedge \overline{z}) \vee (y \wedge \overline{z})\}$ *is a base of* D_1.
7. $\{\mathrm{aut}, 1\}$ *is a base of* L.
8. $\{x \vee \overline{y}\}$ *is a base of* S_0.
9. $\{x \wedge \overline{y}\}$ *is a base of* S_1.
10. $\{x \vee (y \wedge \overline{z})\}$ *is a base of* S_{02}.
11. $\{x \wedge (y \vee \overline{z})\}$ *is a base of* S_{12}.
12. $\{x \vee (y \wedge z)\}$ *is a base of* S_{00}.
13. $\{x \wedge (y \vee z)\}$ *is a base of* S_{10}.
14. $\{\mathrm{et}, 0, 1\}$ *is a base of* E.
15. $\{\mathrm{vel}, 0, 1\}$ *is a base of* V.

Now let B be a finite set of Boolean functions. A *B-formula* Φ is a formula representing a Boolean function $f_\Phi \in [B]$ by describing the substitution structure of f_Φ, where we use fixed symbols for the Boolean functions of B. More precisely, let V be a finite set of variables.

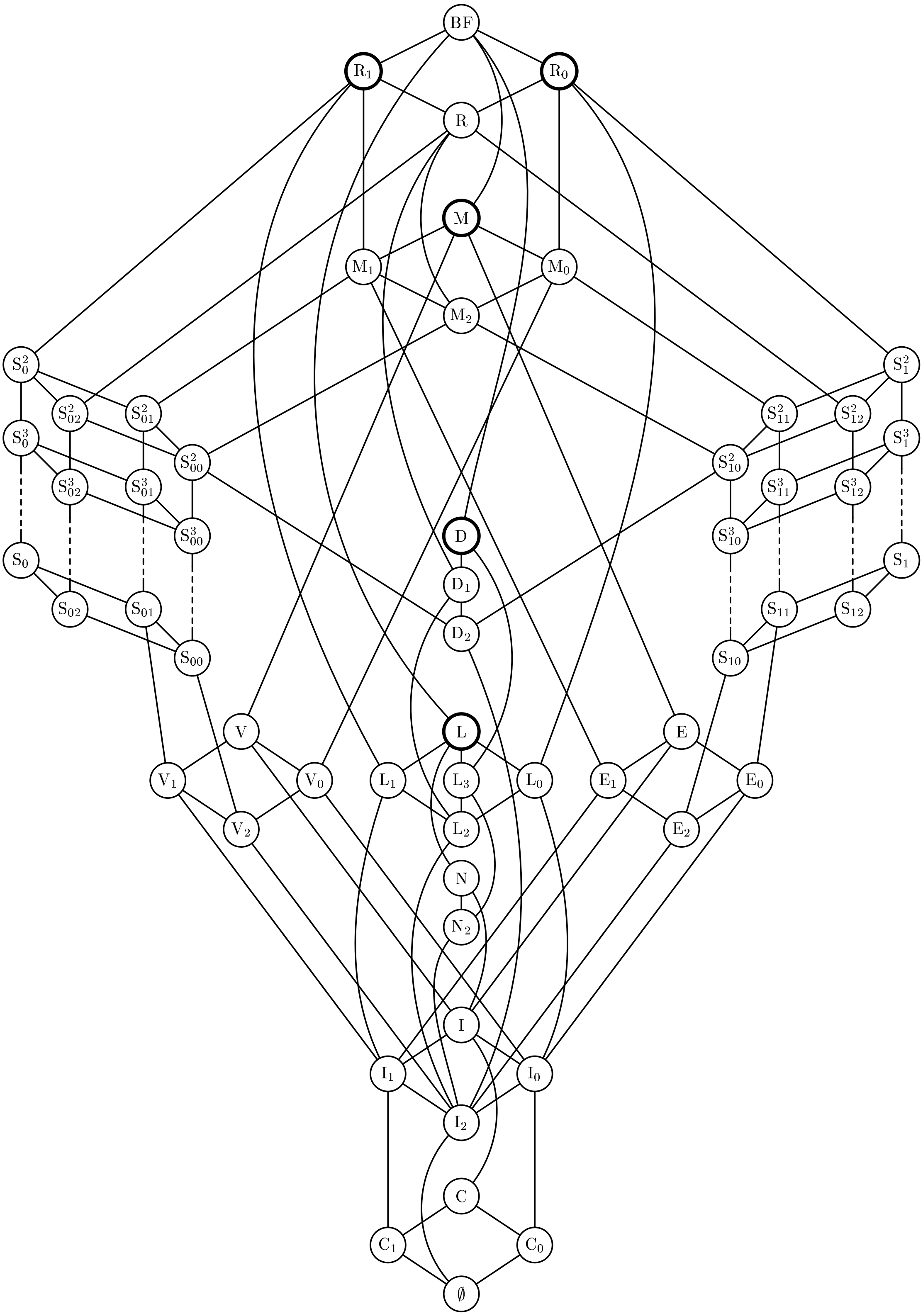


Fig. 1. Graph of all closed classes of Boolean functions

Then we define:

– every variable $x \in V$ is a B-formula over V;
– if $f \in B$ is k-ary, and $\varPhi_1, \dots \varPhi_k$ are B-formulas over V, then $\tilde{f}(\varPhi_1, \dots \varPhi_k)$ is a B-formula over V, where $\tilde{f}$ is a symbol representing f.

Note that not all variables in V actually have to occur in $\varPhi$; those not appearing are fictive variables.

In the following, we will refer to the satisfiability problem for B-formulas as $\mathbf{SAT}(B)$. The following result is known:

Proposition 3 ([Lew79]). *If* $[B] \supseteq \mathrm{S}_1$ *then* $\mathbf{SAT}(B)$ *is* NP*-complete, in all other cases,* $\mathbf{SAT}(B) \in \mathrm{P}$*.*

3.2 Dichotomy theorems for LexMinSAT(B) and LexMaxSAT(B)

In this section we study the complexity of finding the lexicographical smallest (largest, resp.) satisfying assignment of B-formulas, for all finite sets B of Boolean functions; formally:

PROBLEM:	LexMin$\mathbf{SAT}(B)$
INSTANCE:	a B-formula $\varPhi$
OUTPUT:	the lexicographically smallest satisfying assignment of $\varPhi$, or "$\perp$" if $\varPhi$ is unsatisfiable

The corresponding maximization problem is denoted by LexMax$\mathbf{SAT}(B)$.

