Substitutional Definition of Satisfiability in Classical Propositional Logic

Anton Belov and Zbigniew Stachniak*

Department of Computer Science and Engineering,
York University, Toronto, Canada
{antonb, zbigniew}@cs.yorku.ca

Abstract. The syntactic framework of the so-called saturated substitutions is defined and used to obtain new characterizations of SAT as well as the classes of minimal and maximal models of formulas of classical propositional logic.

1 Introduction

The standard two-valued semantics for classical propositional logic can be 'reconstructed' in the logic's proof theory when the logical constants T and F are chosen to represent the truth-values 1 (*true*) and 0 (*false*). In this approach, substitutions of the logical constants T and F for propositional variables are counterparts of truth-value assignments. Furthermore, a formula $\alpha(p_1, \ldots, p_n)$ is satisfiable if and only if there exists a substitution S of T and F for the variables $p_1, \ldots, p_n$ which, when applied to α, transforms this formula into a tautology. Hence, to establish the satisfiability of α, the search for a satisfying truth-value assignment for α (as in WalkSAT procedure, cf. [11]) can be replaced with the search for a 'satisfying' substitution.

In this paper we consider a construction of 'satisfying' substitutions as an alternative to the search. We begin by providing a construction of a certain type of substitutions which we call *saturated substitutions* for propositional formulas. Saturated substitutions define, in a natural way, truth-value assignments. We show that for every formula α, the class of truth-value assignments defined by all saturated substitutions for α coincides with the class of all satisfying truth-value assignments for α (i.e., of all the models of α). Since the construction of saturated substitutions is provided without any explicit reference to semantics, we obtain a syntactic definition of satisfiability.

The second contribution of this paper is a new characterization of minimal and maximal models of propositional formulas. This characterization is given in terms of the so-called *polarized substitutions*. These saturated substitutions define minimal and maximal models in a very special way. Apart from their definitional completeness (polarized substitutions for a formula α define all and only minimal and maximal models of α), these substitutions carry enough semantic information to flag more truth-value assignments as falsifying a formula α than what explicitly follows from the definitions

* Research supported by the grant from the Natural Sciences and Engineering Research Council of Canada.

F. Bacchus and T. Walsh (Eds.): SAT 2005, LNCS 3569, pp. 31–45, 2005.

of a minimal or a maximal model of α. In other words, polarized substitutions carry more semantic information about the space of truth-value assignments of propositional formulas than minimal and maximal models. This property of polarized substitutions could be exploited, for instance, in the design of satisfiability-based problem solving methods that require the construction of multiple or all models of propositional formulas (in areas such as model-based diagnosis, model-based reasoning, or planning, cf. [2,5,6,8,10,12], see also [4]).

This paper is structured as follows. In the next section we introduce and discuss the notions of a saturated and polarized substitutions. In Section 3 we provide a syntactic characterization of SAT in the framework of saturated substitutions. Finally, in Section 4, we characterize the classes of minimal and maximal models of formulas of classical propositional logic in terms of polarized saturated substitutions. For reasons of clarity of presentation, the proofs of all the theoretical results stated in this paper are given in the Appendix.

2 Saturated Substitutions

In this section we define the class of saturated substitutions for propositional formulas. Informally speaking, these substitutions are syntactic counterparts of satisfying truth-value assignments – the relationship that we shall explore in the following sections.

We begin with logical preliminaries. The formulas of classical propositional logic are constructed, in the usual way, in terms of propositional variables, logical connectives (negation $\neg$, disjunction $\vee$, conjunction $\wedge$ and, possibly, other connectives), and the logical constants T (*truth*) and F (*falsehood*). By $Var(\alpha)$ we denote the set of all the variables that occur in α. If $Var(\alpha)$ is of cardinality n, then an enumeration of $Var(\alpha)$ is a bijection p from $\{1, \ldots, n\}$ onto $Var(\alpha)$. If p is an enumeration of $Var(\alpha)$, then we shall frequently denote the $i-th$ variable $p(i)$ by p_i. Finally, we shall write $\alpha(p_1, \ldots, p_k)$ to indicate that $p_1, \ldots, p_k$ are some or all the variables from $Var(\alpha)$.

Given a formula α and an enumeration p of its variables, a substitution is a mapping S that assigns a formula $S(p_i)$ to every variable $p_i \in Var(\alpha)$. We shall frequently represent a substitution S as a finite list of the form $[p_1/S(p_1), \ldots, p_n/S(p_n)]$ which explicitly indicates the assignments of formulas $S(p_i)$ to variables p_i, $1 \leq i \leq n$. If $S = [p_1/\alpha_1, \ldots, p_n/\alpha_n]$ is a substitution, then $S(\alpha)$–the application of S to α–is the formula obtained from α by the simultaneous replacement of every occurrence of every variable p_i with α_i. We shall frequently write $\alpha(p_1/\alpha_1, \ldots, p_n/\alpha_n)$ instead of $S(\alpha)$.

A truth-value assignment is a mapping from the set of all propositional variables into $\{0, 1\}$; its extension to all well-formed formulas of propositional logic is defined in the usual way. A truth-value assignment h satisfies a formula α (or h is a *model* of α), if $h(\alpha) = 1$. On the other hand, if $h(\alpha) = 0$, then α is said to be false under h. A formula is satisfiable if it has a model. We denote by SAT the set of all satisfiable formulas. Finally, if h is a truth-value assignment, q is a propositional variable, and $v \in \{0, 1\}$, then by $h[q/v]$ we denote the truth-value assignment defined exactly like h with the exception that $h[q/v](q) = v$.

As mentioned in the introduction, the satisfiability of a formula α is equivalent to the existence of a substitution S which assigns logical constants T and F to the variables of α in such a way that $S(\alpha)$ is a tautology. For every $\alpha \in SAT$ such a satisfying substitution S (called a saturated substitution in the definition given below) can be constructed using the following lemma.

