The Complexity of Computing Optimal Assignments of Generalized Propositional Formulae

Steffen Reith and *Heribert Vollmer*

Lehrstuhl für Theoretische Informatik
Universität Würzburg
Am Exerzierplatz 3
D-97072 Würzburg, Germany
`[streit,vollmer]@informatik.uni-wuerzburg.de`

Running head: The Complexity of Computing Optimal Assignments

Résumé

Nous étudions le problème du calcul de l'affectation booléenne minimale (ou maximale, selon l'ordre lexicographique) satisfaisant une expression booléenne, en fonction de restrictions imposées à la classe des expressions considérées. Il appert que pour chaque classe envisagée, le problème est soit résoluble en temps polynomial, soit complet pour $OptP$. Nous examinons également le problème de décider si la plus grande variable prends la valeur 1 selon l'affectation optimale. Nos montrons que ce problème est soit dans P, soit complet pour P^{NP}.

Mots-clés: Complexité du calcul, problèmes de satisfaisabilité, optimisation.

Abstract

We consider the problems of finding the lexicographically minimal (or maximal) satisfying assignment of propositional formulae for different restricted formula classes. It turns out that for each class from our framework, the above problem is 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.

Keywords: Computational complexity, satisfiability problems, optimization

1 Introduction

In 1978 Thomas J. Schaefer proved a remarkable result. He examined satisfiability of propositional formulae for certain syntactically restricted formula classes. Each such class is given by a set S of boolean relations allowed when constructing formulae. An *S-formula* is a conjunction of clauses, where each clause consists out of a relation from S applied to some propositional variables. $\mathbf{SAT}(S)$ now is the problem to decide for a given S-formula if it is satisfiable. Schaefer showed that depending on S the problem $\mathbf{SAT}(S)$ is either (1) efficiently (i. e. polynomial time) computable or (2) *NP*-complete; and he gave a simple criterion that, given some S, allows to determine whether (1) or (2) holds. Since (depending on S) the complexity of $\mathbf{SAT}(S)$ is either easy or hard (and there is nothing in between), Schaefer called this a "dichotomy theorem for satisfiability."

In the last few years his result regained interest among complexity theorists. In 1995 Nadia Creignou examined the problem of determining the maximal number of clauses of a given S-formula that can be satisfied simultaneously. Interestingly she also obtained a dichotomy theorem: She proved that this problem is either polynomial-time solvable or *MaxSNP*-complete, depending on properties of S [1]. (In 1997 the approximability of this problem and the corresponding minimization problem was examined in [6, 5], leading to a number of deep results.) The complexity of counting problems and enumeration problems based on satisfiability of S-formulae was examined in [3, 2].

The problem of maximizing (or minimizing) the number of clauses satisfied in (unrestricted) propositional formula is complete for the class *MaxSNP* (or *MinSNP*). These classes, introduced in 1988 by Papadimitriou and Yannakakis [10] (see also [9, pp. 311ff]), are of immense importance in the theory of approximability of hard optimization problems. Of equal importance however is the class *OptP*, introduced by Krentel in 1988 [7]. While *MaxSNP* and *MinSNP* are defined logically making use of Fagin's characterization of *NP* [4], the class *OptP* is defined using Turing machines. *OptP* is a superclass of *MaxSNP* and *MinSNP*. The canonical complete problems for *OptP* are the problems *LexMax*$\mathbf{SAT}$ and *LexMin*$\mathbf{SAT}$ of determining the lexicographically maximal (or minimal) satisfying assignment of a given (unrestricted) propositional formula.

In this paper we examine *LexMax*$\mathbf{SAT}$ and *LexMin*$\mathbf{SAT}$ for classes of S-formulae. We show that both problems are either polynomial-time solvable or *OptP* complete, depending on properties of S. That is, we prove a dichotomy theorem for the *LexMax*$\mathbf{SAT}$ (and *LexMin*$\mathbf{SAT}$) problem. Comparing our results with those of Schaefer we gain insight in the connection between the complexity of a decision problem and the corresponding optimization problem. We show for example that if constants are allowed in S formulae, then the problem of deciding satisfiability is NP-complete if and only if the problem of finding the smallest assignment is $OptP$-complete. (In the case that constants are forbidden, an analogous result does not hold unless $P = NP$.)