The cases of formulas with easy LexMin/Max$\mathbf{SAT}$-problem are easy to identify:

Lemma 4. *If B is a finite set of Boolean functions such that $B \subseteq \mathrm{R}_1$, then* LexMax$\mathbf{SAT}(B) \in$ FP*.*

Proof. If $\varPhi(x_1, \dots, x_n)$ is a B-formula, then $f_\varPhi(1^n) = 1$. □

Lemma 5. *If B is a finite set of Boolean functions such that $B \subseteq \mathrm{M}$ or $B \subseteq \mathrm{L}$, then* LexMin$\mathbf{SAT}(B) \in$ FP *and* LexMax$\mathbf{SAT}(B) \in$ FP*.*

Proof. The following is a slight modification of an algorithm presented in [CH97]:

In both cases $\mathbf{SAT}(B) \in \mathrm{P}$. Hence, given $\varPhi$, a minimal satisfying assignment can be computed as follows: Consider all variables x in $\varPhi$ in their order, first set x to false and if the resulting formula is satisfiable then proceed with $x :=$ false to the next variable. If the resulting formula is not satisfiable, then we set x to true and test satisfiability. If now the result is satisfiable, we proceed (with $x :=$ true) to the next variable, otherwise we output $\perp$. □

We will now show that in all other cases, computing minimal or maximal assignments is hard, i.e., complete for OptP. Clearly, for all B, LexMin$\mathbf{SAT}(B) \in$ MinP (LexMax$\mathbf{SAT}(B) \in$ MaxP, resp.). In the subsequent theorems we hence have to prove hardness via suitable reductions.

The following easy lemma shows that we can construct, for any 3-CNF-formula, a logically equivalent $(B \cup \{0,1\})$-formula, if B together with the constants 0 and 1 forms a complete base.

Lemma 6. *Let B be a finite set of Boolean functions such that there are B-formulas $E(x, y, v, u)$ and $N(x, v, u)$ with $E(x, y, 0, 1) \equiv x \wedge y$ and $N(x, 0, 1) \equiv \neg x$. Then, for any 3-CNF formula $\varPhi(x_1, \dots, x_n)$, there exists a B-formula $\varPsi(x_1, \dots, x_n, u, v)$ such that $\varPsi(x_1, \dots, x_n, 0, 1) \equiv \varPhi(x_1, \dots, x_n)$. Moreover, $\varPsi$ can be computed from $\varPhi$ in logarithmic space.*

Proof. Since $\{\wedge, \neg\}$ forms a complete basis, we can trivially transform every formula in a $(B \cup \{0,1\})$-formula. In the case of a given CNF-formula with m clauses, to avoid an exponential blowup while replacing the m $\wedge$'s, we insert parentheses in such a way that we get a tree of depth $\log m$, and then replace each $\wedge$ by an appropriate $(B \cup \{0,1\})$-formula. Finally, the $\vee$'s inside the clauses are replaced by corresponding B-formulas. □

For some bases B, the constants 0 and 1 needed in the preceding lemma are easy to construct. This is the contents of the following theorem.

Theorem 7. *If B is a finite set of Boolean functions such that $[B] \supseteq \mathrm{S}_1$, then* LexMin**SAT**$(B)$ *(*LexMax**SAT**(B)*, resp.) is $\leq^p_{\mathrm{met}}$-complete for* MinP *(*MaxP*, resp.).*

Proof. From Proposition 2 we know that the function $g(x,y) = x \wedge \overline{y}$ is a base for S_1. It is obvious that $g(x, g(x,y)) = x \wedge y$ and $g(1,x) = \neg x$. Since $[B] \supseteq \mathrm{S}_1$ there exist B-formulas $G(x,y) \equiv g(x,y)$, $E(x,y,v,u) \equiv x \wedge y$ and $N(x,u,1) \equiv \neg x$. Note that the variables u and v are not used in E and that u is not used in N.

Let $\Phi(x_1,\dots,x_n)$ be a 3-CNF formula. We use Lemma 6 to obtain $\Psi'(x_1,\dots,x_n,u,v)$ such that $\Phi(x_1,\dots,x_n) \equiv \Psi'(x_1,\dots,x_n,u,1)$. Now let $\Psi(x_1,\dots,x_n,u,v) =_{\mathrm{df}} E(\Psi'(x_1,\dots,x_n,u,v),v)$. Clearly $I \models_{\min} \Phi$ iff $I \cup \{u := 0, v := 1\} \models_{\min} \Psi$ ($I \models_{\max} \Phi$ iff $I \cup \{u := 1, v := 1\} \models_{\max} \Psi$, resp.).

Let $g_2 \in \mathrm{FP}$ compute the above transformation of Φ into Ψ. From an assignment for Ψ, remove the variables u and v by function $g_1 \in \mathrm{FP}$; or simply output $\perp$ if Ψ is unsatisfiable. The functions g_1, g_2 witness LexMin3-**SAT** $\leq^p_{\mathrm{met}}$ LexMin**SAT**(B) (LexMax3-**SAT** $\leq^p_{\mathrm{met}}$ LexMax**SAT**(B)). □

For other bases, the constants are not so easily obtained. Next we show how to exploit the order of variables in such a way that the minimal model of a formula is forced to assign the values 0 and 1 to particular variables. We will then use these variables as a replacement for the constants needed in Lemma 6.

Lemma 8. *Let B be a finite set of Boolean functions. If there are B-formulas $E(x,y,v,u)$, $N(x,v,u)$ and $F(v,u,x)$ such that $E(x,y,0,1) \equiv x \wedge y$, $N(x,0,1) \equiv \neg x$, $F(0,0,x) \equiv 0$, $F(0,1,x) \equiv x$ and not $F(1,0,x) \equiv F(1,1,x) \equiv 0$, then* LexMin**SAT**$(B)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP.

Proof. We show that LexMin3-**SAT** $\leq^p_{\mathrm{met}}$ LexMin**SAT**(B). Let be $\Phi(x_1,\dots,x_n)$ a 3-CNF formula. The function g_2 for the metric reduction works as described below:

Since there exist B-formulas E and N we can use Lemma 6 to obtain $\Psi'(x_1,\dots,x_n,v,u)$ in logarithmic space such that $\Phi(x_1,\dots,x_n) \equiv \Psi'(x_1,\dots,x_n,0,1)$. Let be $\Psi(x_1,\dots,x_n,v,u) =_{\mathrm{df}} F(v,u,\Psi')$. The variables are ordered by $v < u < x_1 < x_2 < \cdots < x_n$.

Claim 1: $I \models_{\min} \Phi$ iff $I \cup \{v := 0, u := 1\} \models_{\min} \Psi$.

"$\Rightarrow$": Let $I \models_{\min} \Phi$. Any $I' \models \Psi$ does not assign $u = v = 0$, so if there exists an $I' < I \cup \{v := 0, u := 1\}$ then $I'/\{u,v\} = \{v := 0, u := 1\}$ and $I'/\{x_1,\dots,x_n\} < I$. But this is a contradiction since $I'/\{x_1,\dots,x_n\} \models \Phi$ (recall that $\Psi(x_1,\dots,x_n,0,1) \equiv \Phi(x_1,\dots,x_n)$).

"$\Leftarrow$": Let $I \cup \{v := 0, u := 1\} \models_{\min} \Psi$. If there is an assignment $I' < I$ and $I' \models \Phi$ then $I' \cup \{v := 0, u := 1\} \models \Psi$, which is a contraction since $I' \cup \{v := 0, u := 1\} < I \cup \{v := 0, u := 1\}$.