LEMMA 1. *Let α be a propositional formula, let $q \in Var(\alpha)$, and let h be a model of α. Then:*

(i) h is a model of $\alpha(q/\alpha(q/T))$;
(ii) h is a model of $\alpha(q/\neg\alpha(q/F))$.

If h is a model of $\alpha(p_1, \dots, p_n)$, then, by Lemma 1, it is also a model of $\beta = \alpha(p_1/\gamma, p_2, \dots, p_n)$, where γ is either $\alpha(p_1/T)$ or $\neg\alpha(p_1/F)$. Note that p_1 does not occur in β. By applying Lemma 1 again, this time to β and p_2, we obtain a satisfiable formula with only $p_3, \dots, p_n$ as its variables. It should be evident that $n - 2$ additional applications of Lemma 1 to the remaining variables $p_3, \dots, p_n$ will result in a variable-free formula of the form $\alpha(p_1/\gamma_1, \dots, p_n/\gamma_n)$ from which a required satisfying substitution S can be extracted by making $S(p_i)$ equivalent to $\gamma_i, 1 \leq i \leq n$. This leads us to the following definition.

DEFINITION 1 (Saturated Substitution). *Let α be a formula and suppose that $Var(\alpha)$ is of cardinality n. Furthermore, let p be an enumeration of $Var(\alpha)$ and let τ be a map from $Var(\alpha)$ into $\{0, 1\}$. A* **substitution string** *for α with respect to p and τ is the sequence of substitutions $S_1, \dots, S_n$ defined as follows. Let $S_0(\alpha) \stackrel{def}{=} \alpha$. For every $1 \leq i \leq n$:*

$$S_i(p_i) = \begin{cases} (S_{i-1}(\alpha))(p_i/T), & \text{if } \tau(p_i) = 1, \\ (S_{i-1}(\neg\alpha))(p_i/F), & \text{if } \tau(p_i) = 0. \end{cases}$$

Moreover,

for every $j < i, S_i(p_j) = (S_{i-1}(p_j))(p_i/S_i(p_i))$;
for every $i < j \leq n, S_i(p_j) = p_j$.

The substitution S_n is called a **saturated substitution** *for α.*

The complexity of the definition of a substitution string requires some clarifications. Let us suppose that $S_1, \dots, S_n$ is the substitution string for $\alpha(p_1, \dots, p_n)$ with respect to some enumeration p and a mapping τ. First, let us note that for every $1 \leq i \leq n$, the substitution S_i affects only the first i variables $p_1, \dots p_i$, leaving the remaining variables $p_{i+1}, \dots p_n$ unchanged. In particular, $S_1(p_1)$ is either $\alpha(p_1/T)$ (when $\tau(p_1) = 1$, see Lemma 1(i)) or $\neg\alpha(p/F)$ (when $\tau(p_1) = 0$, see Lemma 1(ii)). For every $j > 1$, $S_1(p_j) = p_j$. Next, let us observe that for every $i > 1$, we first define $S_i(p_i)$ guided by the mapping τ. Only then we proceed to defining $S_i(p_j)$, for every variable p_j such that $j < i$. Hence, $S_2(p_2)$ is either $(S_1(\alpha))(p_2/T)$ (when $\tau(p_2) = 1$) or $(S_1(\neg\alpha))(p_2/F)$ (when $\tau(p_2) = 0$). Having $S_2(p_2)$ defined, we define $S_2(p_1)$ as $(S_1(p_1))(p_2/S_2(p_2))$, i.e., we substitute $S_2(p_2)$ for p_2 in $S_1(p_1)$. Let us look at the following example.

EXAMPLE 1. Let α be $q \vee r$, let $p_1 = q, p_2 = r$, and let τ be defined by: $\tau(p_1) = 1$ and $\tau(p_2) = 0$. The substitution string for α with respect to p and τ is defined as follows:
$S_1(p_1) = (\alpha(p_1, p_2))(p_1/T) = T \vee p_2$;
$S_1(p_2) = p_2$;
$S_2(p_2) = (\neg S_1(p_1 \vee p_2))(p_2/F) = (\neg(T \vee p_2) \vee p_2))(p_2/F) = \neg((T \vee F) \vee F)$ and is equivalent to F;
$S_2(p_1) = (S_1(p_1))(p_2/S_2(p_2)) = (T \vee p_2)(p_2/S_2(p_2)) = T \vee (\neg((T \vee F) \vee F))$ and is equivalent to T.
S_2 is a saturated substitution for α. ■

Note that if S is a saturated substitution for a formula α, then for every variable $q \in Var(\alpha)$, $S(q)$ is equivalent to either T or F, and, hence, so is $S(\alpha)$.

In Section 4, we shall characterize minimal and maximal models of formulas in terms of polarized substitutions defined as follows. Suppose that S is a saturated substitution for a formula α defined with respect to some enumeration p and a map τ. S is said to be *positive*, if $\tau(q) = 1$, for all $q \in Var(\alpha)$. Similarly, S is said to be *negative*, if $\tau(q) = 0$, for all $q \in Var(\alpha)$. Finally, we shall call S a *polarized substitution* for α, if it is either a positive or a negative saturated substitution for α.

EXAMPLE 2. Consider the formula $\alpha = p_1 \vee p_2$ from Example 1. Let $\tau(p_1) = \tau(p_2) = 1$. The following steps define one of the positive saturated substitutions for α:
$S_1(p_1) = (p_1 \vee p_2)(p_1/T) = T \vee p_2$;
$S_1(p_2) = p_2$;
$S_2(p_2) = (S_1(p_1 \vee p_2))(p_2/T) = ((T \vee p_2) \vee p_2)(p_2/T) = (T \vee T) \vee T$ and is equivalent to T;
$S_2(p_1) = (S_1(p_1))(p_2/S_2(p_2)) = (T \vee p_2)(p_2/S_2(p_2)) = T \vee ((T \vee T) \vee T)$ and is equivalent to T. ■

3 Saturated Substitutions and Propositional Satisfiability

In this section we establish a correspondence between saturated substitutions and satisfying truth-value assignments of propositional formulas (cf. Theorem 1).