Generally the connection between decision problems and optimization problems is open. It can very well be that an optimization problem is hard (complete) though the decision problem is trivial. Here we show that in the case that constants are allowed, this cannot happen: a decision problem is hard if and only if the corresponding optimization problem is hard. In contrast to this, if constants are forbidden then we completely identify those cases where the optimization problem is hard and the decision problem is easy. We hope that these results help to better understand the connection between the complexity of decision problems and optimization problems.

From an $OptP$-complete optimization problem one can sometimes obtain a decision problem that is complete for P^{NP}. In our case this is the $OddMin$**SAT** (or $OddMax$**SAT**) problem, for an exact definition refer to Sect. 5. We prove that this problem is either polynomial-time solvable or complete for P^{NP}; that is we again get a dichotomy theorem.

2 Preliminaries

Any subset $R \subseteq \{0,1\}^k$ is called a *k-ary boolean relation* (*k-ary logical relation*). The integer k is called the *rank* of R. If k is not needed or is clear from the context we use *boolean relation* (*logical relation*) for short. Since we need symbols representing boolean relations in the formulae we construct, we always use lowercase letters for relation symbols and uppercase letters for the relation itself. So the relation symbol r represents the relation R.

We will consider different types of relations, following the terminology of Schaefer [11].

1. The boolean relation R is *0-valid* (*1-valid*, resp.) iff $(0, \ldots, 0) \in R$ ($(1, \ldots, 1) \in R$, resp.).
2. The boolean relation R is *Horn* (*anti-Horn*, resp.) iff R is logically equivalent to a CNF formula having at most one unnegated (negated, resp.) variable in any conjunct.
3. A boolean relation R is *bijunctive* iff it is logically equivalent to a CNF formula having one or two variables in each conjunct.

4. The boolean relation R is *affine* iff it is logically equivalent to a system of linear equations over the finite field $\mathbb{Z}_2$. This means that any tuple $(v_1, \ldots, v_k) \in R$ is a solution of a system of formulae of the form $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$ or $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$.

Now let $S = \{R_1, \ldots, R_n\}$ be a set of boolean relations. In the rest of this paper we will always assume that such S are nonempty and finite. S is called 0-valid (1-valid, Horn, anti-Horn, affine, bijunctive, resp.) iff every relation $R_i \in S$ is 0-valid (1-valid, Horn, anti-Horn, affine, bijunctive, resp.).

S formulae will now be propositional formulae consisting of clauses built by using relations from S applied to arbitrary variables. Formally, let $S = \{R_1, R_2, \ldots, R_n\}$ be a set of logical relations and V be a set of variables. We will always assume an ordering on V. An *S-formula 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 $r(x_1, \ldots, x_k)$, $R \in S$, r is the symbol representing R, k is the rank of R, and $x_1, \ldots, x_k \in V$. If some variables of an S-formula Φ are replaced by the constants 0 or 1 then this new formula Φ' is called *S-formula with constants*. By $Var(\Phi) \subseteq V$ we denote the subset of those variables actually used in Φ.

The satisfiability problem for S-formulae (S-formulae with constants, resp.) is denoted by $\mathbf{SAT}_{NC}(S)$ ($\mathbf{SAT}_{C}(S)$, resp.).

By $\Phi \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ we denote the formula created by simultaneously replacing each occurrence of x in Φ by y, where x, y are either variables or a constants. Now we define the set of *existentially quantified S-formulae with constants*, again following Schaefer. Let $Gen_C(S)$ the smallest set of formulae having the following closure properties: For any $k \in \mathbb{N}$ and any k-ary relation $R \in S$ where $x_1, \ldots, x_k \in V$, the formula $r(x_1, \ldots, x_k)$ is in $Gen_C(S)$. Now let Φ and Ψ be in $Gen_C(S)$, $x, y \in V$, then $\Phi \wedge \Psi$, $\Phi \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$, $\Phi \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix}\right]$, $\Phi \left[\begin{smallmatrix} x \\ 0 \end{smallmatrix}\right]$ and $(\exists x)\Phi$ are in $Gen_C(S)$, for $x, y \in V$. Define $Gen_{NC}(S) =_{\text{def}} \{\Phi | \Phi \in Gen_C(S)$ and Φ has no constants$\}$. For $\Phi \in Gen_C(S)$ let $Var(\Phi)$ be the set of variables with free occurrences in Φ.

