INFORMATIQUE THÉORIQUE ET APPLICATIONS

N. CREIGNOU

J.-J. HEBRARD

On generating all solutions of generalized satisfiability problems

Informatique théorique et applications, tome 31, nº 6 (1997), p. 499-511

<http://www.numdam.org/item?id=ITA_1997__31_6_499_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 31, n° 6, 1997, pp. 499-511)

ON GENERATING ALL SOLUTIONS OF GENERALIZED SATISFIABILITY PROBLEMS (*)

by N. CREIGNOU (**) (1) and J.-J. HEBRARD (2)

Communicated by Peter van EMDE BOAS

Abstract. – *We examine whether all solutions of Generalized Satisfiability problems can be generated efficiently. By refining Schaefer's [11] result we show that there exists a class $\mathcal{G}$ of problems such that for every problem in $\mathcal{G}$ there exists a polynomial delay generating algorithm and for every Generalized Satisfiability problem not in $\mathcal{G}$ such an algorithm does not exist unless $P = NP$. The class $\mathcal{G}$ is made up of the problems equivalent to the satisfiability problem for conjunction of Horn clauses, anti-Horn clauses, 2-clauses or XOR-clauses.*
[11] Thomas J. Schaefer, The complexity of satisfiability problems, Proc. 10th Ann., ACM Symposium of Theory of Computing, Association for Computing Machinery, New-York (1978), 216-226.

Résumé. – *Nous regardons si toutes les solutions de problèmes de satisfaisabilité généralisée peuvent être énumérées efficacement. En raffinant le résultat de Schaefer [11] nous prouvons qu'il existe une classe de problèmes $\mathcal{G}$ telle que pour tout problème dans $\mathcal{G}$ il existe un algorithme à délai polynomial qui génère toutes les solutions alors que pour tout problème n'appartenant pas à $\mathcal{G}$ un tel algorithme n'existe pas à moins que $P = NP$. La classe $\mathcal{G}$ est constituée des problèmes équivalents au problème de la satisfaisabilité pour une conjonction de clauses de Horn, de clauses anti-Horn, de 2-clauses ou de XOR-clauses.*

1. INTRODUCTION

A generalized satisfiability problem (GS problem) is defined by fixing a finite set Σ of predicate symbols and for each $R \in \Sigma$ a set $\mathcal{E}(R) \subseteq \{0,1\}^k$, where k is the arity of R. The GS problem $\mathrm{Sat}(\Sigma, \mathcal{E})$ is to decide whether a given conjunction e of expressions of the form $R(v_1, \dots, v_k)$, where $R \in \Sigma$ and $v_1, \dots, v_k$ are variables, is satisfiable. More precisely the question is whether there exists a function s from the set of variables of e to $\{0,1\}$ such

(*) Received february 1996, corrigé le 14 août 1997, accepté le 7 janvier 1998.
(**) Corresponding author. The paper was partially written during a secondment to CNRS, Institut de Mathématiques de Luminy
(1) Département de Mathématiques, Université de Caen, 14032 Caen, France.
Nadia.Creignou@math.unicaen.fr
(2) Département d'Informatique, Université de Caen, 14032 Caen, France.
hebrard@info.unicaen.fr

that for each $R(v_1, \ldots, v_k)$ in e, $(s(v_1), \ldots, s(v_k)) \in \mathcal{E}(R)$. In other words, GS problems are Constraint Satisfaction Problems on Boolean domain [10].

Schaefer [11] showed that every GS problem is either in P or NP-complete. This result is surprising and unexpected. Indeed, a classic theorem from Ladner [9] shows that if $P \neq NP$ then there are infinitely many complexity classes which are sandwiched in between. Rarely in complexity theory one comes across an infinite class of problems where every problem belongs to a finite set of complexity classes. Let us mention that a similar dichotomy theorem was obtained by Hell and Nešetřil [6] for H-coloring problems (which can be seen as binary constraint satisfaction problems). With each GS problem is associated a counting problem in which the question is to determine the number of satisfying assignments of a given expression. Creignou and Hermann [3] showed that if the GS problem is equivalent to the satisfiability problem of a conjunction of XOR-clauses then the number of solutions can be computed in polynomial time, otherwise it is $\#P$-complete.