DEFINITION 2. *Let α be a formula and let S be one of its saturated substitutions. The truth-value assignment h_S is defined in the following way: for every variable $q \in Var(\alpha)$,*

$$h_S(q) = \begin{cases} 1, & \text{if } q \in Var(\alpha) \text{ and } S(q) \equiv T, \\ 0, & \text{if } q \in Var(\alpha) \text{ and } S(q) \equiv F, \\ \text{arbitrary}, & \text{if } q \notin Var(\alpha). \end{cases}$$

THEOREM 1. *Let α be a propositional formula. Then:*

(i) *if $\alpha \in SAT$, then for every saturated substitution S, h_S is a model of α;*
(ii) *for every model h of α there is a saturated substitution S such that $h = h_S$.*

COROLLARY 2. *Let α be a propositional formula and S be one of its saturated substitutions. Then:*

$$\alpha \in SAT \textit{ iff } S(\alpha) \equiv T.$$

In view of Theorem 1, saturated substitutions for a formula α can be considered syntactic counterparts of models of α, that is, models of α can be defined in the syntax of propositional logic in terms of saturated substitutions. Corollary 2 is the characterization of SAT in terms of saturated substitutions: for every saturated substitution S, $S(\alpha)$ is variable free and is equivalent to either T or F; the result of this equivalence test determines the membership in SAT.

4 Minimal and Maximal Models

Minimal and maximal models of propositional formulas are theoretical tools frequently applied in Computer Science in areas such as model-based diagnosis, model-preference default reasoning, or reasoning with models (cf. [5,6,8,10,12]). Maximal and minimal model generation problems have also been widely discussed in the computational complexity literature (cf. [1,7]). In this section we provide a new characterization of these classes of models in terms of polarized substitutions introduced in Section 2. We also prove that polarized substitutions carry, in general, more information about counter-models of propositional formulas than the definitions of minimal and maximal models.

For every propositional formula α we define the relation $\leq_\alpha$ between truth-value assignments in the following way. Given truth-value assignments h_0 and h_1,

$$h_0 \leq_\alpha h_1 \textit{ iff } \text{for every } p \in Var(\alpha), h_0(p) = 1 \textit{ implies } h_1(p) = 1.$$

DEFINITION 3 (Minimal and maximal models). *Let α be a propositional formula. A model h of α is said to be a* **minimal** *(resp. a* **maximal***) model of α, if it is $\leq_\alpha$-minimal (resp. $\leq_\alpha$-maximal).*

THEOREM 3. *Let α be a propositional formula and let h be a truth-value assignment. Then*

(i) *h is a minimal model of α iff $h = h_S$, for some negative saturated substitution S for α;*
(ii) *h is a maximal model of α iff $h = h_S$, for some positive saturated substitution S for α.*

In view of Theorem 3, the definitions of negative (resp. positive) saturated substitutions can be viewed as syntactic definitions of minimal (resp. maximal) models. It turns out that these definitions contain more semantic information about truth-value assignments than merely the fact that they define all the minimal and maximal models of propositional formulas. The following theorem unveils the 'semantic contents' of polarized substitutions.

THEOREM 4 (Blocking Theorem). *Let $\alpha(p_1, \ldots, p_n)$ be a satisfiable formula and let h_{S_+} and h_{S_-} be truth-value assignments defined by a positive and a negative saturated substitutions S_+ and S_-, respectively. Then:*

(i) if for some i, $1 \leq i \leq n$, $h_{S_+}(p_i) = 0$, *then for every choice of truth-values* $v_1, \ldots, v_{i-1}$,

$$h_{S_+}\left[p_1/v_1, \ldots, p_{i-1}/v_{i-1}, p_i/1\right](\alpha) = 0.$$

(ii) if for some i, $1 \leq i \leq n$, $h_{S_-}(p_i) = 1$, *then for every choice of truth-values* $v_1, \ldots, v_{i-1}$,

$$h_{S_-}\left[p_1/v_1, \ldots, p_{i-1}/v_{i-1}, p_i/0\right](\alpha) = 0.$$

When searching for multiple models of a formula α, one can use the definition of a particular maximal or a minimal model of α (given in terms of a polarized substitution) to flag some truth-value assignments as falsifying α. In view of the Blocking Theorem, the definitions of polarized substitutions may prune the space of truth-value assignments far more extensively than the definitions of minimal and maximal models alone. We illustrate this comment with the following example.

EXAMPLE 4. Let α be $\neg p_1 \wedge p_2 \wedge p_3$. Consider the negative saturated substitution S (generated with respect to the enumeration p). The minimal model h_S defined by S is given by:

$$h_S(p_1) = 0, h_S(p_2) = 1, h_S(p_3) = 1.$$

Since h_S is a minimal model, the following three truth-value assignments cannot be models of α:

$$\begin{aligned} h_1(p_1) = 0, h_1(p_2) = 0, h_1(p_3) = 0, \\ h_2(p_1) = 0, h_2(p_2) = 0, h_2(p_3) = 1, \\ h_3(p_1) = 0, h_3(p_2) = 1, h_3(p_3) = 0. \end{aligned}$$

However, the Blocking Theorem flags not only these three but also the additional three assignments as falsifying α. These are:

$$\begin{aligned} h_4(p_1) = 1, h_4(p_2) = 0, h_4(p_3) = 0 \text{ (Theorem 4(ii), } i = 3\text{)}, \\ h_5(p_1) = 1, h_5(p_2) = 1, h_5(p_3) = 0 \text{ (Theorem 4(ii), } i = 3\text{)}, \\ h_6(p_1) = 1, h_6(p_2) = 0, h_6(p_3) = 1 \text{ (Theorem 4(ii), } i = 2\text{)}. \end{aligned}$$

There is one more assignment to consider

$$h_7(p_1) = 1, h_7(p_2) = 1, h_7(p_3) = 1.$$

Neither Theorem 4(ii) nor the definition of a minimal model can flag this assignment as falsifying α. However, it is easy to see that h_S can be also defined by any positive substitution S_+ for α (i.e., $h_S = h_{S_+}$). In other words, h_S is also a maximal model of α (cf. Theorem 3). Hence by either maximality of h_S or by Theorem 4(i) ($i = 1$), h_7 falsifies α. To conclude, by applying the Blocking Theorem to any negative substitution S_- and any positive substitution S_+ for α we were able to demonstrate that h_S is the only model of α. ■

To conclude the discussion on Blocking Theorem, let us note that, in general, different enumerations of $Var(\alpha)$ in Definitions 1 and 2 may result in different polarized substitutions for α. Theorem 4, of course, is applicable to all these substitutions.