Claim 2: Φ is unsatisfiable iff $I' \models_{\min} \Psi$ where $I' \geq \{x_1 = x_2 = \cdots = u := 0, v := 1\}$.

First note that Ψ is satisfiable since $F(1,0,\Xi) \not\equiv 0$ or $F(1,1,\Xi) \not\equiv 0$. So there exists an assignment $I \models \Psi$.

"$\Rightarrow$": Let $\Phi(x_1,\dots,x_n)$ be unsatisfiable. Suppose that there is an $I' \models \Psi$ such that $I' < \{x_1 = x_2 = \cdots x_n = u := 0, v := 1\}$ then $I'/\{x_1,x_2,\dots,x_n\} \models \Phi$, because I' cannot assign $v = 0$ and $u = 0$. But this is a contradiction since Φ is not satisfiable.

"$\Leftarrow$": Let $I' \models_{\min} \Psi$ and $I' \geq \{x_1 = x_2 = \cdots = u := 0, v := 1\}$. If $I \models \Phi$ then $I \cup \{v := 0, u := 1\} < \{x_1 = x_2 = \cdots x_n = u := 0, v := 1\}$ and $I \cup \{v := 0, u := 1\} \models \Psi$ which is a contradiction since $I' \models_{\min} \Psi$.

Now the function g_1 of the metric reduction simply outputs "$\perp$" if it gets an assignment I' greater than or equal to $\{x_1 = x_2 = \cdots = u := 0, v := 1\}$ as an input. In all other cases it removes the assignments for v and u and outputs the resulting assignment. This proves LexMin-3-**SAT** $\leq^p_{\mathrm{met}}$ LexMin**SAT**(B). □

For various non complete sets B one can show that formulas E, N, and F, as required in the above lemma, exist. Hence we obtain complexity results for more non complete classes of Boolean functions. As we pointed out before the statement of the lemma, in this case we do depend on the order of the variables (in contrast to Theorem 7).

Theorem 9. *If B is a finite set of functions such that $[B] \supseteq \mathrm{S}_{02}$, $[B] \supseteq \mathrm{S}_{12}$ or $[B] \supseteq \mathrm{D}_1$, then* LexMin**SAT**$(B)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP.

Proof. $[B] \supseteq \mathrm{S}_{02}$: From Proposition 2 we know that $g(x, y, z) = x \vee (y \wedge \neg z)$ is a base for S_{02}. Let be $g'(x, y) =_{\mathrm{df}} g(0, x, y) = x \wedge \overline{y}$. Clearly $g'(x, g'(x, y)) = x \wedge y$ and $g'(1, x) = \neg x$. Since $[B] \supseteq \mathrm{S}_{02}$ there exist B-formulas $E(x, y, v, u)$ and $N(x, v, u)$ such that $E(x, y, 0, u) \equiv x \wedge y$ (again the variable u is not used in E) and $N(x, 0, 1) \equiv \neg x$. Moreover note that $g(x, y, g(x, y, z)) = x \wedge (y \vee z)$ and therefore there exists a B-formula F such that $F(v, u, x) \equiv v \wedge (u \vee x)$. We obtain $F(0, 0, x) \equiv 0$, $F(1, 0, x) \equiv x$, $F(0, 1, x) \equiv 0$ and $F(1, 1, x) \equiv 1$. Using Lemmas 6 and 8 the statement follows.
$[B] \supseteq \mathrm{S}_{12}$: From Proposition 2 we know that $g(x, y, z) = x \wedge (y \vee \overline{z})$ is a base for S_{12}. Obviously $g(x, y, 1) = x \wedge y$, $g(1, 0, x) = \neg x$ and $g(x, y, g(x, y, z)) = x \vee (y \wedge z)$. Hence there exist B-formulas $E(x, y, v, u)$, $N(x, v, u)$ and $F(v, u, z)$ such that $E(x, y, 0, 1) \equiv x \wedge y$, $N(x, 0, 1) \equiv \neg x$ and $F(v, u, z) \equiv v \vee (u \wedge z)$. Hence, $F(0, 0, x) \equiv 0$, $F(0, 1, x) \equiv x$, $F(1, 0, x) \equiv 1$ and $F(1, 1, x) \equiv 1$. Using Lemmas 6 and 8 the statement follows.
$[B] \supseteq \mathrm{D}_1$: From Proposition 2 we know that $g(x, y, z) = xy \vee x\overline{z} \vee y\overline{z}$ is a base for D_1. We obtain $g(x, y, 1) = x \wedge y$, $g(0, 1, x) = \neg x$ and $g(x, y, g(x, y, z)) = xy \vee xz \vee yz$. Since $[B] \supseteq \mathrm{D}_1$ we know that there exists B-formulas $E(x, y, v, u)$, $N(x, v, u)$ and $F(v, u, x)$ such that $E(x, y, 0, 1) = x \wedge y$, $N(x, 0, 1) = \neg x$ and $F(v, u, x) = vu \vee vx \vee ux$. Hence, $F(0, 0, x) \equiv 0$, $F(0, 1, x) \equiv x$ and $F(1, 1, x) \equiv 1$. Using Lemmas 6 and 8 the statement follows. □

Combining Lemma 5 and Theorem 9, we are now ready to prove a dichotomy theorem for LexMin**SAT**(B) for arbitrary finite sets of Boolean functions B:

Corollary 10 (Dichotomy Theorem for LexMin**SAT**(B)**).** *Let B be a finite set of Boolean functions. If $[B] \supseteq \mathrm{S}_{02}$, $[B] \supseteq \mathrm{S}_{12}$ or $[B] \supseteq \mathrm{D}_1$, then* LexMin**SAT**$(B)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP. *In all other cases* LexMin**SAT**$(B) \in$ FP.

Proof. The first part of the statement is proved by Theorem 9. Now let B be a finite set of Boolean functions such that $[B] \not\supseteq \mathrm{S}_{02}$, $[B] \not\supseteq \mathrm{S}_{12}$ and $[B] \not\supseteq \mathrm{D}_1$. By inspecting Figure 1 we obtain that either $B \subseteq \mathrm{M}$ or $B \subseteq \mathrm{L}$. Using Lemma 5 the second part of the statement follows. □

Turning now to LexMax**SAT**(B), we observe that, if non $\in [B]$, the problem of finding the minimal satisfying assignment of a B-formula reduces to the problem of determining the maximal satisfying assignment of a B-formula. We will use this to show that LexMax**SAT**(B) is hard for OptP if B is the set of selfdual functions (D).