The aim of this paper is to examine whether the solutions of GS problems can be generated efficiently, i.e. generated with polynomial delay. A generating algorithm has polynomial delay if it generates all solutions in such a way that the delay between any two consecutive solutions is bounded by a polynomial in the input size [8]. By refining Schaefer's result we show that there exists a class $\mathcal{G}$ of GS problems such that for every problem in $\mathcal{G}$ there exists a polynomial delay generating algorithm and for every GS problem not in $\mathcal{G}$ such an algorithm does not exist unless $P = NP$. The class $\mathcal{G}$ is made up of the problems equivalent to the satisfiability problem for conjunction of Horn clauses, anti-Horn clauses, 2-clauses or XOR-clauses.

2. PRELIMINARIES

A *literal* l is a variable x or its negation $\neg x$. A *clause* is a disjunction of literals $l_1 \vee \ldots \vee l_n$. A *CNF formula* is a conjunction of clauses. A *2-clause* is a clause having at most two literals. A *2-CNF formula* is a conjunction of 2-clauses. A *Horn clause* (respectively an *anti-Horn clause*) is a clause having at most one unnegated (resp. negated) variable. A *Horn* (resp. *anti-Horn*) *CNF formula* is a conjunction of Horn (resp. anti-Horn) clauses. An *XOR-clause*, $l_1 \oplus \ldots \oplus l_n$, is a clause in which the usual disjunction $\vee$ is replaced by the exclusive-or operator denoted by $\oplus$. An *XOR-CNF formula* is a conjunction of XOR-clauses.

Let Σ be a finite set of predicate symbols. Let V be a set of variables. A *simple expression* is of the form $R(x_1, \ldots, x_k)$, where $R \in \Sigma$ is a predicate symbol of arity k and $x_1, \dots, x_k$ are variables. An *expression* e is a conjunction of simple expressions.

A *simple expression with constants* is of the form $R(x_1, \dots, x_k)$, where $R \in \Sigma$ is a predicate symbol of arity k and $x_1, \dots, x_k$ belong to $V \cup \{0, 1\}$. An *expression with constants* e is a conjunction of simple expressions with constants.

Let e be an expression with constants. If u is either a variable or a constant and v is either a variable or a constant, then $e[u/v]$ denotes the expression formed from e by replacing each occurrence of v by u. If V is a set of variables, then $e[u/V]$ denotes the result of substituting u for every occurrence of every variable in V. If $v_1, \dots, v_k$ are variables occurring in e and if $u_1, \dots, u_k$ belong to $V \cup \{0, 1\}$ then $e[u_1/v_1, \dots, u_k/v_k]$ denotes the expression with constants obtained from e by replacing each occurrence of v_i by u_i for $i = 1, \dots, k$.

With each predicate symbol R of arity k we associate the set $\mathcal{E}(R) \subseteq \{0, 1\}^k$, its interpretation.

Let $e = e_1 \wedge \dots \wedge e_m$ be an expression and $\mathcal{V}ar(e) = \{v_1, \dots, v_n\}$ be the set of variables occurring in e. Let $(a_1, \dots, a_n)$ be in $\{0, 1\}^n$ and $e' = e[a_1/v_1, \dots, a_n/v_n] = e'_1 \wedge \dots \wedge e'_m$. We say that *the vector* $(a_1, \dots, a_n)$ *satisfies* e if for every $e'_i = R_i(a_{i_1}, \dots, a_{i_{k_i}})$ we have $(a_{i_1}, \dots, a_{i_{k_i}}) \in \mathcal{E}(R_i)$.

We say that the expression e is satisfiable if there exists $s : \mathcal{V}ar(e) = \{v_1, \dots, v_n\} \mapsto \{0, 1\}$ such that $(s(v_1), \dots, s(v_n))$ satisfies e.

With each finite set $\Sigma = \{R_1, \dots, R_p\}$ of predicate symbols with their interpretation $\mathcal{E}(\Sigma) = \{\mathcal{E}(R_1), \dots, \mathcal{E}(R_p)\}$ we associate a satisfiability decision problem, denoted by $\mathrm{Sat}(\Sigma, \mathcal{E})$, defined as follows:

DEFINITION 2.1 [Generalized Satisfiability Problem ($\mathrm{Sat}(\Sigma, \mathcal{E})$)]:
Input: *An expression* e *over* Σ.

Question: *Is* e *satisfiable?*

This class contains many well-known Satisfiability problems such as 3Sat, Monotone-3Sat, Not-All-Equal-3Sat or One-In-3Sat [5, page 259].