Let Φ be an S-formula with k variables. If $Var(\Phi) = \{x_1, \ldots, x_k\}$, $x_1 < \cdots < x_k$ (recall that V is ordered), then an assignment I: $Var(\Phi) \to \{0, 1\}$ where $I(x_i) = a_i$ will also be denoted by $(a_1, \ldots, a_k)$. The ordering on 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 $i \le k$ 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*. That an assignment $(a_1, \ldots, a_k) \in \{0, 1\}^k$ satisfies Φ will be denoted by $(a_1, \ldots, a_k) \models \Phi$. 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 $(a'_1, \ldots, a'_k) \models \Phi$.

If $I: \{x_1, x_2, \ldots\} \to \{0,1\}$ is an arbitrary assignment, y is variable and $a \in \{0,1\}$, then $I \cup \{y := a\}$ denotes the assignment I' defined by $I'(y) = a$ and $I'(x) = I(x)$ for all $x \neq y$.

Let $[\Phi] =_{\text{def}} \{(a_1, \ldots, a_k) \in \{0,1\}^k | \mathit{Var}(\Phi) = \{x_1, \ldots, x_k\}$ and $(a_1, \ldots, a_k) \models \Phi\}$ be the logical relation defined by Φ, and let

$$\mathit{Rep}_C(S) =_{\text{def}} \{[\Phi] | \Phi \in \mathit{Gen}_C(S)\}$$
$$\mathit{Rep}_{NC}(S) =_{\text{def}} \{[\Phi] | \Phi \in \mathit{Gen}_{NC}(S)\}$$

The following results proved by Schaefer will be needed in this paper.

Proposition 1 ([11], Theorem 3.0). *Let S be a set of logical relations. If S is Horn, anti-Horn, affine or bijunctive, then $\mathit{Rep}_C(S)$ satisfies the same condition. Otherwise, $\mathit{Rep}_C(S)$ is the set of all logical relations.*

Proposition 2 ([11], Lemma 4.3). *Let S be a set of logical relations. Then at least one of the following four statements holds:*

(1) *S is 0-valid*
(2) *S is 1-valid*
(3) $[x], [\neg x] \in \mathit{Rep}_{NC}(S)$
(4) $[x \not\equiv y] \in \mathit{Rep}_{NC}(S)$

Schaefer's main result, a dichotomy theorem for satisfiability of propositional formulae, can be stated as follows:

Proposition 3 (Dichotomy Theorem for Satisfiability with Constants). *Let S be a set of logical relations. If S is Horn, anti-Horn, affine or bijunctive, then $\mathbf{SAT}_C(S)$ is polynomial-time decidable. Otherwise $\mathbf{SAT}_C(S)$ is NP-complete.*

Proposition 4 (Dichotomy Theorem for Satisfiability). *Let S be a set of logical relations. If S is 0-valid, 1-valid, Horn, anti-Horn, affine or bijunctive, then $\mathbf{SAT}_{NC}(S)$ is polynomial-time decidable. Otherwise $\mathbf{SAT}_{NC}(S)$ is NP-complete.*

By $\mathbf{SAT}^*(S)$ we denote the problem to decide whether there exists a satisfying assignment for an S-formula which is different from $(0,0,\ldots,0)$ and $(1,1,\ldots,1)$. The following proposition is from Creignou and Hébrard [2].

Proposition 5. *Let S be a set of logical relations. If S is not Horn, anti-Horn, affine and bijunctive, then $\mathbf{SAT}^*(S)$ is NP-complete.*

3 Maximization and Minimization Problems

The study of optimization problems in computational complexity theory started with the work of Krentel [7, 8]. 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 classes relevant in our context using a characterization given in [13].

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. Using one of the well-known bijections between Σ^* and the set of natural numbers (e.g. dyadic encoding) we may also think of *FP* (and the other classes of functions defined below) as a class of number-theoretic functions. Say that a function h belongs to the class *MinP* if there is a function $f \in FP$ and a polynomial p such that for all x,

$$h(x) = \min_{|y| \le p(|x|)} f(x, y).$$

The class *MaxP* is defined by taking the maximum of these values. Finally, let $OptP = MinP \cup MaxP$.

