Journal of Automated Reasoning **6**: 173–187, 1990.

The Automation of Syllogistic

II. *Optimization and Complexity Issues*★

D. CANTONE
Courant Institute of Mathematical Sciences, New York University, U.S.A. and University of Catania, Italy

E. G. OMODEO
ENIDATA, Bologna, Italy

and

A. POLICRITI
Courant Institute of Mathematical Sciences, New York University, New York, U.S.A.

(Received: 20 October 1988; revised: 17 May 1989)

Abstract. In the first paper of this series it was shown that any unquantified formula p in the collection MLSSF (multilevel syllogistic extended with the singleton operator and the predicate *Finite*) can be decomposed as a disjunction of set-theoretic formulae called *syllogistic schemes*. The syllogistic schemes are satisfiable and no two of them have a model in common, therefore the previous result already implied the decidability of the class MLSSF by simply checking if the set of syllogistic schemes associated with the given formula is empty.

In the first section of this paper a new and improved searching algorithm for syllogistic schemes is introduced, based on a proof of existence of a 'minimum effort' scheme for any given satisfiable formula in MLSF. The algorithm addressed above can be piloted quite effectively even though it involves backtracking.

In the second part of the paper, complexity issues are studied by showing that the class of $(\forall)^{l}_{0}$-*simple prenex* formulae (an extension of MLS) has a decision problem which is NP-complete. The decision algorithm that proves the membership of this decision problem to NP can be seen as a different decision algorithm for MLS.

Key words. Decision procedures, set theory, syllogistic schemes, NP-completeness, model graphs.

> Beyond that, and much more difficult still,
> is the problem of handling the membership
> relation in an efficient way, so that
> theorem-proving problems involving
> set theoretic notions can be treated.
>
> (J. A. Robinson, 1967)

1. Introduction

In the first paper of this series [4] a family Σ of set-theoretic formulae, called *syllogistic schemes related to X*, was introduced for any given finite collection X of

★ Research supported by ENI and ENIDATA within the AXL project.

set-variables. It was proved that every syllogistic scheme is satisfiable, that is, it can be made true by suitably substituting sets for the variables occurring in it. Moreover no two syllogistic schemes related to the same X in $\Sigma(X)$ have any model in common.

It was shown in the same article that any unquantified formula p in the collection MLSSF (see below) can be decomposed as a disjunction of syllogistic schemes. A naive algorithm for determining whether or not a given p in F is satisfiable is to calculate the set Σ_p of all disjuncts of p' and then check whether this set is nonempty. A technique was established for extracting the schemes that form Σ_p from the collection $\Sigma(X)$ of all syllogistic schemes over the variables X of p. In other words, this is a technique for evaluating p in any $\sigma \in \Sigma(X)$; in fact, σ will belong to Σ_p if and only if the result of the evaluation is *TRUE*.

In the first part of this paper we improve the decision algorithm outlined above, restricting our attention to formulae in MLSF. These formulae involve, in addition to $\emptyset$ and set-variables, the binary constructs $\cap$, $\setminus$, $\cup$, $\in$, $\subseteq$, $=$, the unary relator *Finite*, and the propositional connectives. Therefore, the only construct of MLSSF not taken into account in this paper is $\{\bullet, \ldots, \bullet\}$. We will be able to characterize the schemes σ in Σ_p in such a way that Σ_p can be generated *directly*, instead of being obtained by filtering out the schemes in $\Sigma(X)$ that do not satisfy p. We also prove that if Σ_p is nonempty, then it contains 'minimum effort' schemes (in a sense to be explained in the next paragraph - and, in more detail, in the next section). Thus in order to check p for satisfiability only such schemes need to be sought.

To be more specific even at this informal level of discussion, we anticipate that every syllogistic scheme σ in $\Sigma(X)$ is identified by a quadruple $(\sim, G, F, Z)$ where $\sim$ is an equivalence relation over X and G is a directed acyclic graph (or DAG) whose nodes are the $\sim$-classes. We will see that any σ in Σ_p corresponds to an equivalence relation $\sim$ of a particular kind, which will be called a *p-compatible relation*, and a DAG of a particular kind, to be called a *p-$\sim$-compatible DAG*. If Σ_p is non-empty, then a finest p-compatible relation $\sim_p$ exists (this would no longer be the case if the construct $\{\bullet, \ldots, \bullet\}$ was admitted). Moreover this relation, which can be determined by methods of propositional calculus (see [6]), provably has some σ in Σ_p associated with it. Any σ associated with $\sim_p$ is what we have called above a minimum effort syllogistic scheme; establishing the existence of one such scheme amounts to searching for a p-$\sim_p$-compatible DAG. This search, although it involves backtracking, can be piloted quite effectively (see [5, 7]).