EXAMPLE 2.2: *The classical problem* 2Sat *can be expressed as* $\mathrm{Sat}(\Sigma, \mathcal{E})$ *where* $\Sigma = \{R_0, R_1, R_2\}$ *with* $\mathcal{E}(R_0) = \{(0,1), (1,0), (1,1)\}$, $\mathcal{E}(R_1) = \{(0,0), (0,1), (1,1)\}$ *and* $\mathcal{E}(R_2) = \{(0,0), (0,1), (1,0)\}$. *A typical instance, for example* $(x \vee y) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee z) \wedge (\bar{z} \vee \bar{x})$, *is then given by the following expression* $R_0(x, y) \wedge R_1(x, y) \wedge R_1(y, z) \wedge R_2(z, x)$.

Throughout the paper we will use the following notation.
If e is an expression with $\mathcal{V}ar(e) = \{v_1, \ldots, v_n\}$, then $e(v_1, \ldots, v_n)$ denotes the set $\{(a_1, \ldots, a_n) \in \{0,1\}^n \ / \ (a_1, \ldots, a_n)$ satisfies $e\}$ and is referred to as the set of vectors satisfying e.
This notation will naturally be extended to any conjunction of clauses ϕ.

DEFINITION 2.3: *Let I be a subset of $\{0,1\}^k$. The set I is said to be:*

0-valid *if* $(0, \ldots, 0) \in I$, **1-valid** *if* $(1, \ldots, 1) \in I$,

Horn *(respectively* **anti-Horn***) if there exists some Horn (resp. anti-Horn) CNF formula ϕ such that $I = \phi(v_1, \ldots, v_k)$,*

bijunctive *if there exists some 2-CNF formula ϕ such that $I = \phi(v_1, \ldots, v_k)$,*

affine *if there exists some XOR-CNF formula ϕ such that $I = \phi(v_1, \ldots, v_k)$,*

complementive *if* for all $s = (a_1, \ldots a_k) \in I$ *we have* $\bar{s} = 1 - s = (1 - a_1, \ldots, 1 - a_k) \in I$.

From now on we suppose that $(\Sigma, \mathcal{E})$, with $\Sigma = \{R_1, \ldots, R_p\}$ and $\mathcal{E}(\Sigma) = \{\mathcal{E}(R_1), \ldots, \mathcal{E}(R_p)\}$, is fixed.

3. GENERATING ALGORITHMS

A *generating algorithm* is an algorithm that generates all configurations that satisfy a given specification (e.g., all maximal independent sets of a given graph, all satisfying truth assignment of a given formula) without duplicate. One has to be careful in defining the notion of polynomial time for such algorithms. Indeed, in most interesting problems the number of configurations to be generated is potentially exponential in the size of the input (say a graph or a formula). For this reason Johnson, Yannakakis and Papadimitriou defined *polynomial-delay algorithms* [8]:

DEFINITION 3.4: *A generating algorithm has polynomial delay if it generates the configurations, one after the other, in such a way that the delay until the first is output, and thereafter the delay between any two consecutive configurations (and between the last configuration and the halting), is bounded by a polynomial in the input size.*

The aim of this section is to provide an efficient canonical algorithm for listing the set of vectors satisfying a given expression for some special generalized satisfiability problems.

Throughout this section ϕ with $\mathcal{V}ar(\phi) = \{v_1, \ldots, v_n\}$ will denote either a 2-CNF formula, or a Horn CNF formula or an anti-Horn CNF formula or an XOR-CNF formula. Let us recall that one can decide in polynomial time whether ϕ is satisfiable: an algorithm for 2-CNF formulae is given in [1], Horn and anti-Horn formulae are treated in [4], and for XOR-CNF formulae it suffices to identify ϕ with a system of linear equations over the field $GF(2)$ and to test its consistency by using Gaussian elimination.

The algorithm A below outputs all the vectors that satisfy ϕ. The variable M denotes a list made up of 0's and 1's and is initially empty. The procedure $Cons(a, M)$, where $a = 0$ or 1, inserts a onto the head of M. When the instruction $Output(M)$ is performed, the list M is a new vector $(a_1, \ldots, a_n)$ which satisfies ϕ. The algorithm A has polynomial delay since one can decide whether $\phi \wedge v_p$ and $\phi \wedge \neg v_p$ are satisfiable in polynomial time for $p = 1, \ldots, n$ (see the criterion introduced by Valiant [12, [Fact 7]).