Krentel considered the following reducibility in connection with these classes: A function f is *metric reducible* to h ($f \le^p_{\mathrm{met}} h$) if there exist two functions $g_1, g_2 \in FP$ such that for all x:

$$f(x) = g_1(h(g_2(x)), x).$$

As a side remark let us mention that the closure of all three classes *MinP*, *MaxP*, and *OptP* under metric reductions coincides with the class FP^{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 [7, 13, 12].

Krentel gave in [7] a number of problems complete for *OptP* under metric reducibility. The for us most important complete problem for *OptP* is the problem of finding the lexicographically minimal satisfying assignment of a given formula.

PROBLEM:	*LexMin***SAT**
INSTANCE:	a propositional formula Φ
OUTPUT:	the lexicographically smallest satisfying assignment of Φ

The problem *LexMax***SAT** is defined analogously.

Proposition 6 ([7]). *$LexMin$**SAT** and $LexMax$**SAT** are complete for OptP under metric reductions.*

One of the main points of this paper is to answer the question for what syntactically restricted classes of formulae (given by a set S of boolean relations) the above proposition remains valid. For this, we will consider the following problems:

PROBLEM:	Lexicographically Minimal **SAT** ($LexMin\mathbf{SAT}_{NC}(S)$)
INSTANCE:	An S-formula Φ
OUTPUT:	The lexicographically smallest satisfying assignment of Φ

PROBLEM:	Lexicographically Minimal **SAT** with constants ($LexMin\mathbf{SAT}_{C}(S)$)
INSTANCE:	An S-formula Φ with constants
OUTPUT:	The lexicographically smallest satisfying assignment of Φ

PROBLEM:	Lexicographically Maximal **SAT** ($LexMax\mathbf{SAT}_{NC}(S)$)
INSTANCE:	An S-formula Φ
OUTPUT:	The lexicographically largest satisfying assignment of Φ

PROBLEM:	Lexicographically Maximal **SAT** with constants ($LexMax\mathbf{SAT}_{C}(S)$)
INSTANCE:	An S-formula Φ with constants
OUTPUT:	The lexicographically largest satisfying assignment of Φ

4 A Dichotomy Theorem for *OptP*

There are known algorithms for deciding satisfiability of given formulae in polynomial time for certain restricted classes of formulae. We first observe that these algorithms can easily be modified to find minimal satisfying assignments. We first consider formulae with constants and then turn to the case where no constants are allowed.

Theorem 1. *Let S be a set of logical relations. If S is bijunctive, Horn, anti-Horn or affine, then we have $LexMin\mathbf{SAT}_C(C) \in FP$. In all other cases $LexMin\mathbf{SAT}_C(C) \notin FP$ unless $P = NP$.*

Proof. For the cases that S is bijunctive, Horn, anti-Horn or affine, there are well-known polynomial time procedures to *decide* satisfiability of a given formula (see e.g. [9]; for the case of affine S we use Gaussian elimination).

Now we can use the algorithm in Fig. 1 for finding the lexicographically smallest satisfying solution. This algorithm is an easy modification of an algorithm from [2]. Note that lines 5 and 8 of the algorithm do not change one of the properties bijunctive, horn, anti-horn and affine; so the test whether e is satisfiable runs also in deterministic polynomial time for the modified formula. Since we always try first to assign $x_i = 0$ we obtain the lexicographically smallest satisfying assignment.

Now let S contain at least one relation which is not bijunctive, one relation which is not Horn, one relation which is not anti-Horn, and one relation which is not affine. Then $\mathit{LexMin}\mathbf{SAT}_{NC}(S)$ cannot be in FP (unless $P = \mathit{NP}$), because Proposition 3 shows that the corresponding decision problem (which is the problem of deciding whether there is *any* satisfying assignment, not necessarily the minimal one) is log-complete for NP. □

Theorem 2. *Let S be a set of logical relations. If S is 0-valid, bijunctive, Horn, anti-Horn or affine, then we have $\mathit{LexMin}\mathbf{SAT}_{NC}(S) \in \mathit{FP}$. In all other cases $\mathit{LexMin}\mathbf{SAT}_{NC}(S) \notin \mathit{FP}$ unless $P = \mathit{NP}$.*

Proof. The case "0-valid" is obvious. For the cases that S is bijunctive, Horn, anti-Horn or affine, we can use the same algorithms as in the previous theorem to *decide* satisfiability, and again we use the algorithm in Fig. 1 for finding the lexicographically smallest satisfying solution.