Lemma 11. *Let B be a finite set of Boolean functions. If* $\mathrm{non} \in [B]$*, then* $\mathrm{LexMin}\mathbf{SAT}(B) \leq^p_{\mathrm{met}}$ $\mathrm{LexMax}\mathbf{SAT}(B)$*.*

Proof. Let $\Phi(x_1, \ldots, x_n)$ be an arbitrary B-formula. We define the B-formula $\Phi^{\mathrm{R}}(x_1, \ldots, x_n) =_{\mathrm{df}}$ $\Phi(\neg x_1, \ldots, \neg x_n)$. Note that Φ^{R} is a B-formula since non $\in$ [B]. Clearly $I \models_{\min} \Phi \iff \neg I \models_{\max}$ Φ^{R}.

The metric reduction now works as follows: First g_2 replaces any occurrence of x_i by a formula representing non(x), for $1 \leq i \leq n$, resulting in Φ^{R}. The function g_1 simply maps $\perp$ to $\perp$ and any assignment I to $\neg I$. □

Now we are able to prove a Dichotomy Theorem for $\mathrm{LexMax}\mathbf{SAT}(B)$.

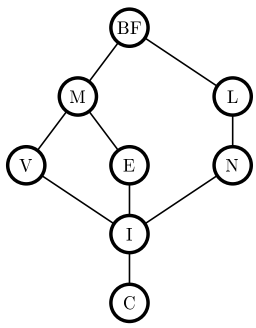
Corollary 12 (Dichotomy Theorem for $\mathrm{LexMax}\mathbf{SAT}(B)$**).** *Let B be a finite set of Boolean functions. If $[B] \supseteq \mathrm{S}_1$ or $[B] \supseteq \mathrm{D}$, then* $\mathrm{LexMax}\mathbf{SAT}(B)$ *is $\leq^p_{\mathrm{met}}$-complete for* MaxP*. In all other cases* $\mathrm{LexMax}\mathbf{SAT}(B) \in \mathrm{FP}$*.*

Proof. The case $[B] \supseteq \mathrm{S}_1$ is given in Theorem 7.

Now let $[B] \supseteq \mathrm{D}$. From Proposition 2 we know that $g(x, y, z) = x\overline{y} \vee x\overline{z} \vee \overline{yz}$ is a base of D. Clearly there exists a B-formula $N(x) \equiv \neg x$, because $g(y, y, x) \equiv \neg x$. Since LexMax-$\mathbf{SAT}(\mathrm{D}) \in \mathrm{MaxP}$, $\mathrm{LexMin}\mathbf{SAT}(\mathrm{D})$ is $\leq^p_{\mathrm{met}}$-complete for MinP (see Corollary 10) and LexMin-$\mathbf{SAT}(\mathrm{D}) \leq^p_{\mathrm{met}} \mathrm{LexMax}\mathbf{SAT}(\mathrm{D})$, we conclude that $\mathrm{LexMax}\mathbf{SAT}(\mathrm{D})$ is $\leq^p_{\mathrm{met}}$-complete for MaxP.

Finally, let B be a finite set of Boolean functions such that $[B] \not\supseteq \mathrm{S}_1$ and $[B] \not\supseteq \mathrm{D}$. Using Figure 1 we see that B must be monotone, linear or 1-reproducing. From Lemma 5 and Proposition 4 it follows that $\mathrm{LexMax}\mathbf{SAT}(B) \in \mathrm{FP}$. □

Finally, let us remark that a dichotomy result for B-formulas with constants, i.e. $0, 1 \in [B]$, is much easier. The graph of all closed classes containing the constants, given to the right, can be simply obtained by adding the constants to all classes shown in Figure 1 and determining the resulting class. The class I consists of all identity functions, the class C of all constant functions.



This graph can easily be found by adding the constant 0-ary functions 0 and 1 to all classes shown in Figure 1 and using Posts techniques to identify the resulting classes, see [RW00]. Now by the techniques of Lemma 5 and Lemma 6 the following dichotomy result can be deduced:

Corollary 13 (Dichotomy Theorem for $\mathrm{LexMin}\mathbf{SAT}(B \cup \{0, 1\})$ **and** $\mathrm{LexMax}\mathbf{SAT}(B \cup \{0, 1\})$**).** *Let B be a finite set of Boolean functions, such that $[B \cup \{0, 1\}] = \mathrm{BF}$. Then* LexMin-$\mathbf{SAT}(B \cup \{0, 1\})$ *(*$\mathrm{LexMax}\mathbf{SAT}(B \cup \{0, 1\})$*, resp.) is $\leq^p_{\mathrm{met}}$-complete for* MinP *(*MaxP*, resp.). In all other cases* $\mathrm{LexMin}\mathbf{SAT}(B \cup \{0, 1\}) \in \mathrm{FP}$ *(*$\mathrm{LexMax}\mathbf{SAT}(B \cup \{0, 1\}) \in \mathrm{FP}$*, resp.).*

3.3 Completeness Results for the Class P^{NP}

Given a function $f \colon \Sigma^* \to \Sigma^*$, define the set $L_f = \{\, x \in \Sigma^* \mid \text{the last bit of } f(x) \text{ is a } 1 \,\}$. Often it turns out that if f is complete for OptP under metric reductions, then the set L_f is complete for P^{NP} under usual many-one reductions; a precise statement is given below.

In our context the above problem translates to the question if the lexicographically minimal (maximal, resp.) satisfying assignment of a given B-formula sets the largest variable to 1. We will call this problem $\mathrm{OddLexMin}\mathbf{SAT}(B)$ ($\mathrm{OddLexMax}\mathbf{SAT}(B)$, resp.). The corresponding problems for unrestricted propositional formulas will be denoted by OddLexMin**SAT** and OddLexMax-**SAT**. The following result is known:

Proposition 14 ([Wag87]). OddLexMin**SAT** *and* OddLexMax**SAT** *are complete for the class* P^{NP} *under many-one reductions.*

If f is complete for MinP or MaxP under functional many-one reductions (that is, metric reductions where the outer function g_1 is the identity), then L_f is complete for P^{NP} under usual many-one reductions [Kre88], see also [Vol94]. In the case that f is only metric complete a similar result is not known. However, by a separate proof, the complexity of OddLexMin**SAT**(B) and OddLexMax**SAT**(B) can be determined.

Theorem 15 (Dichotomy Theorem for OddLexMin**SAT**(B) **and** OddLexMax**SAT**(B)**).** *For any finite set B of Boolean functions,* OddLexMin**SAT**(B) *(*OddLexMax**SAT**(B)*, resp.) is complete for* P^{NP} *iff* LexMin**SAT**(B) *(*LexMax**SAT**(B)*, resp.) is complete for* MinP *(*MaxP*, resp.).*

Proof. Observe that in the case that LexMin**SAT**$(B) \in$ FP (LexMax**SAT**$(B) \in$ FP, resp.) we are able to compute the lexicographically smallest (largest, resp.) solution in polynomial time. Because of this we can clearly decide OddLexMin**SAT**(B) (OddLexMax**SAT**(B), resp.) in polynomial time by checking the value of the most significant bit in this assignment.