5 Conclusions and Future Research

This work is a part of a formal theory of logical substitutions which views saturated substitutions as syntactic counterparts of satisfying truth-value assignments. We show that saturated substitutions can be used to obtain new characterizations of SAT as well as the classes of minimal and maximal models of formulas of classical propositional logic.

If α is a satisfiable formula, then its polarized substitutions define all its maximal and minimal models. If α is not satisfiable, then one can still define this formula's saturated substitutions. If, in addition, α is a conjunction of clauses, then, most likely, the polarized substitutions for α define the solutions to MAX-SAT problem of determining the maximal number of clauses of α that can be simultaneously satisfied by some truth-value assignment. The meaning of a polarized substitution for an unsatisfiable free-form formula is unclear at this point but related to the notion of the cumulative clash measure of a formula introduced in [13,14].

Another line of future research concerns the study of saturated substitutions for the purpose of the development of efficient algorithms for SAT, MAX-SAT, and Maximal (Minimal) Model Generation. This includes the complete and incomplete methods as well as algorithms for SAT based on such techniques as the search in Hamming Balls [3] and Satisfiability Coding Lemma [9].

6 Appendix: Technical Results

This Appendix contains the proofs of all the technical results presented in this paper.

Proof of Lemma 1. Let α, q, and h be as stated. The proof of (i) is the case analysis of the value of $h(\alpha(q/T))$.

- If $h(\alpha(q/T)) = 1$, then $h(\alpha(q/\alpha(q/T))) = h(\alpha(q/T)) = 1$.
- If $h(\alpha(q/T)) = 0$, then $h(q) = 0$ (otherwise $h(\alpha(q/T)) = h(\alpha) = 1$). So $h(\alpha(q/T)) = h(q)$ and $h(\alpha(q/\alpha(q/T))) = h(\alpha) = 1$.

The proof of (ii) is similar.

- If $h(\neg\alpha(q/F)) = 0$, then $h(\alpha(q/\neg\alpha(q/F))) = h(\alpha(q/F)) = 1 - h(\neg\alpha(q/F)) = 1$.
- If $h(\neg\alpha(q/F)) = 1$, then $h(q) = 1$ (otherwise $h(\neg\alpha(q/F)) = h(\neg\alpha) = 0$). So $h(\neg\alpha(q/F)) = h(q)$ and $h(\alpha(q/\neg\alpha(q/F))) = h(\alpha) = 1$. ■

LEMMA 2. *Let $S_1, \ldots, S_n$ be the substitution string for a formula α with respect to an enumeration p and a map τ. Then for every $1 \leq i < n$,*

(i) $S_{i+1}(\alpha) = \big[p_{i+1}/S_{i+1}(p_{i+1})\big] S_i(\alpha)$;
(ii) $S_n(p_i) = \big[p_{i+1}/S_n(p_{i+1}), \ldots, p_n/S_n(p_n)\big] S_i(p_i)$.

Proof. Let α and $S_1, \ldots, S_n$ be as stated.
To show (i) let us note that by the definition of a substitution string we have

$$S_{i+1}(\alpha) = \big[p_1/S_{i+1}(p_1), \ldots, p_i/S_{i+1}(p_i), p_{i+1}/S_{i+1}(p_{i+1})\big](\alpha) \qquad (1)$$

Furthermore, for every $j < i+1$,

$$S_{i+1}(p_j) = \big[p_{i+1}/S_{i+1}(p_{i+1})\big]\, S_i(p_j).$$

We can therefore rewrite (1) as follows

$$\begin{aligned} S_{i+1}(\alpha) &= \big[p_{i+1}/S_{i+1}(p_{i+1})\big]\big[p_1/S_i(p_1),\ldots,p_i/S_i(p_i)\big](\alpha) \\ &= \big[p_{i+1}/S_{i+1}(p_{i+1})\big]\, S_i(\alpha). \end{aligned}$$

We prove (ii) in a similar way. By the definition of a substitution string

$$S_n(p_i) = \big[p_n/S_n(p_n)\big]\big[p_{n-1}/S_{n-1}(p_{n-1})\big]\ldots\big[p_{i+1}/S_{i+1}(p_{i+1})\big](S_i(p_i)) \qquad (2)$$

When computing the value of $S_n(p_i)$ in (2), each variable p_j, $j > i$, will be replaced by

$$\big[p_n/S_n(p_n)\big]\ldots\big[p_{j+1}/S_{j+1}(p_{j+1})\big]\, S_j(p_j),$$

which, by the definition of a substitution string, is $S_n(p_j)$. This justifies (ii). ■

LEMMA 3. *Let $S_1,\ldots,S_n$ be the substitution string for a formula α with respect to an enumeration p and a map τ. Furthermore, let $S'_1,\ldots,S'_{n-1}$ be the substitution string for $S_1(\alpha)$ with respect to the enumeration p' of $Var(S_1(\alpha))$ and the map τ' defined as follows:*

- *for every $1 \le i < n, p'_i = p_{i+1}$,*
- *for every $q \in Var(S_1(\alpha))$, $\tau'(q) = \tau(q)$.*

Then, for every $2 \le i \le n$,

(i) $S_i(p_j) = S'_{i-1}(p_j)$*, for every $2 \le j \le i$;*
(ii) $S_i(\alpha) = S'_{i-1}(S_1(\alpha))$.