Now let S contain at least one relation which is not 0-valid, one relation which is not bijunctive, one relation which is not Horn, one relation which is not anti-Horn, and one relation which is not affine.

Case 1: There is a relation in S which is not 1-valid. Then $\mathit{LexMin}\mathbf{SAT}_{NC}(S)$ cannot be in FP (unless $P = \mathit{NP}$), because Proposition 4 shows that the corresponding decision problem is log-complete for NP.

Case 2: S is 1-valid, i.e. we know that the 0-vector is not a satisfying assignment of the given formula but the 1-vector is; and we have to solve the question if there is a lexicographically smaller one. However Proposition 5 shows that the problem of deciding whether *any* assignment different from the 0- or 1-vector exists is NP-complete; thus finding the lexicographically smallest solution cannot be in FP unless $P = \mathit{NP}$. □

Now we know that there are easy (polynomial time solvable) cases of finding lexicographically minimal satisfying assignments, and other cases where under the assumption

Input: Boolean formula Φ over S with $Var(\Phi) = \{x_1, \ldots, x_n\}$
Output: Lexicographically minimal satisfying assignment $A \in \{0,1\}^n$

$e \leftarrow \Phi$;
if (Φ is satisfiable) **then**
    for $i \leftarrow 1$ **to** n **do**
        if ($e \wedge \neg x_i$ is satisfiable) **then**
            $e \leftarrow (e \wedge \neg x_i)$;
            $A[i] \leftarrow 0$;
        else
            $e \leftarrow (e \wedge x_i)$;
            $A[i] \leftarrow 1$;
        end if
    end for
    writeln(A);
else
    writeln("0");
end if

Fig. 1. Algorithm to calculate the lexicographically minimal satisfying assignment

that $P \neq NP$ no efficient way exists. However this leaves open the possibility that in the latter case different levels of inefficiency depending on the properties of S can occur. The following two theorems rule out this possibility. In the case that the lex min sat problem is not in P it is already $MinP$ complete under metric reductions.

We first consider the (easier) case of formulae where constants are allowed.

Theorem 3. *Let S be a set of logical relations. If S does not fulfill the properties Horn, anti-Horn, bijunctive or affine then $LexMin\mathbf{SAT}_C(S)$ is $\leq^p_{\mathrm{met}}$-complete for $MinP$.*

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

If S does not fulfills the properties Horn, anti-Horn, bijunctive or affine then Proposition 1 shows that $Rep_C(S)$ includes all boolean relations.

Let R_i be any logical relation. Proposition 1 tells us that there exists an S-formula $\Phi = \exists y_1 \ldots \exists y_k \Phi'$, representing R_i, where Φ' contains no quantifier. Any clause of a 3-**SAT** formula can be represented by a finite number of boolean relations. So any clause C_i of a 3-**SAT** formula Φ can be represented by an S-formula Φ_i. $Var(\Phi_i)$ consists of the variables in $Var(C_i)$ plus a number of variables of the form y_j. We pick different sets of y_j-variables for different formulae Φ_i.

Now we construct a function $g_2 \in FP$ mapping a 3-**SAT** formula Φ into an S-formula Φ' by replacing each C_i by the corresponding Φ'_i, where $Var(\Phi')$ consists out of $\{x_1, \ldots, x_n\}$ 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 Φ'. The ordering of the variables ensures that in the minimal satisfying assignment of Φ' the variables in $\{x_1, \ldots, x_n\}$ will be minimal with respect to satisfaction of Φ.

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

Note that our proof heavily hinges on Schaefer's Proposition 1. However Schaefer's technique always introduces new variables, which pose no problem in his context, but are not allowed here. We can only remove these new variables in the end because we have the power of metric reductions.