Algorithm A

Input: ϕ with $\mathcal{V}ar(\phi) = \{v_1, \ldots, v_n\}$.
Output: All vectors satisfying ϕ.

Begin
  If ϕ is satisfiable
  Then $Generate(\phi, (), n)$
End

Procedure $Generate(\phi, M, p)$
Begin
  If $p = 0$
  Then $Output(M)$
  Else [**if** $\phi \wedge v_p$ is satisfiable **then** $Generate(\phi \wedge v_p, Cons(1, M), p-1)$;
    if $\phi \wedge \neg v_p$ is satisfiable **then** $Generate(\phi \wedge \neg v_p, Cons(0, M), p-1)$]
End

PROPOSITION 3.5: *If Σ and $\mathcal{E}$ verify one of the four following conditions, then there exists a polynomial-delay algorithm that generates all satisfying vectors of a given expression.*

1. *For every R in Σ, $\mathcal{E}(R)$ is Horn.*

2. *For every R in Σ, $\mathcal{E}(R)$ is anti-Horn.*

3. *For every R in Σ, $\mathcal{E}(R)$ is affine.*

4. *For every R in Σ, $\mathcal{E}(R)$ is bijunctive.*

Proof: This follows quite easily from the algorithm described above. We only have to transform the expression given as input into a CNF formula having the appropriate form. For example, suppose that for every R in Σ, $\mathcal{E}(R)$ is bijunctive. Let $e = e_1 \wedge \ldots \wedge e_m$ be an expression over Σ with n variables. For each i, $i = 1, \ldots, m$, $e_i = R_i(x_1, \ldots, x_{k_i})$ and $\mathcal{E}(R_i)$ is bijunctive. Let ψ_i be the 2-CNF formula such that $\mathcal{E}(R_i) = \psi_i(y_1, \ldots, y_{k_i})$. We set $\phi_i = \psi_i[x_1/y_1, \ldots, x_{k_i}/y_{k_i}]$ and $\phi = \phi_1 \wedge \ldots \wedge \phi_m$. Let $(a_1, \ldots, a_n)$ be a vector in $\{0,1\}^n$. By definition $(a_1, \ldots, a_n)$ satisfies e iff $(a_1, \ldots, a_n)$ satisfies ϕ. Hence, in order to generate all the vectors satisfying e it suffices to construct ϕ and to apply the algorithm A. Since ϕ can be constructed in linear time from e we get a polynomial-delay algorithm. •

4. DICHOTOMY THEOREM

Our aim is to show that the problems identified in the previous section are exactly those for which the set of satisfying vectors of a given expression can be enumerated with polynomial delay.

First let us recall Schaefer's result that identifies problems for which finding a first solution is NP-hard.

PROPOSITION 4.6 [11]: *If there are predicate symbols R_0, R_1, R_2, R_3, R_4 and R_5 in Σ such that $\mathcal{E}(R_0)$ is non-0-valid, $\mathcal{E}(R_1)$ is non-1-valid, $\mathcal{E}(R_2)$ is non-Horn, $\mathcal{E}(R_3)$ is non-anti-Horn, $\mathcal{E}(R_4)$ is non-affine and $\mathcal{E}(R_5)$ is non-bijunctive, then* $\mathrm{Sat}(\Sigma, \mathcal{E})$ *is NP-complete.*

Consequently, under the conditions stated in this proposition it is obvious that there is no polynomial-delay listing algorithm, unless P=NP.

Now it remains to deal with the case where every R in Σ is such that $\mathcal{E}(R)$ is 0-valid (or 1-valid). In this case it is clear that listing a first satisfying assignment for any given expression is trivial. Hence, a natural question arises: how difficult is it to find a second solution? The following Proposition 4.7 answers this question. It entails that there exist satisfiability problems for which a first solution can be generated trivially, after which finding a second one becomes hard.

By $\mathrm{Sat}^*(\Sigma, \mathcal{E})$ we denote a variant of $\mathrm{Sat}(\Sigma, \mathcal{E})$ in which the question is whether there exists a satisfying vector which is different from all-zero, $\vec{0}$, and all-one, $\vec{1}$.