Proof. We prove (i) and (ii) by induction on i.
Base case: $i = 2$. From the definition of S_2 we have

$$S_2(p_2) = \begin{cases} (S_1(\alpha))(p_2/T), & \text{if } \tau(p_2) = 1, \\ \neg(S_1(\alpha))(p_2/F), & \text{if } \tau(p_2) = 0. \end{cases}$$

Since $p'_1 = p_2$ and $\tau'(p'_1) = \tau(p_2)$, we have

$$S_2(p_2) = \begin{cases} (S_1(\alpha))(p'_1/T), & \text{if } \tau'(p'_1) = 1 \\ \neg(S_1(\alpha))(p'_1/F), & \text{if } \tau'(p'_1) = 0. \end{cases}$$

This means that $S_2(p_2) = S'_1(p'_1) = S'_1(p_2)$, as required.
To demonstrate (ii), let us note that, by Lemma 2(i),

$$S_2(\alpha) = (S_1(\alpha))(p_2/S_2(p_2)).$$

By (i) and the fact that $p_2 = p'_1$, we conclude that

$$S_2(\alpha) = (S_1(\alpha))(p'_1/S'_1(p'_1)) = S'_1(S_1(\alpha)).$$

Inductive step: Assume now that the lemma holds for $i = m < n$. We shall show that the result also holds for $i = m + 1$.

We prove (i) by, first, showing that the result holds for p_{m+1} and, then, for the remaining variables $p_2, \ldots, p_m$.

By the definition of the substitution string $S_1, \ldots, S_n$, we have

$$S_{m+1}(p_{m+1}) = \begin{cases} (S_m(\alpha))(p_{m+1}/T), & \text{if } \tau(p_{m+1}) = 1, \\ \neg(S_m(\alpha))(p_{m+1}/F), & \text{if } \tau(p_{m+1}) = 0. \end{cases}$$

By the inductive hypothesis, $S_m(\alpha) = S'_{m-1}(S_1(\alpha))$. Since $p_{m+1} = p'_m$ and $\tau(p_{m+1}) = \tau'(p'_m)$, we get

$$S_{m+1}(p_{m+1}) = \begin{cases} (S'_{m-1}(S_1(\alpha)))(p'_m/T), & \text{if } \tau'(p'_m) = 1, \\ \neg(S'_{m-1}(S_1(\alpha)))(p'_m/F), & \text{if } \tau'(p'_m) = 0, \end{cases}$$

that is,

$$S_{m+1}(p_{m+1}) = S'_m(p'_m) = S'_m(p_{m+1}), \tag{3}$$

as required.

Now, consider a variable p_j, $2 \leq j \leq m$. From the definition of substitution string $S_1, \ldots, S_n$, we have

$$S_{m+1}(p_j) = (S_m(p_j))(p_{m+1}/S_{m+1}(p_{m+1})),$$

which, in view of the inductive hypothesis, the equality (3), and the assumption that $p_{m+1} = p'_m$, gives us

$$S_{m+1}(p_j) = (S'_{m-1}(p_j))(p'_m/S'_m(p'_m)).$$

Since, by the assumption of the lemma, $p_j = p'_{j-1}$, we finally obtain

$$S_{m+1}(p_j) = (S'_{m-1}(p'_{j-1}))(p'_m/S'_m(p'_m)) = S'_m(p'_{j-1}) = S'_m(p_j),$$

as required.

To prove (ii), let us note that, by Lemma 2(i),

$$S_{m+1}(\alpha) = S_m(\alpha)(p_{m+1}/S_{m+1}(p_{m+1})),$$

which, by (3) and the assumption that $p_{m+1} = p'_m$, means that

$$S_{m+1}(\alpha) = S_m(\alpha)(p'_m/S'_m(p'_m)).$$

By the inductive hypothesis, $S_m(\alpha) = S'_{m-1}(S_1(\alpha))$. So

$$S_{m+1}(\alpha) = (S'_{m-1}(S_1(\alpha)))(p'_m/S'_m(p'_m)).$$

By Lemma 2(i), the right hand side of this equation equals $S'_m(S_1(\alpha))$, which gives us the desired $S_{m+1}(\alpha) = S'_m(S_1(\alpha))$. ■

Proof of Theorem 1. Let α be an arbitrary propositional formula and $n = |Var(\alpha)|$. To show (i), let h be a model of α and let $S_1, \dots, S_n$ be a substitution string for α with respect to some enumeration p of $Var(\alpha)$ and a map τ. We claim that h_{S_n} is a model of α. We first show that for every $1 \le i \le n$, h is also a model of $S_i(\alpha)$. We proceed by induction on i.

Base case: $i = 1$. If $\tau(p_1) = 1$, then $S_1(p_1) = \alpha(p_1/T)$. Therefore, $S_1(\alpha) = \alpha(p_1/\alpha(p_1/T))$. Similarly, if $\tau(p_1) = 0$, then $S_1(\alpha) = \alpha(p_1/\neg\alpha(p_1/F))$. Thus,

$$S_1(\alpha) = \begin{cases} \alpha(p_1/\alpha(p_1/T)), & \text{if } \tau(p_1) = 1, \\ \alpha(p_1/\neg\alpha(p_1/F)), & \text{if } \tau(p_1) = 0. \end{cases}$$

Since $h(\alpha) = 1$, by Lemma 1 we must have $h(S_1(\alpha)) = 1$.

Inductive step: Assume now that $h(S_k(\alpha)) = 1$, for some $1 \le k < n$. Consider the substitution S_{k+1}. By the definition of a substitution string

$$S_{k+1}(p_{k+1}) = \begin{cases} (S_k(\alpha))(p_{k+1}/T), & \text{if } \tau(p_{k+1}) = 1, \\ \neg(S_k(\alpha))(p_{k+1}/F), & \text{if } \tau(p_{k+1}) = 0. \end{cases}$$

By Lemma 2(i), $S_{k+1}(\alpha) = (S_k(\alpha))(p_{k+1}/S_{k+1}(p_{k+1}))$. So,

$$S_{k+1}(\alpha) = \begin{cases} (S_k(\alpha))(p_{k+1}/(S_k(\alpha))(p_{k+1}/T)), & \text{if } \tau(p_{k+1}) = 1, \\ (S_k(\alpha))(p_{k+1}/\neg(S_k(\alpha))(p_{k+1}/F)), & \text{if } \tau(p_{k+1}) = 0. \end{cases}$$

By induction hypothesis, $h(S_k(\alpha)) = 1$. So, using Lemma 1, we get $h(S_{k+1}(\alpha)) = 1$, which completes the inductive proof.