Mainly we are interested in formulae 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 4. *Let S be a set of logical relations. If S is not 0-valid, Horn, anti-Horn, bijunctive or affine, then $\mathit{LexMin}\mathbf{SAT}_{NC}(S)$ is $\leq^p_{\mathrm{met}}$-complete for MinP.*

Proof. Clearly $\mathit{LexMin}\mathbf{SAT}_{NC}(S) \in \mathit{MinP}$. We want to show that $\mathit{LexMin}\mathbf{SAT}_{C}(S)$ reduces to $\mathit{LexMin}\mathbf{SAT}_{NC}(S)$.

Case 1: S is not 1-valid.

Using Proposition 2 we know, that $[x], [\neg x] \in \mathit{Rep}_{NC}(S)$ or $[x \not\equiv y] \in \mathit{Rep}_{NC}(S)$. In what follows, we again sort all variables by index and alphabet.

Case 1.1: $[x], [\neg x] \in \mathit{Rep}_{NC}(S)$.

Let Φ an S-formula with constants and $\mathit{Var}(\Phi) = \{x_1, \ldots, x_n\}$. 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 $I' =_{\mathrm{def}} (I \cup \{y_0 := 0, y_1 := 1\}) \models_{\min} \Phi'$, where $\Phi' =_{\mathrm{def}} g_2(\Phi)$. The function g_1 removes the last two bits (assignments of y_0 and y_1) from I', showing that $\mathit{LexMin}\mathbf{SAT}_{C}(C) \leq^p_{\mathrm{met}} \mathit{LexMin}\mathbf{SAT}_{NC}(S)$.

Case 1.2: $[x \not\equiv y] \in \mathit{Rep}_{NC}(S)$.

Let Φ an S-formula with constants and $\mathit{Var}(\Phi) = \{x_1, \ldots, x_n\}$. We construct an S-formula $\Phi' =_{\mathrm{def}} \Phi \begin{bmatrix} 0 \\ u \end{bmatrix} \begin{bmatrix} 1 \\ v \end{bmatrix} \wedge (u \not\equiv v)$ without constants. Define g_2 by $g_2(\Phi) = \Phi'$. Now suppose there exists a satisfying assignment $I' =_{\mathrm{def}} I_w \cup \{u := 1, v := 0\}$. This would be an unwanted assignment, since v should represent 1 and u should represent 0. But there exists also the correct satisfying assignment $I'' =_{\mathrm{def}} I_r \cup \{u := 0, v := 1\}$, where $I_r \models_{\min} \Phi$. This assignment is clearly lexicographically smaller than I' and thus $I'' \models_{\min} \Phi'$ iff $I_r \models_{\min} \Phi$.

Now we remove the assignment for u and v by g_1. The functions g_1 and g_2 show that $\mathit{LexMin}\mathbf{SAT}_{C}(S) \leq^p_{\mathrm{met}} \mathit{LexMin}\mathbf{SAT}_{NC}(S)$.

Case 2: S is 1-valid.

Having an S-formula with constants we construct one without constants in polynomial time by g_2 as follows. Let $R \in S$ a relation which is not 0-valid but 1-valid and $\Phi' =_{\mathrm{def}} \Phi \begin{bmatrix} 0 \\ u \end{bmatrix} \begin{bmatrix} 1 \\ v \end{bmatrix} \wedge R(v, \ldots, v)$. We claim that $I \models_{\min} \Phi$ iff $I \cup \{u := 0, v := 1\} \models_{\min} \Phi'$.

First suppose that $I \models_{\min} \Phi$. It is clear from the clause $R(v, \ldots, v)$ 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 $I \cup \{u := 0, v := 1\} \models \Phi'$ and certainly also $I \cup \{u := 0, v := 1\} \models_{\min} \Phi'$. Now let $I \cup \{u := 0, v := 1\} \models_{\min} \Phi'$. 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 $I \cup \{u := 0, v := 1\} \models_{\min} \Phi'$.

We remove the assignment for u and v by g_1, showing that *LexMin*$\mathbf{SAT}_C(C) \leq^p_{\mathrm{met}}$ *LexMin*$\mathbf{SAT}_{NC}(S)$. □

Observe that Schaefer's Proposition 2 is not sufficient to obtain the above result. Our proof substantially depends on the ability to force a suitable ordering of the assignments by ordering the variables in a reasonable way.