Clearly OddLexMin**SAT**$(B) \in \mathrm{P}^{\mathrm{NP}}$ (OddLexMax**SAT**$(B) \in \mathrm{P}^{\mathrm{NP}}$, resp.). Now let LexMin-**SAT**(B) (LexMax**SAT**(B), resp.) be complete for MinP (MaxP, resp.). In these cases we show how to reduce OddLexMin3-**SAT** to OddLexMin**SAT**(B) (OddLexMax3-**SAT** to OddLexMax-**SAT**(B), resp.). Note that in all these cases $[B \cup \{0,1\}] = \mathrm{BF}$. Then there exists a B-formula $A(x,y,v,u)$ such that $A(x,y,0,1) \equiv x \leftrightarrow y$. Now modify the reduction function g_2 of Lemma 8 such that we obtain a formula $\Psi = F(u,v,(\Psi' \wedge A(x_n,z,v,u)))$, where z is larger than any other variable used in this construction. Now for any 3-CNF-formula Φ where $I \models_{\min} \Phi$ and $I' \models_{\min} \Psi$ it holds that $I(x_n) = I'(x_n) = I(z_n)$. So $\Phi \in$ OddLexMin3-**SAT** iff $\Psi \in$ OddLexMin**SAT**(B). By the same modification of the reduction function in Theorem 7 we can show that $\Phi \in$ OddLexMax-3-**SAT** iff $\Phi' \in$ OddLexMax**SAT**(B) holds. □

4 Constraint Satisfaction Problems

In this section, we will consider formulas that are given as a conjunction of constraints (given by Boolean functions applied to a subset of the variables). At first sight, one might hope that the machinery developed by Post and others is applicable here. However, the "upper-level" conjunction is of a restricted nature, and this does not fit into Post's definition of (unrestricted) superposition. Informally, Schaefer-like results cannot be obtained mechanically from Post/Lewis-like results, but new proofs are needed.

Dichotomy theorems for the problem to determine minimal satisfying assignments of constraint satisfaction problems with respect to the component-wise order of Boolean vectors were obtained in [KK99]. We here consider lexicographical ordering, as in the previous section.

4.1 Formulas Given by a Set of Constraints

Let S be a set of Boolean functions. In this section we will always assume that such S are nonempty and finite. *S-formulas in the Schaefer sense*, or, *S-CSPs*, will now be propositional formulas consisting of clauses built by using functions from S applied to arbitrary variables. Formally, let $S = \{f_1, f_2, \ldots, f_n\}$ be a set of Boolean functions and V be a set of variables. An *S-CSP* Φ (over V) is a finite conjunction of clauses $\Phi = C_1 \wedge \ldots \wedge C_k$, where each C_i is of the form $\tilde{f}(x_1, \ldots, x_k)$, $f \in S$, $\tilde{f}$ is the symbol representing f, k is the arity of f, and $x_1, \ldots, x_k \in V$. If some variables of an S-CSP Φ are replaced by the constants 0 or 1 then this new formula Φ' is called *S-CSP with constants.* If $\Phi = C_1 \wedge \ldots \wedge C_k$ is a CSP over V, and I is an assignment with respect to V, then $I \models \Phi$ if Φ satisfies all clauses C_i. Here, a clause $\tilde{f}(x_1, \ldots, x_k)$ is satisfied, if $f(I(x_1), \ldots, I(x_k)) = 1$.

We will consider different types of Boolean functions, following the terminology of Schaefer [Sch78].

- The Boolean function f is *0-valid* (*1-valid* , resp.) if $f(0, \ldots, 0) = 1$ ($f(1, \ldots, 1) = 1$, resp.).
- The Boolean function f is *Horn* (*anti-Horn*, resp.) if f is represented by a CNF formula having at most one unnegated (negated, resp.) variable in any conjunct.
- A Boolean function f is *bijunctive* if it is represented by a CNF formula having at most two variables in each conjunct.
- The Boolean function f is *affine* if it can be represented by a conjunction of affine functions.

We remark that Schaefer's term 1-valid coincides with Post's 1-reproducing.

A set S of Boolean functions is called 0-valid (1-valid, Horn, anti-Horn, affine, bijunctive, resp.) iff every function in S is 0-valid (1-valid, Horn, anti-Horn, affine, bijunctive, resp.).

The satisfiability problem for S-CSPs (S-CSPs with constants, resp.) is denoted by $\mathbf{CSP}(S)$ ($\mathbf{CSP}_C(S)$, resp.). Schaefer's main result, a dichotomy theorem for satisfiability of constraint satisfaction problems (i.e., propositional formulas of the form "conjunction of a set of constraints"), can be stated as follows:

Proposition 16 (Dichotomy Theorem for Constraint Satisfaction with Constants). *Let S be a set of Boolean functions. If S is Horn, anti-Horn, affine or bijunctive, then $\mathbf{CSP}_C(S)$ is polynomial-time decidable. Otherwise $\mathbf{CSP}_C(S)$ is* NP*-complete.*NP.

Proposition 17 (Dichotomy Theorem for Constraint Satisfaction). *Let S be a set of Boolean functions. If S is 0-valid, 1-valid, Horn, anti-Horn, affine or bijunctive, then $\mathbf{CSP}(S)$ is polynomial-time decidable. Otherwise $\mathbf{CSP}(S)$ is* NP*-complete.*

As a technical tool to obtain the above results, Schaefer defined the set of *existentially quantified S-CSPs with constants*, $\mathrm{Gen_C}(S)$, to be the smallest set of formulas having the following closure properties:

- For any $k \in \mathbb{N}$, any k-ary function $f \in S$, and $u \in (V \cup \{0,1\})^k$, the formula $\tilde{f}(u)$ is in $\mathrm{Gen_C}(S)$, where $\tilde{f}$ is a symbol for f.
- If Φ and Ψ are in $\mathrm{Gen_C}(S)$, then $\Phi \wedge \Psi$ and $(\exists x)\Phi$ (for $x \in V$) are in $\mathrm{Gen_C}(S)$.

Define $\mathrm{Gen}(S) =_{\mathrm{df}} \{\, \Phi \mid \Phi \in \mathrm{Gen_C}(S) \text{ and } \Phi \text{ has no constants} \,\}$, and

let $\mathrm{Rep_C}(S) =_{\mathrm{df}} \{\, f_\Phi \mid \Phi \in \mathrm{Gen_C}(S) \,\}$ and $\mathrm{Rep}(S) =_{\mathrm{df}} \{\, f_\Phi \mid \Phi \in \mathrm{Gen}(S) \,\}$ be the sets of functions represented by formulas from $\mathrm{Gen_C}(S)$ and $\mathrm{Gen}(S)$, resp. The following results proved by Schaefer will be needed in this paper:

Proposition 18 ([Sch78], Theorem 3.0). *Let S be a set of Boolean functions. If S is Horn, anti-Horn, affine or bijunctive, then* $\mathrm{Rep_C}(S)$ *satisfies the same condition. Otherwise,* $\mathrm{Rep_C}(S)$ *is the set of all Boolean functions* BF.