In the second part of this paper, complexity issues on the decision problem for classes of set-theoretic formulae are studied. In particular the class of $(\forall)_0$-*simple prenex* formulae (an extension of MLSS) is introduced and is proved to be NP-complete, when the number of universal quantifiers in every prefix conjunct is bounded. The decision algorithm presented in the second part of the paper differs from the one presented in the first part and provides another possible approach to the decision problem for extensions of MLS.

The second algorithm also associates a graph with the formula to be decided. The graph keeps information about a possible model of the formula in two ways: by using the nodes and the edges of the graph to represent the sets in the model, some of their elements and the membership relation among them, and by defining a map from the graph onto the hereditarily finite sets which is obtained by induction and preserves all the properties of the model expressible by $(\forall)_0$-simple prenex formulae.

It turns out that, when the formula is satisfiable, the algorithm can run in polynomial time on a nondeterministic Turing machine.

Refined versions of this algorithm are presented in [9] to prove the decidability of an extension of the class of $(\forall)_0$-simple prenex formulae and to study the decision problem of classes of formulae involving a choice operator.

2. A Decision Algorithm for MLSF

The constituents of the theory MLSF are:

- a denumerable infinity of set variables $x, y, z, \ldots$;
- the operators $\cup$ (binary union), $\cap$ (binary intersection), $\setminus$ (set difference);
- the predicates $=$ (equality), $\in$ (membership), *Finite* (finiteness), $\subseteq$ (set-inclusion);
- the boolean connectives $\neg$, &, $\vee$, $\rightarrow$, $\leftrightarrow$.

By a simple normalization process, the satisfiability problem for MLSF can be reduced to the problem of testing for satisfiability conjunctions of literals of the following types:

$$\begin{array}{ll}
(=) & x = y, x = y \cup z, x = y \setminus z, \\
(\neq) & x \neq y, \\
(\in, \notin) & x \in y, x \notin y, \\
(F, \not{F}) & \mathit{Finite}\ x, \neg \mathit{Finite}\ x,
\end{array}$$

where each x, y, z is a set variable.

Let p be such a conjunction, and let X be the set of all variables appearing in p. We denote by $p_=$ the conjunction of the literals of type $(=)$ belonging to p. Moreover, p_* will stand for the propositional formula obtained from $p_=$ by replacing the symbols $=$, $\cup$, and $\setminus$ by $\leftrightarrow$, $\vee$, and & $\neg$ respectively, and accordingly regarding each set variable as a propositional variable. (Notice that in our use of the word 'propositional' we are conforming here to a well-established tradition – see, e.g., [6]. In particular, by 'propositional variable' we mean a variable that ranges over the truth values *FALSE*, *TRUE*.)

In the following definition of a *p-compatible* equivalence relation $\sim$ over X, our aim is to capture the properties that $\sim$ must enjoy in order that it can be induced by a model M of p, in the sense that $x \sim y$ if and only if $x = y$ is true in M, for all x, y in X. It will be obvious that there exists an algorithm for establishing whether any given $\sim$ is p-compatible or not.

DEFINITION 1. Let $\sim$ be an equivalence relation over the set X of all variables that occur in a given conjunction p of literals $(=)$, $(\neq)$, $(\in, \notin)$ and $(F, \not{F})$; moreover, let $p_{\sim}$ denote the propositional formula p_{*} & $\&_{x \sim y}(x \leftrightarrow y)$.

An *acceptable place* of the pair p, $\sim$ is any set $P \subseteq X$ whose characteristic function χ_P is a model of $p_{\sim}$, where

$$\chi_P(x) = \begin{cases} \text{TRUE} & \text{if } x \in P \\ \text{FALSE} & \text{if } x \in X \backslash P. \end{cases}$$

By abuse of language, we will often say that a place satisfies a given propositional formula to actually mean that its characteristic function satisfies it.