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

Corollary 1 (Dichotomy Theorem for *LexMin*SAT with constants). *Let S be a set of logical relations. If S is bijunctive, Horn, anti-Horn or affine, then we have LexMin-*$\mathbf{SAT}_C(C) \in$ *FP. In all other cases LexMin*$\mathbf{SAT}_C(C)$ *is $\leq^p_{\mathrm{met}}$-complete for MinP.*

Corollary 2 (Dichotomy Theorem for *LexMin*SAT). *Let S be a set of logical relations. If S is 0-valid, bijunctive, Horn, anti-Horn or affine, then we have LexMin-*$\mathbf{SAT}_{NC}(S) \in$ *FP. In all other cases LexMin*$\mathbf{SAT}_{NC}(S)$ *is $\leq^p_{\mathrm{met}}$-complete for MinP.*

If we compare the classes of relations in the statements of the above corollaries with those needed in Schaefer's results (Propositions 3 and 4), the following consequence is immediate:

Corollary 3. *Let S be a set of logical relations.*

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

The above corollary completely clarifies the connection between decision and optimization for the optimal assignments problem.

Example 1. Hierarchical **SAT** is the variant of 3-**SAT** where only unnegated variables occur and we require that in each clause if either the first or the second variable are satisfied then the third variable is not satisfied, and if the third variable is satisfied then also the first and second variable are satisfied. In our framework this problem is given by $S = \{R\}$, where $R = \{(1,0,0),(0,1,0),(1,1,1)\}$. It can be seen using techniques from [11] that S is 1-valid but is not 0-valid, Horn, anti-Horn, bijunctive, or affine. Thus $\mathbf{SAT}_{NC}(S)$ is in P but $\mathit{LexMin}\mathbf{SAT}_{NC}(S)$ is MinP complete.

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

Theorem 5 (Dichotomy Theorem for LexMax**SAT).** *Let S be a set of logical relations.*

1. *If S is bijunctive, Horn, anti-Horn or affine, then $\mathit{LexMax}\mathbf{SAT}_C(C) \in \mathit{FP}$. Otherwise $\mathit{LexMax}\mathbf{SAT}_C(C)$ is $\leq^p_{\mathrm{met}}$-complete for MaxP.*
2. *If S is 1-valid, bijunctive, Horn, anti-Horn or affine, then $\mathit{LexMax}\mathbf{SAT}_{NC}(S) \in \mathit{FP}$. Otherwise $\mathit{LexMax}\mathbf{SAT}_{NC}(S)$ is $\leq^p_{\mathrm{met}}$-complete for MaxP.*

If we look at the definition of metric reductions (see Sect. 3) and compare this with the proofs given above, we see that we do not need the full power of metric reductions here. In fact the function g_1 in our proof is a function which, first, does not depend on x but only on $g_2(x)$, and second, g_1 is "almost" the identity function—$g_1(z)$ is obtained from z by simply stripping away a few bits. Since g_1 is almost the identity, let us call these reductions *weak many-one reductions;* that is, f is weakly many-one reducible to h if there are two functions $g_1, g_2 \in \mathit{FP}$ where $g_1(z)$ is always a sub-word of z, such that for all x,

$$f(x) = g_1(h(g_2(x))).$$

Theorem 6. *All the above given completeness results also hold for weak many-one reductions instead of metric reductions.*

Proof. A close look at Krentel's work shows that Proposition 6 also holds for weak many-one reductions. The reductions given above in the proofs of Theorems 3 and 4 are in fact weak many-one reductions. Since these reductions are transitive our theorem follows. □

The question that now arises is of course if we can even prove our completeness results for many-one reductions, which are weak many-one reductions where g_1 is the identity function. However this cannot be expected for "syntactic" reasons, since when we manipulate a given formula Φ constructing Φ' such that $Var(\Phi) \neq Var(\Phi')$ then an assignment of Φ' simply by definition cannot be an assignment of Φ. And it seems that there is no way of getting around this; we have to change the variable set.

5 A Dichotomy Theorem for P^{NP}

Given a function $f\colon \mathbb{N} \to \mathbb{N}$, define the set $L_f = \{x \in \Sigma^* \mid f(x) \equiv 1 \pmod 2\}$. 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 largest variable in a lexicographically minimal assignment of a given S-formula gets the value 1. Let us denote this problem by $OddMin\mathbf{SAT}_{NC}(S)$, and in the case that S-formulae with constants are allowed by $OddMin\mathbf{SAT}_{C}(S)$. (In the case of maximal assignments we use the notation $OddMax\mathbf{SAT}_{C}(S)$ and $OddMax\mathbf{SAT}_{NC}(S)$.) The corresponding problems for unrestricted propositional formulae will be denoted by $OddMin\mathbf{SAT}$ and $OddMax\mathbf{SAT}$.