Proposition 19 ([Sch78], Lemma 4.3). *Let S be a set of Boolean functions. Then at least one of the following four statements holds:*

(1) S is 0-valid
(2) S is 1-valid
(3) $f_x, f_{\neg x} \in \mathrm{Rep}(S)$
(4) $f_{x \not\equiv y} \in \mathrm{Rep}(S)$

4.2 Dichotomy Theorems for Constraint Satisfaction Problems

The main result of this section is to answer the question for what syntactically restricted classes of formulas, given by a set S of Boolean constraints, Proposition 1 remains valid. For this, we will consider the following problems:

PROBLEM:	Lexicographically Minimal CSP (LexMin$\mathbf{CSP}(S)$)
INSTANCE:	an S-CSP Φ
OUTPUT:	the lexicographically smallest satisfying assignment of Φ, or "$\perp$" if Φ is unsatisfiable

The corresponding problem for S-CSPs with constants is denoted by LexMin$\mathbf{CSP}_C(S)$. We also examine the analogous maximization problems LexMax$\mathbf{CSP}(S)$ and LexMax$\mathbf{CSP}_C(S)$.

There are known algorithms for deciding satisfiability of CSPs in polynomial time for certain restricted classes of formulas [CH97]. We first observe that these algorithms can easily be modified to find minimal satisfying assignments.

Lemma 20. *Let S be a set of Boolean functions. If S is bijunctive, Horn, anti-Horn or affine, then* LexMin$\mathbf{CSP}_C(C)$, LexMin$\mathbf{CSP}(C) \in$ FP. *If S is 0-valid, then* LexMin$\mathbf{CSP}(S) \in$ FP.

Proof. It is well known (see, e.g., [Pap94]) that in all four cases, the satisfiability problem is in P. Hence, the claim follows using the algorithm given in the proof of Lemma 5. □

We remark that, if S contains at least one function which is not bijunctive, one function which is not Horn, one function which is not anti-Horn, and one function which is not affine, then LexMin$\mathbf{CSP}_C(S)$ cannot be in FP (unless P = NP), because Proposition 16 shows that the corresponding decision problem (which is the problem of deciding whether there is *any* satisfying assignment, not necessarily the minimal one) is NP-complete.

An analogous result, this time relying on Proposition 17, holds for LexMin$\mathbf{CSP}(S)$. The only case which requires a bit care is that of a 1-valid set S. Hardness here follows a result in [CH97], which shows that the problem to decide if there exists a satisfying assignment which differs from the vector $(1, 1, \ldots, 1)$ is NP-complete.

In the following we strengthen these observations by showing that if LexMin$\mathbf{CSP}_C(S)$ or LexMin$\mathbf{CSP}(S)$ are not contained in FP, then they are already complete for OptP.

Theorem 21. *Let S be a set of Boolean functions. If S does not fulfill the properties Horn, anti-Horn, bijunctive or affine, then* LexMin$\mathbf{CSP}_C(S)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP.

Proof. Obviously LexMin$\mathbf{CSP}_C(S) \in$ MinP. Now we have to prove $\leq^p_{\mathrm{met}}$-hardness for MinP.

If S does not fulfills the properties Horn, anti-Horn, bijunctive or affine then Proposition 18 shows that $\mathrm{Rep}_{\mathrm{C}}(S)$ includes all Boolean functions.

Let f_i be any Boolean function. Proposition 18 tells us that there exists an existentially quantified S-CSP $\Psi = \exists y_1 \ldots \exists y_k \Psi'$, representing f_i, where Ψ' contains no quantifier. Any clause of a 3-**SAT** formula can be represented by one out of a finite number of Boolean functions. So any clause C_i of a 3-**SAT** formula Φ can be represented by an S-CSP Φ_i. $\mathrm{Var}(\Phi_i)$ consists of the variables in $\mathrm{Var}(C_i)$ plus a number of variables of the form y_j. We pick different sets of y_j-variables for different formulas Φ_i.

Now we construct a function $g_2 \in \mathrm{FP}$ mapping a 3-**SAT** formula Φ into an S-CSP Ψ by replacing each C_i by the corresponding Φ'_i, where $\mathrm{Var}(\Psi)$ consists of $\mathrm{Var}(\Phi) = \{x_1, \ldots, x_n\}$ plus a set of variables of the form y_j. We order the variables by their index and by alphabet, i.e., $x_1 < x_2 < x_3 < \cdots < y_1 < y_2 < \cdots$. Note that we can drop the $\exists$-quantifiers of the variables y_j since we ask for a satisfying assignment of Ψ. The ordering of the variables ensures that in the minimal satisfying assignment of Ψ the variables in $\{x_1, \ldots, x_n\}$ will be minimal with respect to satisfaction of Φ.

Now the function $g_1 \in \mathrm{FP}$ shortens the assignment and removes all bits belonging to the variables y_j. Thus g_1 applied to the minimal satisfying assignment of $\Psi = g_2(\Phi)$ produces the minimal satisfying assignment for Φ. This shows that LexMin3-**SAT** $\leq^p_{\mathrm{met}}$ LexMin$\mathbf{CSP}_C(S)$. □

Mainly we are interested in formulas without constants. So we have to get rid of the constants in the construction of the just given proof. This is achieved in the reduction which we now present.

Theorem 22. *Let S be a set of Boolean functions. If S is not 0-valid, Horn, anti-Horn, bijunctive or affine, then* LexMin$\mathbf{CSP}(S)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP.

Proof. Clearly LexMin$\mathbf{CSP}(S) \in$ MinP. We want to show that LexMin$\mathbf{CSP}_C(S)$ reduces to LexMin$\mathbf{CSP}(S)$.

Case 1: S is not 1-valid.

Using Proposition 19 we know, that $f_x, f_{\neg x} \in \mathrm{Rep}(S)$ or $f_{x \not\equiv y} \in \mathrm{Rep}(S)$. In what follows, we again sort all variables first alphabetically, and then by index.

Case 1.1: $f_x, f_{\neg x} \in \mathrm{Rep}(S)$.

Let Φ an S-CSP with constants and $\mathrm{Var}(\Phi) = \{x_1, \ldots, x_n\}$. Now we can remove the constants by replacing any 1 by y_1 and 0 by y_0 and adding clauses representing $\{y_1\}$ and $\{\neg y_0\}$. Define the function g_2 such that $g_2(\Phi)$ performs exactly the just described replacement.

Now $I \models_{\min} \Phi$ if and only if $I' =_{\mathrm{df}} I \cup \{y_0 := 0, y_1 := 1\} \models_{\min} \Psi$, where $\Psi =_{\mathrm{df}} g_2(\Phi)$. The function g_1 removes the last two bits (assignments of y_0 and y_1) from I', showing that LexMin$\mathbf{CSP}_C(S) \leq^p_{\mathrm{met}}$ LexMin$\mathbf{CSP}(S)$.

Case 1.2: $f_{x \not\equiv y} \in \mathrm{Rep}(S)$.