We have just demonstrated that $h(S_n(\alpha)) = 1$. Since $S_n(\alpha)$ is variable-free,

$$T \equiv S_n(\alpha) = \big[p_1/S_n(p_1), \dots, p_n/S_n(p_n)\big](\alpha) \tag{4}$$

Since all the formulas $S_n(p_i)$, $1 \le i \le n$, are variable-free (and, hence, equivalent to either T or F), (4) implies that h^* defined as follows: for every $1 \le i \le n$,

$$h^*(p_i) = \begin{cases} 1, & \text{if } S_n(p_i) \equiv T, \\ 0, & \text{if } S_n(p_i) \equiv F, \end{cases}$$

is a model of α. But h^* coincides with h_{S_n} on the variables of α. Hence, $h_{S_n}(\alpha) = 1$, which completes the proof of (i).

To show (ii), let h be a model of α, let p be an arbitrary enumeration of $Var(\alpha)$, and let τ be h restricted to $Var(\alpha)$. Let $S_1, \dots, S_n$ be a substitution string for α with respect to p and τ. We claim that h_{S_n} and h coincide on $Var(\alpha)$, or, equivalently, that for any $1 \le i \le n$,

$$S_n(p_i) \equiv \begin{cases} T, & \text{if } h(p_i) = 1, \\ F, & \text{if } h(p_i) = 0. \end{cases} \tag{5}$$

We proceed by induction on n, the number of variables in α.

Base case: Suppose $n = 1$. Let h_0 and h_1 be truth-value assignments such that $h_0(p_1) = 0$ and $h_1(p_1) = 1$.

If $h = h_0$, then $S_1(p_1) = \neg\alpha(p_1/F)$, because $\tau(p_1) = h(p_1) = 0$. Since $h(\alpha) = 1$ and $h(p_1) = 0$ we have $\alpha(p_1/F) \equiv T$ and, hence, $S_1(p_1) = \neg\alpha(p_1/F) \equiv F$, confirming (5). Similarly, if $h = h_1$, then $S_1(p_1) = \alpha(p_1/T)$. Since $h(\alpha) = 1$ and $h(p_1) = 1$, $\alpha(p_1/T) \equiv T$. Thus, $S_1(p_1) = \alpha(p_1/T) \equiv T$, confirming (5).

Inductive step: Assume now that the result holds for any propositional formula of k variables, $k \geq 1$, and that α is a formula of $n = k + 1$ variables. We need to show that for every $1 \leq i \leq k + 1$,

$$S_{k+1}(p_i) \equiv \begin{cases} T, & \text{if } h(p_i) = 1, \\ F, & \text{if } h(p_i) = 0. \end{cases} \tag{6}$$

Consider the substitution S_1. Since $\tau(p_1) = h(p_1)$ and $S_1(p_j) = p_j$, for every $1 < j \leq k + 1$, we have

$$S_1(\alpha) = \begin{cases} \alpha(p_1/\alpha(p_1/T)), & \text{if } h(p_1) = 1, \\ \alpha(p_1/\neg\alpha(p_1/F)), & \text{if } h(p_1) = 0. \end{cases} \tag{7}$$

Since $h(\alpha) = 1$, by Lemma 1 we have $h(S_1(\alpha)) = 1$. Furthermore, $S_1(\alpha)$ is a formula of k variables. Consider the enumeration p' of $Var(S_1(\alpha))$ defined by: for $1 \leq i \leq k$,

$$p'_i = p_{i+1},$$

and the map $\tau' : Var(S_1(\alpha)) \mapsto \{0, 1\}$ defined as follows: for every $1 \leq i \leq k$,

$$\tau'(p'_i) = h(p'_i). \tag{8}$$

By the inductive hypothesis, the substitution S'_k of the string $S'_1, \ldots, S'_k$ for $S_1(\alpha)$ with respect to p' and τ', has the following property: for every $1 \leq i \leq k$,

$$S'_k(p'_i) \equiv \begin{cases} T, & \text{if } h(p'_i) = 1, \\ F, & \text{if } h(p'_i) = 0. \end{cases} \tag{9}$$

It is easy to verify that the substitution strings $S_1, \ldots, S_{k+1}$ and $S'_1, \ldots, S'_k$ satisfy the assumptions of Lemma 3, and, therefore, $S_{k+1}(p_i) = S'_k(p_i)$, for all $2 \leq i \leq k + 1$. Thus, in view of (9), we obtain

$$S_{k+1}(p_i) \equiv \begin{cases} T, & \text{if } h(p_i) = 1, \\ F, & \text{if } h(p_i) = 0, \end{cases} \tag{10}$$

for every $2 \leq i \leq k + 1$.

To complete the the proof it remains to show that

$$S_{k+1}(p_1) \equiv \begin{cases} T, & \text{if } h(p_1) = 1, \\ F, & \text{if } h(p_1) = 0. \end{cases}$$

To this end, let us note that by Lemma 2(ii)

$$S_{k+1}(p_1) = \big[p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](S_1(p_1)).$$

If $h(p_1) = 1$, then $\tau(p_1) = 1$ and $S_1(p_1) = \alpha(p_1/T)$. Hence

$$\begin{aligned} S_{k+1}(p_1) &= \big[p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](\alpha(p_1/T)) \\ &= \big[p_1/T, p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](\alpha) \equiv T. \end{aligned}$$

The last equivalence holds since $h(\alpha) = 1$ and $h(p_1) = 1$. If $h(p_1) = 0$, then the proof is similar. ■

Proof of Theorem 3. Let α be a satisfiable formula of n variables. We shall demonstrate (i) only. The proof of (ii) is similar and is left to the reader.