An equivalence relation $\sim$ over X is said to be *p-compatible* iff the following conditions are met:

(a) $v \sim w$ if (and only if) $v \leftrightarrow w$ is satisfied by every acceptable place of p, $\sim$;
(b) if $v \neq w$ belongs to p, then $v \nsim w$;
(c) if $v \in z_0$, $w \notin z_1$ belong to p and $z_0 \sim z_1$, then $v \nsim w$;
(d) if $\neg$*Finite* v belongs to p then there must exist an acceptable place P_v such that $v \in P_v$ and P_v does not contain any variable w for which *Finite* w belongs to p.

The following lemma shows that any satisfiable conjunction p admits p-compatible equivalence relations, induced by the models of p:

LEMMA 1. *Let M be a model of p, and let $x \sim_M y$ if and only if $x^M = y^M$. Then*

(I) *$P_\xi = \{x \in X\colon \xi \in x^M\}$ is an acceptable place of p, $\sim_M$, where $\xi \in \bigcup_{x \in X} x^M$;*
(II) *$\sim_M$ is p-compatible.*

Proof. Let $\xi \in \bigcup_{x \in X} x^M$. Clearly if $x \sim_M y$, then $\chi_{P_\xi}(x) = \chi_{P_\xi}(y)$, i.e., $(x \leftrightarrow y)^{\chi_{P_\xi}} = \mathit{TRUE}$. Moreover if $x \leftrightarrow (y \vee z)$ is in p_{*}, then $x = y \cup z$ is in p and $x^M = y^M \cup z^M$. Hence $\xi \in x^M$ if and only if either $\xi \in y^M$ or $\xi \in z^M$, i.e. $\chi_{P_\xi}(x) = \chi_{P_\xi}(y) \vee \chi_{P_\xi}(z)$, proving that χ_{P_ξ} is a model for $x \leftrightarrow y \vee z$. Analogously it can be shown that χ_{P_ξ} satisfies all remaining conjuncts in $p_{\sim}$, which proves that χ_{P_ξ} is an acceptable place of p, $\sim_M$, establishing (I). Note in addition that χ_{P_ξ} also satisfies all formulae $\neg(v \leftrightarrow w)$ for which $\xi \in (v^M \backslash w^M) \cup (w^M \backslash v^M)$.

Next we show that $\sim_M$ is p-compatible. First of all, it is clear that $\sim_M$ is an equivalance relation. If $v \nsim_M w$, for any two variables v, w occurring in p, then $v^M \neq w^M$, so that there exists $\xi \in (v^M \backslash w^M) \cup (w^M \backslash v^M)$. Therefore $(v \leftrightarrow w)^{\chi_{P_\xi}} = \mathit{FALSE}$, which proves that (a) of Definition 1 holds.

(b) is an immediate consequence of the very definition of $\sim_M$, in view of the fact that M satisfies p.

If $v \in z_0$, $w \notin z_1$ belong to p and $z_0 \sim_M z_1$, then $v^M \in z_0^M$, $w^M \notin z_1^M$, and $z_0^M = z_1^M$, which imply $v^M \neq w^M$ and in turn $v \nsim_M w$. Having thus shown that (c) of Definition 1 holds too, we now proceed to prove the only remaining condition, (d). Let us assume that $\neg$*Finite* v belongs to p. Hence v^M is infinite. By (I), P_ξ is an acceptable place of p, $\sim_M$ for every ξ in v^M; moreover, since obviously there are only finitely many acceptable places of p, $\sim_M$, it follows that there must exist an acceptable place P such that $\{\xi \in v^M: P_\xi = P\}$ is infinite. So, if $w \in P$, then $\{\xi \in v^M: P_\xi = P\} \subseteq w^M$, i.e., w^M is infinite, and hence *Finite* w cannot belong to p. This completes the proof of the p-compatibility of $\sim_M$, establishing the lemma. ■

An important equivalence relation is introduced by the following definition.

DEFINITION 2. Given p as above, for any x and y occurring in p we put

$$x \sim_p y \text{ if and only if } p_* \to (x \leftrightarrow y) \text{ is tautological,}$$

where p_* is obtained from p as in Definition 1.

LEMMA 2. *$\sim_p$ is an equivalence relation.*

Proof. The lemma follows at once from the reflexivity, symmetry and transitivity of $\leftrightarrow$. ■

An important property of the equivalence relation $\sim_p$ is stated in the following lemma.

LEMMA 3. *Let $\sim$ be a p-compatible equivalence relation. Then*

(i) if $x \sim_p y$ then $x \sim y$, for x, y in X;

(ii) $\sim_p$ is p-compatible.