Let Φ be an S-CSP with constants and $\mathrm{Var}(\Phi) = \{x_1, \ldots, x_n\}$. We construct an S-CSP $\Psi =_{\mathrm{df}} \Phi \left[\begin{smallmatrix}0\\u\end{smallmatrix}\right] \left[\begin{smallmatrix}1\\v\end{smallmatrix}\right] \wedge (u \not\equiv v)$ without constants, where u, v are new variables and $x_1 < x_2 < \cdots < x_n < u < v$. Define g_2 by $g_2(\Phi) = \Psi$. If $I_r \models_{\min} \Phi$ then clearly $I' =_{\mathrm{df}} I_r \cup \{u := 0, v := 1\} \models_{\min} \Psi$. If Φ is unsatisfiable, then either Ψ is unsatisfiable, too, or $I'' \models_{\min} \Psi$, where $I'' = I_w \cup \{u := 1, v := 0\}$.

A polynomial-time computable function g_1 can distinguish these cases and compute a minimal assignment for Φ or $\perp$, given an assignment for Ψ. The functions g_1 and g_2 show that LexMin$\mathbf{CSP}_C(S) \leq^p_{\mathrm{met}}$ LexMin$\mathbf{CSP}(S)$.

Case 2: S is 1-valid.

Having an S-CSP with constants we construct one without constants in polynomial time by g_2 as follows. Let $f \in S$ be a function which is not 0-valid but 1-valid and $\Psi =_{\mathrm{df}} \Phi \left[{0 \atop u} \right] \left[{1 \atop v} \right] \wedge \tilde{f}(v, \ldots, v)$, where u, v are new variables and $x_1 < x_2 < \cdots < x_n < u < v$. We claim that $I \models_{\min} \Phi$ iff $I \cup \{u := 0, v := 1\} \models_{\min} \Psi$.
First suppose that $I \models_{\min} \Phi$. It is clear from the clause $\tilde{f}(v, \ldots, v)$ that we have to choose $v := 1$. Since we are interested in the lexicographically smallest solution we have to choose $u := 0$ giving us immediately $I \cup \{u := 0, v := 1\} \models_{\min} \Psi$. Now let $I \cup \{u := 0, v := 1\} \models_{\min} \Psi$. Suppose that there exists a satisfying solution I_s for Φ being lexicographically smaller than I. Obviously $I_s \cup \{u := 0, v := 1\}$ is a lexicographically smaller satisfying assignment than $I \cup \{u := 0, v := 1\}$ giving us a contradiction to $I \cup \{u := 0, v := 1\} \models_{\min} \Psi$. The case that Φ is unsatisfiable can be detected as in case 1.2. This shows that $\mathrm{LexMin}\mathbf{CSP}_C(S) \leq^p_{\mathrm{met}}$ $\mathrm{LexMin}\mathbf{CSP}(S)$. □

Thus we get dichotomy theorems for finding lexicographically minimal satisfying assignments of CSPs, both for the case of formulas with constants and without constants.

Corollary 23 (Dichotomy Theorem for $\mathrm{LexMin}\mathbf{CSP}(\cdot)$ with constants). *Let S be a set of Boolean functions. If S is bijunctive, Horn, anti-Horn or affine, then* $\mathrm{LexMin}\mathbf{CSP}_C(S) \in \mathrm{FP}$. *In all other cases* $\mathrm{LexMin}\mathbf{CSP}_C(S)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP.

Corollary 24 (Dichotomy Theorem for $\mathrm{LexMin}\mathbf{CSP}(\cdot)$). *Let S be a set of Boolean functions. If S is 0-valid, bijunctive, Horn, anti-Horn or affine, then we have* $\mathrm{LexMin}\mathbf{CSP}(S) \in \mathrm{FP}$. *In all other cases* $\mathrm{LexMin}\mathbf{CSP}(S)$ *is $\leq^p_{\mathrm{met}}$-complete for* MinP.

If we compare the classes of functions in the statements of the above corollaries with those occurring in Schaefer's results (Propositions 16 and 17), we immediately obtain the following consequence which completely clarifies the connection between decision and optimization problems for constraint satisfaction.

Corollary 25. *Let S be a set of Boolean functions.*

1. $\mathbf{CSP}_C(S)$ *is* NP*-complete if and only if* $\mathrm{LexMin}\mathbf{CSP}_C(S)$ *is* MinP *complete.*
2. *If* $\mathbf{CSP}(S)$ *is* NP*-complete then* $\mathrm{LexMin}\mathbf{CSP}(S)$ *is* MinP *complete.*
3. *If S is a set of Boolean functions which is 1-valid but is not 0-valid, Horn, anti-Horn, bijunctive, or affine, then* $\mathbf{CSP}(S)$ *is in* P *but* $\mathrm{LexMin}\mathbf{CSP}(S)$ *is* MinP *complete.*

Results analogous to the above for the problem of finding maximal assignments can be proved:

Theorem 26 (Dichotomy Theorem for $\mathrm{LexMax}\mathbf{CSP}(\cdot)$). *Let S be a set of Boolean functions.*

1. *If S is bijunctive, Horn, anti-Horn or affine, then* $\mathrm{LexMax}\mathbf{CSP}_C(S) \in \mathrm{FP}$. *In all other cases* $\mathrm{LexMax}\mathbf{CSP}_C(S)$ *is $\leq^p_{\mathrm{met}}$-complete for* MaxP.
2. *If S is 1-valid, bijunctive, Horn, anti-Horn or affine, then* $\mathrm{LexMax}\mathbf{CSP}(S) \in \mathrm{FP}$. *Otherwise* $\mathrm{LexMax}\mathbf{CSP}(S)$ *is $\leq^p_{\mathrm{met}}$-complete for* MaxP.

4.3 Completeness Results for the Class P^{NP}

Similar as in Subsect. 3.3 for B-formulas, we now examine the problem if the largest variable in a lexicographically minimal assignment of a given S-CSP gets the value 1. Let us denote this problem by OddLexMin$\mathbf{CSP}(S)$, and in the case that S-CSPs with constants are allowed by OddLexMin$\mathbf{CSP}_C(S)$. (In the case of maximization we use the notation OddLexMax$\mathbf{CSP}(S)$ and OddLexMax$\mathbf{CSP}_C(S)$.)

Theorem 27 (Dichotomy Theorem for OddLexMin$\mathbf{CSP}(\cdot)$ **with constants).** *Let S be a set of Boolean functions. If S is bijunctive, Horn, anti-Horn or affine, then* OddLexMin$\mathbf{CSP}_C(S) \in \mathrm{P}$. *In all other cases* OddLexMin$\mathbf{CSP}_C(S)$ *is complete for* P^{NP} *under many-one reductions.*

Proof. If S is bijunctive, Horn, anti-Horn or affine, then OddLexMin$\mathbf{CSP}_C(S) \in \mathrm{P}$, since we can use the technique of Lemma 20 to find the minimal assignment, and then we accept if and only if the truth value 1 is assigned to the largest variable.