PROPOSITION 4.7: *If there are predicate symbols R_2, R_3, R_4 and R_5 in Σ such that, $\mathcal{E}(R_2)$ is non-Horn, $\mathcal{E}(R_3)$ is non-anti-Horn, $\mathcal{E}(R_4)$ is non-affine and $\mathcal{E}(R_5)$ is non-bijunctive, then $\mathrm{Sat}^*(\Sigma, \mathcal{E})$ is NP-complete.*

Let us denote by $\mathrm{Sat}_c(\Sigma, \mathcal{E})$ a variant of $\mathrm{Sat}(\Sigma, \mathcal{E})$ in which the input is an expression with constants. Our proof will be based on the following result due to Schaefer.

PROPOSITION 4.8 [11]: *If there are predicate symbols R_2, R_3, R_4 and R_5 in Σ such that, $\mathcal{E}(R_2)$ is non-Horn, $\mathcal{E}(R_3)$ is non-anti-Horn, $\mathcal{E}(R_4)$ is non-affine and $\mathcal{E}(R_5)$ is non-bijunctive, then $\mathrm{Sat}_c(\Sigma, \mathcal{E})$ is NP-complete.*

We will also need the following technical lemma.

LEMMA 4.9: *Suppose that there are predicate symbols R, R' and R'' in Σ such that $\mathcal{E}(R)$ is non-affine, $\mathcal{E}(R')$ is non-Horn and $\mathcal{E}(R'')$ is non-anti-Horn.*

1. *If for every R in Σ, $\mathcal{E}(R)$ is 0-valid, then there exists an expression g_0 over Σ having no constant other than 0 such that:*

   - *either g_0 involves exactly two variables u, v and*
   $$g_0(u, v) = \{(0,0), (0,1), (1,1)\}$$
   - *or g_0 involves exactly three variables u, v, w and*
   $$g_0(u, v, w) = \{(0,0,0), (1,0,1), (0,1,1)\}.$$

2. *If for every R in Σ, $\mathcal{E}(R)$ is 1-valid, then there exists an expression g_1 over Σ having no constant other than 1 such that:*

   - *either g_1 involves exactly two variables u, v and*
   $$g_1(u, v) = \{(0,0), (0,1), (1,1)\}$$
   - *or g_1 involves exactly three variables u, v, w and*
   $$g_1(u, v, w) = \{(1,1,1), (0,0,1), (0,1,0)\}.$$

   Roughly speaking g_0 and g_1 represent the implication or "almost" the implication.

Proof: Let us first introduce some terminology. If s is a vector of $\{0,1\}^n$ then $s(i)$, $1 \leq i \leq n$, denotes its ith component. We define three operations on vectors:

$$(s_1 \cap s_2)(i) = 1 \text{ iff } s_1(i) = s_2(i) = 1,$$
$$(s_1 \cup s_2)(i) = 0 \text{ iff } s_1(i) = s_2(i) = 0,$$
$$(s_1 \oplus s_2)(i) = 0 \text{ iff } s_1(i) = s_2(i).$$

Suppose that for every R in Σ, $\mathcal{E}(R)$ is 0-valid (the case 1-valid can be treated in a similar manner). Let R be a predicate symbol in Σ of arity k such that $\mathcal{E}(R)$ is non-affine. A *linear set* is, by definition, a set which is both 0-valid and affine. It is clear that closure under sums characterizes linear sets. Following this characterization there exist two vectors s_1 and s_2 in $\mathcal{E}(R)$ such that $s_1 \oplus s_2$ does not belong to $\mathcal{E}(R)$. For $i, j = 0, 1$, construct the sets

$$V_{i,j} = \{p/\ 0 \leq p \leq k, s_1(p) = i \text{ and } s_2(p) = j\}$$

and create the simple expression having no constant other than 0

$$h_0 = R(a_1, \ldots a_k), \text{ where for } 1 \leq p \leq k$$

$a_p = 0$ if $p \in V_{0,0}$, $a_p = x$ if $p \in V_{0,1}$, $a_p = y$ if $p \in V_{1,0}$ and $a_p = z$ if $p \in V_{1,1}$.

The set Σ also contains a predicate symbol R' of arity k' such that $\mathcal{E}(R')$ is non-Horn and a predicate symbol R" of arity k" such that $\mathcal{E}(R")$ is non-anti-Horn . Following a well-known characterization of Horn sets [Horn-51] there exist two vectors s'_1 and s'_2 in $\mathcal{E}(R')$ such that $s'_1 \cap s'_2$ does not belong to $\mathcal{E}(R')$. For $i, j = 0, 1$, construct the sets

$$V'_{i,j} = \{p/\ 0 \leq p \leq k', s'_1(p) = i \text{ and } s'_2(p) = j\}$$

and create the simple expression

$$h'_0 = R'(a'_1, \ldots a'_k), \text{ where for } 1 \leq p \leq k$$

$a'_p = 0$ if $p \in V'_{0,0}$, $a'_p = x$ if $p \in V'_{0,1}$, $a'_p = y$ if $p \in V'_{1,0}$ and $a'_p = z$ if $p \in V'_{1,1}$.