(In other words, $\sim_p$ is the finest p-compatible equivalence relation over the set X of variables occurring in p.)

Proof. Let $x \sim_p y$. Then $p_* \to (x \leftrightarrow y)$ is a tautology. If P is an acceptable place of p, $\sim$, then χ_P satisfies $p_\sim$, and in particular p_*. Hence $(x \leftrightarrow y)^{\chi_P} = TRUE$. This shows that $x \leftrightarrow y$ is satisfied by all acceptable places of p, $\sim$. Therefore the p-compatibility of $\sim$ implies $x \sim y$, establishing (i).

In order to prove that $\sim_p$ is p-compatible, we show that conditions (a)–(d) of Definition 1 are met for $\sim_p$. Observe that the acceptable places of p, $\sim_p$ are those sets P such that χ_P satisfies p_*. Therefore, if $v \leftrightarrow w$ is satisfied by every acceptable place of p, $\sim_p$, it follows that $p_* \to (v \leftrightarrow w)$ is a tautology, i.e. $v \sim_p w$, proving (a).

To establish (b), assume that $v \neq w$ occurs in p. If $v \sim_p w$, then by (i) above $v \sim w$, contradicting the p-compatibility of $\sim$. Thus $v \nsim_p w$, and (b) is proved.

Next suppose that $z_0 \sim_p z_1$, and assume also that $v \in z_0$ and $w \notin z_1$ occur in p. Then again by (i) $z_0 \sim z_1$, implying $v \nsim w$. Thus, as above, $v \nsim_p w$, concluding the verification of (c).

Finally, let $\neg$*Finite* v occur in p. From the p-compatibility of $\sim$ it follows that there exists an acceptable place P_v of p, $\sim$ such that $v \in P_v$ and P_v does not contain any variable w for which *Finite* w occurs in p. Then it is enough to observe that P_v is also an acceptable place of p, $\sim_p$. This shows that (d) of Definition 1 is also fulfilled, which in turn establishes (ii) of the present lemma. ■

An immediate consequence of the preceding lemma is given in the following corollary.

COROLLARY 1. *If p is satisfiable, then $\sim_p$ is p-compatible.*

Proof. Let M be a model of p. Lemma 1 shows that $\sim_M$ is p-compatible, and therefore Lemma 3 entails the p-compatibility of $\sim_p$. ■

The preceding corollary says that the p-compatibility of $\sim_p$ is a necessary (algorithmically verifiable) condition for p to be satisfiable. In what follows, we will bring to light another necessary condition for the satisfiability of p. Taken together with the p-compatibility of $\sim_p$, the latter condition will suffice to imply that p admits a model M_p, such that $\sim_p$ coincides with $\sim_{M_p}$ (cf. Lemma 1).

In view of Lemmas 1(II), 3(i), this will entail that when p is satisfiable then any two variables x, y of p are $\sim_{Mp}$-equivalent if and only if $x \sim_M y$ (i.e., the set-values of x and y coincide) in every model M of p.

We incidentally note that the existence of a finest equivalence $\sim_M$, with M a model of p, is no longer insured if one adds new operators to those admitted in MLSF. For example, the models of the formula

$$z = z \backslash z \;\&\; s = \{x\} \;\&\; y \backslash s = z$$

of MLSSF can be grouped into two classes, with

$$y \sim_{M_0} z, \quad y \not\sim_{M_0} s,$$
$$y \not\sim_{M_1} z, \quad y \sim_{M_1} s$$

(and hence

$$y \sim_{M_0} z \text{ not implying } z \sim_{M_1} z,$$
$$y \sim_{M_1} z \text{ not implying } y \sim_{M_0} z)$$

whenever M_0 belongs to the first class and M_1 belongs to the second. Other formulae with similar pathology are $y \in \{a, b\} \;\&\; a \neq b$ and $x, y, z \in \{a, b\}$.

The above discussion indicates that $\sim_p$ is mainly related to the equalities holding in a sought model of p. To fully take into account the membership literals as well, we introduce the *p-$\sim$-compatible directed acyclic graphs.*

DEFINITION 3. Let $\sim$ be a p-compatible equivalence relation and let S be the collection of representatives of the variables occurring in p (for example, if we assume that all variables of set theory are arranged in a denumerable sequence, then we can choose the first variable in each equivalence class C as the representative of C). A DAG $(S, \dot{})$ is said to be *p-$\sim$-compatible* iff

(a) for each v in S there is an acceptable place P_v of p, $\sim$ such that $\dot{v} = P_v \cap S$;
(b) if $v \in w$ [respectively: $v \notin w$] occurs in p, then $s_w \in \dot{s}_v$, [resp.: $s_w \notin \dot{s}_v$] where $s_v, s_w \in S$ and $s_v \sim v$, $s_w \sim w$.