In the other cases we reduce OddLexMin3-$\mathbf{SAT}$ to OddLexMin$\mathbf{CSP}_C(S)$. In the proof of Theorem 21 we showed how to transform an arbitrary formula Φ with $\mathrm{Var}(\Phi) = \{x_1, \ldots, x_n\}$ into an S-CSP at the cost of introducing new variables of the form y_j. We modify this construction as follows: Introduce one more variable z (larger than all the other variables). Transform Φ into Φ' as described in Theorem 21. Finally set $\Phi'' = \Phi' \wedge (x_n \equiv z)$. (Observe that the predicate $\equiv$ is in $\mathrm{Rep}_{\mathrm{C}}(S)$.) Let I, I', I'' be the minimal satisfying assignments of Φ, Φ' and Φ''. Observe that they all agree on assignments of the variables in $\mathrm{Var}(\Phi)$. Now we have $I(x_n) = I'(x_n) = I''(x_n) = I''(z)$. Thus $\Phi \in$ OddLexMin3-$\mathbf{SAT}$ if and only if $\Phi'' \in$ OddLexMin$\mathbf{CSP}_C(S)$, which proves the claimed hardness result. □

Theorem 28 (Dichotomy Theorem for OddLexMin$\mathbf{CSP}(\cdot)$**).** *Let S be a set of Boolean functions. If S is 0-valid, bijunctive, Horn, anti-Horn or affine, then* OddLexMin$\mathbf{CSP}(S) \in \mathrm{P}$. *In all other cases* OddLexMin$\mathbf{CSP}(S)$ *is complete for* P^{NP} *under many-one reductions.*

Proof. Similar to the proof of the previous theorem. The easy case is obvious. In the hard case define Φ'' as above, and then use the construction of Theorem 22 to remove the constants. Let Φ''' be the resulting formula. The variables introduced in this last step should be smaller than z. Then we can argue as in the previous proof that z is assigned one in a minimal assignment for Φ''' if and only if x_n is assigned one in a minimal assignment for Φ. □

Again, results for maximal assignments are proved analogously:

Theorem 29 (Dichotomy Theorem for OddLexMax$\mathbf{CSP}(\cdot)$**).** *Let S be a set of Boolean functions.*

1. *If S is bijunctive, Horn, anti-Horn or affine, then* OddLexMax$\mathbf{CSP}_C(S) \in \mathrm{P}$. *In all other cases* OddLexMax$\mathbf{CSP}_C(S)$ *is complete for* P^{NP} *under many-one reductions.*
2. *If S is 0-valid, bijunctive, Horn, anti-Horn or affine, then* OddLexMax$\mathbf{CSP}(S) \in \mathrm{P}$. *In all other cases* OddLexMax$\mathbf{CSP}(S)$ *is complete for* P^{NP} *under many-one reductions.*

5 Conclusion

In this paper we determined the complexity of the problem to compute the lexicographically minimal or maximal satisfying assignment of a given propositional formula, and the problem to determine if in this assignment the largest variable is one, for different restricted formula classes. We obtained a number of dichotomy results, showing that the complexity of the first problem is either in FP or OptP-complete, while the latter problem is either in P or P^{NP}-complete.

One might ask if it is not possible to obtain our results about constraint satisfaction problems from the seemingly more general results obtained in Sect. 3. However, as we pointed out at the beginning of Sect. 4, this seems not to be the case. Another hint in that direction is that the results we obtain in the constraint satisfaction context do not speak about closed sets of Boolean functions (the Schaefer classes 0-valid, (anti-)Horn, and bijunctive are not closed in the sense of Post).

It can be seen that we do not need the full power of metric reductions in this paper. In fact, let us define f to be weakly many-one reducible to h if there are functions $g_1, g_2 \in \mathrm{FP}$ where $g_1(z)$ is always a sub-word of z, such that for all x, $f(x) = g_1(h(g_2(x)))$. The step to many-one reductions (where g_1 is the identity) thus is smaller than in the case of metric reductions. It is easy to observe that all the above given completeness results also hold for weak many-one reductions instead of metric reductions. The question that now arises is of course if we can even prove our completeness results for many-one reductions (i.e, always have $g_1(x) = x$). However this cannot be expected for "syntactic" reasons. For example, in Sect. 3, if we use a non-complete base $\mathrm{S}_1 \subseteq B \subset \mathrm{BF}$ we have to introduce new variables for using them as a replacement for the constants we need to construct our B-formulas. The assignments to these variables have to be removed later, which means that we need the full power of a weak many-one reduction to do some final manipulation of the value of the function we reduce to.

Acknowledgment. We are grateful to Nadia Creignou, Caen, and Klaus W. Wagner, Würzburg, for a lot of helpful hints.

References

[CH96] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125:1–12, 1996.

[CH97] N. Creignou and J.-J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 31(6):499–511, 1997.

[CKS00] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems.* Monographs on Discrete Applied Mathematics. SIAM, 2000.

[Cre95] N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51:511–522, 1995.

[JGK70] S. W. Jablonski, G. P. Gawrilow, and W. B. Kudrajawzew. *Boolesche Funktionen und Postsche Klassen.* Akademie-Verlag, 1970.

[KK99] L. Kirousis and P. K. Kolaitis. Dichotomy theorems for minimal satisfiability. manuscript, 1999.

[Köb89] J. Köbler. *Strukturelle Komplexität von Anzahlproblemen.* PhD thesis, Universität Stuttgart, Fakultät für Informatik, 1989.

[Kre88] M. W. Krentel. The complexity of optimization functions. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[Kre92] M. W. Krentel. Generalizations of OptP to the polynomial hierarchy. *Theoretical Computer Science*, 97:183–198, 1992.

[KST97] S. Khanna, M. Sudan, and L. Trevisan. Constraint satisfaction: The approximability of minimization problems. In *Proceedings 12th Computational Complexity Conference*, pages 282–296. IEEE Computer Society Press, 1997.

[KSW97] S. Khanna, M. Sudan, and D. Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proceedings 29th Symposium on Theory of Computing*, pages 11–20. ACM Press, 1997.

[Lad75] R. Ladner. On the structure of polynomial-time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[Lew79] Harry R. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[Pos41] E. L. Post. Two-value iterative systems, 1941.

[RW00] Steffen Reith and Klaus W. Wagner. The complexity of problems defined by boolean circuits. Technical Report 255, Institut für Informatik, Universität Würzburg, 2000. To appear in Proceedings *International Conference Mathematical Foundation of Informatics*, Hanoi, October 25 - 28, 1999.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proccedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.

[Vol94] H. Vollmer. On different reducibility notions for function classes. In *Proceedings 11th Symposium on Theoretical Aspects of Computer Science*, volume 775 of *Lecture Notes in Computer Science*, pages 449–460. Springer Verlag, 1994.

[VW95] H. Vollmer and K. W. Wagner. Complexity classes of optimization functions. *Information and Computation*, 120:198–219, 1995.

[Wag87] K. W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.