Observe that the three variables x, y and z occur in the formula h'_0. By construction $h'_0(x, y, z)$ contains $(1,0,1)$ and $(0,1,1)$, it also contains $(0,0,0)$ for R' is 0-valid, but it does not contain the vector $(0,0,1)$. Using the symbol R" and a similar characterization of anti-Horn sets (stability under the operation $\cup$), one can construct a simple expression h_0" having no constant other than 0 such that the three variables x, y and z occur in h_0" and $h_0"(x, y, z)$ contains the set $\{(0,0,0),(1,0,1),(0,1,1)\}$ but does not contain the vector $(1,1,1)$.

Now, let us consider the following expression:

$$k_0 = h_0 \wedge h_0' \wedge h_0".$$

Whatever the variables effectively occuring in h_0, the three variables x, y and z do occur in k_0. Moreover it is easy to see that $k_0(x, y, z)$ contains the set $\{(0,0,0),(1,0,1),(0,1,1)\}$ but contains neither the vector $(1,1,1)$ (because of h_0") nor $(0,0,1)$ (because of h_0') nor $(1,1,0)$ (because of h_0). We do not know whether it contains $(0,1,0)$ and $(1,0,0)$. There are three cases to distinguish.

- If $k_0(x, y, z)$ contains $(0,1,0)$ then $g_0(u, v) = k_0[0/x, u/z, v/y]$ is suitable.
- If $k_0(x, y, z)$ contains $(1,0,0)$ then $g_0(u, v) = k_0[0/y, u/z, v/x]$ is suitable.
- If $k_0(x, y, z)$ contains neither $(0,1,0)$ nor $(1,0,0)$ then $g_0(u, v, w) = k_0[u/x, v/y, w/z]$ satisfies the second condition of the lemma.

This finishes the proof of the lemma. •

PROOF OF PROPOSITION 4.7: Under the conditions stated in Proposition 4.7, $\mathrm{Sat}_c(\Sigma, \mathcal{E})$ is NP-complete (see Proposition 4.8). Hence, our proof is a reduction from $\mathrm{Sat}_c(\Sigma, \mathcal{E})$ to $\mathrm{Sat}^*(\Sigma, \mathcal{E})$. Let e^{01} be an expression with constants over a set of variables $\mathcal{V}ar(e^{01}) = \{x_1, \ldots, x_n\}$ given as instance for $\mathrm{Sat}_c(\Sigma, \mathcal{E})$. We have to construct, in polynomial time, an expression e (without constants) such that e^{01} is satisfiable iff there is a vector different from $\vec{0}$ and $\vec{1}$ that satisfies e. Let us first introduce two new variables f and t, which will play the role of the constants 0 and 1, and let us consider the expression

$$e^{ft} = e^{01}[f/0, t/1].$$

There are several cases to analyze.

- *Case 1:* There are predicate symbols R and R' in Σ of arity k and k' respectively such that, $\mathcal{E}(R)$ is non-0-valid and $\mathcal{E}(R')$ is non-1-valid.

- *Case 1.a:* There is R" in Σ of arity k" such that $\mathcal{E}(R")$ is non-complementive.
  If $\mathcal{E}(R)$ is 1-valid and $\mathcal{E}(R')$ is 0-valid then the simple expressions

$$h = R(x, \ldots, x) \text{ and } h' = R'(y, \ldots, y)$$

  verify $h(x) = \{1\}$ and $h'(x) = \{0\}$.
  Then, it suffices to consider the following expression

$$e = e^{ft} \wedge h[t/x] \wedge h'[f/y].$$

  Otherwise, $\mathcal{E}(R)$, for instance, is non-0-valid and non-1-valid. Let s be a vector in $\mathcal{E}(R)$. For $i = 0, 1$, construct the sets