Let p be some enumeration of $Var(\alpha)$ and let S_+ be a positive saturated substitution for α with respect to p. We must show that h_{S_+} is a maximal model of α.

By Theorem 1, h_{S_+} is a model of α. If $h_{S_+}(q) = 1$, for all $q \in Var(\alpha)$, then h_{S_+} is maximal. Otherwise, the maximality of h_{S_+} follows from Theorem 4(i).

Conversely, suppose that h is a maximal model of α. We shall construct an enumeration p of $Var(\alpha)$ such that for every $1 \leq i \leq n$,

$$S_n(p_i) \equiv \begin{cases} T, & \text{if } h(p_i) = 1, \\ F, & \text{if } h(p_i) = 0, \end{cases} \tag{11}$$

where S_n is the positive saturated substitution for α with respect to p. We proceed by induction on n.

Base case: Suppose that α has just one variable p_1. Let S_1 be the positive saturated substitution for α, that is

$$S_1(p_1) = \alpha(p_1/T). \tag{12}$$

If $h(p_1) = 0$, then $\alpha(p_1/T) \equiv F$, because h is maximal. Hence $S_1(p_1) \equiv F$, as required. If $h(p_1) = 1$, then $\alpha(p_1/T) \equiv T$, because h is a model of α. Hence $S_1(p_1) \equiv T$, as required.

Inductive step: Assume now that (i) holds for any propositional formula of k variables, $k \geq 1$, and that α has $n = k + 1$ variables. We shall construct an enumeration p of $Var(\alpha)$ such that for every $1 \leq i \leq k + 1$,

$$S_{k+1}(p_i) \equiv \begin{cases} T, & \text{if } h(p_i) = 1, \\ F, & \text{if } h(p_i) = 0, \end{cases} \tag{13}$$

where S_{k+1} is the positive saturated substitution for α with respect to p.

Let $q \in Var(\alpha)$ be such that $h(q) = 0$. If there is no such q, then we can repeat the argument presented in the proof of Theorem 1 to show that for every positive saturated substitution S for α, $h = h_S$. Otherwise, consider the formula $\beta = \alpha(q/\alpha(q/T))$. β has k variables and h is also one of its maximal models (this can be demonstrated by induction on the number of variables in β). Thus, by the induction hypothesis, there exists an enumeration p' of $Var(\beta)$, such that for every $1 \leq i \leq k$,

$$S'_k(p'_i) \equiv \begin{cases} T, & \text{if } h(p'_i) = 1, \\ F, & \text{if } h(p'_i) = 0, \end{cases} \tag{14}$$

where S'_k is a positive saturated substitution for β with respect to p'.
Define the enumeration p of $Var(\alpha)$ in the following way:

$$p_i = \begin{cases} q, & \text{if } i = 1, \\ p'_{i-1}, & \text{if } 2 \leq i \leq k+1. \end{cases} \tag{15}$$

Note that the substitution string $S_1, \ldots, S_{k+1}$ for α with respect to p and the substitution string $S'_1, \ldots, S'_k$ for β with respect to p' satisfy the assumptions of Lemma 3. Indeed, since $p_1 = q$, and $\tau(q) = 1$, we have

$$S_1(\alpha) = \alpha(q/\alpha(q/T)) = \beta, \tag{16}$$

and, by (15), $p'_i = p_{i+1}$, for every $1 \leq i \leq k$. The second assumption of Lemma 3 is also satisfied since both strings are positive. So, by Lemma 3(i), for every $2 \leq i \leq k+1$, $S_{k+1}(p_i) = S'_k(p_i)$. Hence, we can use (14) to conclude that (13) holds for every $2 \leq i \leq k+1$.

To complete the proof we only need to show that (13) holds also for $i = 1$ or, equivalently, that $S_{k+1}(p_1) \equiv F$ (since $h(p_1) = h(q) = 0$). To this end, let us note that by Lemma 2(ii),

$$S_{k+1}(p_1) = \big[p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](S_1(p_1)).$$

Since $S_1(p_1) = \alpha(p_1/T)$, this means that

$$S_{k+1}(p_1) = \big[p_1/T, p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](\alpha).$$

By (13), the maximality of h, and by the assumption that $h(q) = 0$, we conclude that $S_{k+1}(p_1) \equiv h[q/T](\alpha) \equiv F$. ■

Proof of Theorem 4. Let α, h_{S_+}, h_{S_-}, S_+, and S_- be as stated. We shall demonstrate (i) only. The proof of (ii) is similar and is left to the reader. We prove (i) by induction on n, the cardinality of $Var(\alpha)$.

Base case: suppose $n = 1$. Let p_1 be the only variable of α and let S_1 be the positive substitution for α. Since $h_{S_1}(p_1) = 0$, we must have $S_1(p_1) \equiv F$. On the other hand, by the definition of a positive substitution, $S_1(p_1) = \alpha(p_1/T)$. Thus, $\alpha(p_1/T) \equiv F$ which shows that $h_{S_1}\big[p_1/1\big](\alpha) = 0$.

Inductive step: Assume now that (i) holds for any formula of k variables, $k \geq 1$, and that α has $n = k+1$ variables. Let us also assume that $S_+ = S_{k+1}$, for the substitution string $S_1, \ldots, S_{k+1}$ defined for some enumeration p of $Var(\alpha)$. Finally, let us assume that $h_{S_{k+1}}(p_i) = 0$, for some $1 \leq i \leq k+1$. We shall consider the cases $i = 1$ and $1 < i \leq k+1$ separately.