Proposition 7 ([7]). *$OddMin\mathbf{SAT}$ and $OddMax\mathbf{SAT}$ are complete for the class P^{NP} under many-one reductions.*

It is known that if f is complete for $MinP$ or $MaxP$ under many-one reductions (see the discussion at the end of Sect. 4) then L_f is complete for P^{NP} under usual many-one reductions [7], see also [12]. In the case that f is only metric complete or weakly many-one complete, a similar result is not known. Since in Sect. 4 we proved completeness under weak many-one reductions we cannot by the above remark mechanically translate our results for $\mathbf{SAT}_{NC}(S)$ to completeness results for $OddMin\mathbf{SAT}_{NC}(S)$ for the class P^{NP}. However by separate proofs we can determine the complexity of $OddMin\mathbf{SAT}_{C}(S)$ and $OddMin\mathbf{SAT}_{NC}(S)$.

Theorem 7 (Dichotomy Theorem for $OddMin\mathbf{SAT}$ with constants). *Let S be a set of logical relations. If S is bijunctive, Horn, anti-Horn or affine, then we have $OddMin\mathbf{SAT}_{C}(S) \in P$. In all other cases $OddMin\mathbf{SAT}_{C}(S)$ is complete for P^{NP} under many-one reductions.*

Proof. If S is bijunctive, Horn, anti-Horn or affine, then $OddMin\mathbf{SAT}_{C}(S) \in P$, since we can use Algorithm 1 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 *OddMin*3-**SAT** to *OddMin*$\mathbf{SAT}_C(S)$. In the proof of Theorem 3 we showed how to transform an arbitrary formula Φ with $Var(\Phi) = \{x_1, \ldots, x_n\}$ into an S-formula 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 3. Finally set $\Phi'' = \Phi' \wedge (x_n \equiv z)$. (Observe that the predicate $\equiv$ is in $Rep_C(S)$.) 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 $Var(\Phi)$. Now we have

$$I(x_n) = I'(x_n) = I''(x_n) = I''(z).$$

Thus $\Phi \in$ *OddMin*3-**SAT** if and only if $\Phi'' \in$ *OddMin*$\mathbf{SAT}_C(S)$, which proves the claimed hardness result. □

Theorem 8 (Dichotomy Theorem for ***OddMin*****SAT).** *Let S be a set of logical relations. If S is 0-valid, bijunctive, Horn, anti-Horn or affine, then we have OddMin-*$\mathbf{SAT}_{NC}(S) \in P$. *In all other cases OddMin*$\mathbf{SAT}_{NC}(S)$ *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 4 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, analogous results for maximal assignments can be proved:

Theorem 9 (Dichotomy Theorem for ***OddMax*****SAT).** *Let S be a set of logical relations.*

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

Acknowledgment. We are extremely grateful to Nadia Creignou, Caen, for a lot of helpful hints.

References

1. N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51:511–522, 1995.
2. N. Creignou and J.-J. Hébrard. On generating all solutions of generalized satisfiability problems, 1997. Submitted for publication.
3. N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125:1–12, 1996.
4. R. Fagin. Generalized first-order spectra and polynomial time recognizable sets. In R. Karp, editor, *Complexity of Computations*, pages 43–73, 1974.
5. 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.
6. 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.
7. M. W. Krentel. The complexity of optimization functions. *Journal of Computer and System Sciences*, 36:490–509, 1988.
8. M. W. Krentel. Generalizations of OptP to the polynomial hierarchy. *Theoretical Computer Science*, 97:183–198, 1992.
9. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.
10. C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. In *Proceedings 20th Symposium on Theory of Computing*, pages 229–234. ACM Press, 1988.
11. T. J. Schaefer. The complexity of satisfiability problems. In *Proccedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.
12. 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.
13. H. Vollmer and K. W. Wagner. Complexity classes of optimization functions. *Information and Computation*, 120:198–219, 1995.