$$V_i = \{p/\ 0 \leq p \leq k, s(p) = i\}$$

  and create the simple expression

$$h = R(a_1, \ldots, a_k) \text{ where for } 1 \leq p \leq k$$

$$a_p = x \text{ if } p \in V_0, a_p = y \text{ if } p \in V_1.$$

  By construction $h(x, y) = \{(0, 1)\}$ or $h(x, y) = \{(0, 1), (1, 0)\}$.
  Now, since $\mathcal{E}(R")$ is non-complementive there exists a vector s" in $\mathcal{E}(R")$ such that $\bar{s}" = 1 - s"$ does not belong to $\mathcal{E}(R")$. For $i = 0, 1$, construct the sets

$$V"_i = \{p/\ 0 \leq p \leq k", s"(p) = i\}$$

  and create the simple expression

$$h" = R"(a_1, \ldots, a_{k"}) \text{ where for } 1 \leq p \leq k"$$

$$a_p = x \text{ if } p \in V"_0, a_p = y \text{ if } p \in V"_1.$$

  By construction the expression $g(x, y) = h(x, y) \wedge h"(x, y)$ verifies $g(x, y) = \{(0, 1)\}$.
  Finally, it suffices to consider the following expression

$$e = e^{ft} \wedge g[f/x, t/y].$$

- *Case 1.b:* For every R in Σ, $\mathcal{E}(R)$ is complementive.
  Let s be a vector in $\mathcal{E}(R)$. For $i = 0, 1$, construct the sets

$$V_i = \{p/\ 0 \leq p \leq k, s(p) = i\}$$

and create the simple expression

$$h = R(a_1, \ldots, a_k) \text{ where for } 1 \leq p \leq k$$

$$a_p = x \text{ if } p \in V_0, a_p = y \text{ if } p \in V_1.$$

Observe that $s \neq \vec{0}$ and $s \neq \vec{1}$, hence the two variables x and y occur in h. By construction $h(x, y) = \{(0,1), (1,0)\}$.
Now, let us consider the following expression

$$e = e^{ft} \wedge h[f/x, t/y].$$

Observe that e is only satisfied by $f = 0, t = 1$ or $f = 1, t = 0$, but now both assignments are equally adequate due to the fact that the entire expression is complementive as well.

- *Case 2:* For every R in Σ, $\mathcal{E}(R)$ is 0-valid and there is a predicate symbol R of arity k such that $\mathcal{E}(R)$ is non-complementive.

  - *Case 2.a:* $\mathcal{E}(R)$ is 1-valid.
  Then, there exists a vector s in $\mathcal{E}(R)$ such that $\bar{s} = 1 - s$ does not belong to $\mathcal{E}(R)$. For $i = 0, 1$, construct the sets

  $$V_i = \{p / \ 0 \leq p \leq k, s(p) = i\}$$

  and create the simple expression

  $$h = R(a_1, \ldots, a_k) \text{ where for } 1 \leq p \leq k$$

  $$a_p = x \text{ if } p \in V_0, a_p = y \text{ if } p \in V_1.$$

  By construction $h(x, y) = \{(0,0), (1,1), (0,1)\}$.
  Now, let us consider the following expression

  $$e = e^{ft} \wedge h[f/x, t/y] \wedge \bigwedge_{i=1}^{n} h[x_i/x, t/y] \wedge \bigwedge_{i=1}^{n} h[f/x, x_i/y].$$

  This expression can be interpreted as

  $$e = e^{ft} \wedge (f \longrightarrow t) \wedge \bigwedge_{i=1}^{n} (x_i \longrightarrow t) \wedge \bigwedge_{i=1}^{n} (f \longrightarrow x_i).$$

  Thus it is clear that e^{01} is satisfiable iff there is a vector different from $\vec{0}$ and $\vec{1}$ that satisfies e.

- *Case 2.b:* $\mathcal{E}(R)$ is not 1-valid.
  In this case let us use the expression g_0 defined in Lemma 4.9. Let us consider $g_f = g_0[f/0]$. It is easy to see that if f is evaluated to false then g_f represents the implication and we can proceed as above. But, since $\mathcal{E}(R)$ is 0-valid and non-1-valid, $R(f, \ldots, f)$ is satisfied iff f is evaluated to false. Therefore, according to the number of variables occurring in g_0 it suffices to consider

$$\text{either } e = e^{ft} \wedge R(f, \ldots, f) \wedge \bigwedge_{i=1}^{n} g_f[x_i/u, t/v],$$

$$\text{or } e = e^{ft} \wedge R(f, \ldots, f) \wedge \bigwedge_{i=1}^{n} g_f[x_i/u, x_i'/v, t/w],$$

$$\text{where } x_i', \ i = 1, \ldots, n, \text{ are new variables .}$$

  (Observe that $t = 0$ implies $x_i = x_i' = 0$ for every i, else $x_i' = 1 - x_i$ is suitable.)