If $i = 1$, then $h_{S_{k+1}}(p_1) = 0$ and, hence, $S_{k+1}(p_1) \equiv F$. On the other hand, by Lemma 2(ii),

$$S_{k+1}(p_1) = \big[p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](S_1(p_1)).$$

Since $S_1(p_1) = \alpha(p_1/T)$, we have

$$S_{k+1}(p_1) = \big[p_1/T, p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](\alpha).$$

Thus,

$$\big[p_1/T, p_2/S_{k+1}(p_2), \ldots, p_{k+1}/S_{k+1}(p_{k+1})\big](\alpha) \equiv F,$$

and, by the definition of $h_{S_{k+1}}$, we obtain $h_{S_{k+1}}\big[p_1/1\big](\alpha) = 0$, as required.

Now, suppose that $1 < i \le k+1$. By the definition of a positive saturated substitution, $S_1(p_1) = \alpha(p_1/T)$. Since $S_1(p_j) = p_j$, for the remaining variables of α, we have

$$S_1(\alpha) = \alpha(p_1/S_1(p_1)) = \alpha(p_1/\alpha(p_1/T)). \tag{17}$$

By Lemma 1, $S_1(\alpha)$ is a satisfiable formula of k variables. Let us define the enumeration p' of $Var(S_1(\alpha))$ in the following way: for every $1 \le j \le k$,

$$p'_j = p_{j+1}. \tag{18}$$

Consider the substitution string $S'_1, \ldots, S'_k$ for $S_1(\alpha)$ with respect to p' and the map τ' such that for every variable $q, \tau'(q) = 1$. It is straightforward to verify that the substitution strings $S_1, \ldots, S_{k+1}$ and $S'_1, \ldots, S'_k$ satisfy the assumptions of Lemma 3, and, hence, for every $2 \le j \le k+1$,

$$S_{k+1}(p_j) = S'_k(p_j). \tag{19}$$

Since, by assumption, $h_{S_{k+1}}(p_i) = 0$, $S_{k+1}(p_i)$ must be equivalent to F. By (19), this means that $S'_k(p_i) \equiv F$ or that $h_{S'_k}(p_i) = 0$. If we put $w = i - 1$, then, in view of (18), we get $h_{S'_k}(p'_w) = 0$.

To summarize, $S_1(\alpha)$ satisfies the assumptions of the inductive hypothesis with respect to $h_{S'_k}$. Hence, for every choice of truth-values $v_1, \ldots, v_{w-1}$,

$$h_{S'_k}\big[p'_1/v_1, \ldots, p'_{w-1}/v_{w-1}, p'_w/1\big](S_1(\alpha)) = 0. \tag{20}$$

By the definition of $h_{S'_k}$, by (18) and (19), for every $1 \le j \le k$,

$$h_{S'_k}(p_{j+1}) = \begin{cases} 1, & \text{if } S_{k+1}(p_{j+1}) \equiv T, \\ 0, & \text{if } S_{k+1}(p_{j+1}) \equiv F. \end{cases}$$

In other words, for every $2 \le j \le k+1$, $h_{S'_k}(p_i) = h_{S_{k+1}}(p_i)$. From this and the fact that $p_1 \notin Var(S_1(\alpha))$ it follows that (20) can be rewritten as $h_{S_{k+1}}\big[p_2/v_1, \ldots, p_{i-1}/v_{w-1}, p_i/1\big](S_1(\alpha)) = 0$, and, further, using (17), as

$$h_{S_{k+1}}\big[p_2/v_1, \ldots, p_{i-1}/v_{w-1}, p_i/1\big](\alpha(p_1/\alpha(p_1/T))) = 0.$$

By Lemma 1, this implies that neither

$$h_{S_{k+1}}\big[p_2/v_1, \ldots, p_{i-1}/v_{w-1}, p_i/1\big]$$

nor

$$h_{S_{k+1}}\big[p_1/1 - h_{S_{k+1}}(p_1), p_2/v_1, \ldots, p_{i-1}/v_{w-1}, p_i/1\big]$$

are models of α. This completes the proof of (i). ■

References

[1] Z. Chen and S. Toda. The complexity of selecting maximal solutions. *Information and Computation* 119 (1995), pp. 231–239.

[2] H. Kautz and B. Selman. Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search. *Proc. of AAAI'96* (1996).

[3] E. Dantsin, E.A. Hirsch, and A. Wolper. Algorithms for SAT based on Search in Hamming Balls. *Proc. of STACS 2004, LNCS* 2996 (2004), pp. 141–151.

[4] D.J. Kavvadias and E.C. Stavropoulos. An algorithm for generating all maximal models of Boolean expressions. *Proc. of the 4th Panhellenic Logic Symposium* (2003), pp. 125-129.

[5] R. Khardon and D. Roth. Reasoning with Models. *Artificial Intelligence* 87 (1996), pp. 187-213.

[6] R. Khardon and D. Roth. Default-reasoning with Models. *Proc. of IJCAI'95* (1995), pp. 319–325.

[7] L.M. Kirousis and P.G. Kolaitis. The complexity of minimal satisfiability problems. *Proc. of STACS 2001, LNCS* 2010 (2001), pp. 407–418.

[8] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence* 56 (1992), pp. 197–222.

[9] R. Paturi, P. Pudlák, and F. Zane. Satisfiability Coding Lemma. *Proc. of FOCS'97* (1997), pp. 566–574.

[10] B. Selman, H.A. Kautz. Model-preference default theories. *Artificial Intelligence* 45 (1990), pp. 287–322.

[11] B. Selman, H. Kautz, and B. Cohen. Noise Strategies for Local Search. *Proc. of AAAI'94* (1994), pp. 337–343.

[12] A. Smith, A. Veneris and A. Viglas. Design Diagnosis Using Boolean Satisfiability. *IEEE Asian-South Pacific Design Automation Conference* (2004).

[13] Z. Stachniak. Exploiting Polarity in Multiple-Valued Inference Systems. *Proc. of the 31st IEEE Int. Symp. on Multiple-Valued Logic* (2001), pp. 149–156.

[14] Z. Stachniak. Going Non-clausal. *Proc. of the Fifth International Symposium on the Theory and Applications of Satisfiability Testing, SAT 2002* (2002), pp. 316–322.