- *Case 3:* For every R in Σ $\mathcal{E}(R)$ is 1-valid and there is a predicate symbol R of arity k such that $\mathcal{E}(R)$ is non-complementive.

  - *Case 3.a:* $\mathcal{E}(R)$ is 0-valid.
    Similar to *Case 2.a*.

  - *Case 3.b:* $\mathcal{E}(R)$ is non-0-valid.
    Use the expression g_1 defined in Lemma 4.9 and proceed in the same way as in *Case 2.b*.

- *Case 4:* For every R in Σ, $\mathcal{E}(R)$ is 0-valid, 1-valid and complementive. In this case let use once more the expression g_0 defined in Lemma 4.9. Let us consider $g_f = g_0[f/0]$. Let us recall that if f is evaluated to false then g_f represents the implication. Now, due to the complementiveness, only assignments satisfying $f = 0$ have to be considered. Therefore it suffices to consider the following expression:

$$\text{either } e = e^{ft} \wedge \bigwedge_{i=1}^{n} g_f[x_i/u, t/v],$$

$$\text{or } e = e^{ft} \wedge \bigwedge_{i=1}^{n} g_f[x_i/u, x_i'/v, t/w],$$

$$\text{where } x_i', \ i = 1, \ldots, n, \text{ are new variables .}$$

This completes the proof of Proposition 4.7. •

According to Proposition 3.5 and Proposition 4.7 we can now state our dichotomy theorem.

THEOREM 4.10: *If Σ and $\mathcal{E}$ verify one of the following conditions, then there exists a polynomial-delay algorithm that generates all satisfying vectors of a given expression, otherwise such an algorithm does not exist unless P=NP.*

1. *For every R in Σ, $\mathcal{E}(R)$ is Horn.*
2. *For every R in Σ, $\mathcal{E}(R)$ is anti-Horn.*
3. *For every R in Σ, $\mathcal{E}(R)$ is affine.*
4. *For every R in Σ, $\mathcal{E}(R)$ is bijunctive.*

REFERENCES

1. B. ASPVALL, M. F. PLASS and R. E. TARJAN, A linear-time algorithm for testing the truth of certain quantified boolean formulas, *Information Processing Letters*, 1979, *8*, (3), pp. 121-123.
2. S. A. COOK, The complexity of theorem-proving procedures. In *Third Annual ACM Symposium on Theory of Computing*, 1971, pp. 151-158.
3. N. CREIGNOU and M. HERMANN, Complexity of Generalized Satisfiability Counting Problems, *Information and Computation*, 1996, *125*, (1), pp. 1-12.
4. W. F. DOWLING and J. H. GALLIER, Linear-time algorithms for testing the satisfiability of propositional Horn formulæ, *Journal of Logic Programming*, 1984, *3*, pp. 267-284.
5. M. R. GAREY and D. S. JOHNSON, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Co, 1979.
6. P. HELL and J. NEŠETŘIL, On the complexity of H-coloring, *Journal of Combinatorial Theory, Series B*, 1990, *48*, pp. 92-110.
7. A. HORN, On sentences which are true of direct unions of algebras, *Journal of Symbolic Logic*, 1951, *16*, pp. 14-21.
8. D. S. JOHNSON, M. YANNAKAKIS and C. H. PAPADIMITRIOU, On generating all maximal independent sets, *Information Processing Letters*, 1988, *27*, pp. 119-123.
9. R. E. LADNER, On the structure of polynomial time reducibility, *Journal of the Association for computing Machinery*, 1975, *22*, pp. 155-171.
10. A. K. MACKWORTH, *Constraint Satisfaction*, in S. C. Shapiro, ed., The encyclopedia of Artificial Intelligence, Wiley, New York, 1992, pp. 285-293.
11. T. J. SCHAEFER, The complexity of satisfiability problems. In *Proceedings 10th STOC, San Diego (CA, USA)*, Association for Computing Machinery, 1978, pp. 216-226.
12. L. G. VALIANT, The complexity of enumeration and reliability problems, *SIAM Journal on Computing*, 1979, *8*, (3), pp. 410-421.