Later in this section we will show that a normalized conjunction p of MLSF has a model if and only if $\sim_p$ is p-compatible and there exists a p-$\sim_p$-compatible DAG $(S_p, \dot{})$, where S_p is the set of all $\sim_p$-representatives. This provides at once a satisfiability algorithm (alternative to the one described in Corollary 1) for MLSF:

in fact, the number of possible DAGs $(S_p, {}^{\cdot})$ is clearly finite and, moreover, there is an algorithm to test whether a given DAG is p-$\sim_p$-compatible.

A first step towards proving the completeness of the above test is the following lemma.

LEMMA 4. *If p is satisfiable, then there exists a p-compatible equivalence relation $\sim$ which admits a p-$\sim$-compatible DAG.*

Proof. Let M be a model of p. Consider the relation $\sim_M$ defined by putting $x \sim_M y$ if and only if $x^M = y^M$. (II) of Lemma 1 shows that $\sim_M$ is p-compatible. It only remains to show that the set S_M of $\sim_M$-representatives of the variables of p can be given the structure of a p-$\sim_M$-compatible DAG. To this end we put

$$\dot{v} = \{w \in S_M \colon v^M \in w^M\} \quad \text{for every } v \text{ in } S_M,$$

so that $\dot{v} = \{x \in X \colon v^M \in x^M\} \cap S_M$, where X denotes the collection of all variables occurring in p. Since by Lemma 1(I) the set $\{x \in X \colon v^M \in x^M\}$ is an acceptable place of p, $\sim_M$, (a) of Definition 3 is satisfied. Observing that (b) of the same definition is an immediate consequence of the fact that M is a model of p, it follows that $(S_M, {}^{\cdot})$ is a p-$\sim_M$-compatible DAG, which proves the lemma. ■

LEMMA 5. *Let p be a normalized conjunction of MLSF. If there exists a p-compatible equivalence relation $\sim$ admitting a p-$\sim$-compatible DAG, then $\sim_p$* (cf. Definition 2) *is also p-compatible and admits a p-$\sim_p$-compatible DAG.*

Proof. Lemma 3 ensures the p-compatibility of $\sim_p$. Next, let $(S_\sim, {}^{\cdot})$ be a p-$\sim$-compatible DAG. We will show how the set S_p of $\sim_p$-representatives of the variables occurring in p can be given a structure of p-$\sim_p$-compatible DAG.

For each v in S_p let $v_* \in S_\sim$ be such that $v \sim v_*$ and put $\overset{*}{v} = \{w \in S_p \colon w_* \in \dot{v}_*\}$. We must simply show that $(S_p, {}^*)$ is a p-$\sim_p$-compatible DAG. Since $(S_\sim, {}^{\cdot})$ is a DAG, so is $(S_p, {}^*)$. The p-$\sim$-compatibility of $(S_p, {}^{\cdot})$ can be proved as follows. Let $v \in S_p$. As $(S_\sim, {}^{\cdot})$ is p-$\sim$-compatible, $\dot{v}_* = P \cap S_\sim$, for some acceptable place P of p, $\sim$. We have $\overset{*}{v} = P \cap S_p$. Indeed, if $w \in \overset{*}{v}$ then $w_* \in \dot{v}_* = P \cap S_\sim$. In particular $w_* \in P$, and since $w \sim w_*$ we have $w \in P$.

Therefore $w \in P \cap S_p$. Conversely, if $w \in P \cap S_p$, from the fact that P is an acceptable place of p, $\sim$ it follows $w_* \in P \cap S_\sim = \dot{v}_*$, so that $w \in \overset{*}{v}$. Hence (a) of Definition 3 is proved for $(S_p, {}^*)$.

Assume now that $S_\sim \ni s_v \sim v \sim_p g_v \in S_p$ and that $S_\sim \ni s_w \sim w \sim_p g_w \in S_p$. Lemma 3(i) implies that $g_v \sim s_v$ and $g_w \sim s_w$, so that $\overset{*}{g}_v = \{x \in S_p \colon x_* \in \dot{s}_v\}$. If $v \in w$ occurs in p, then by the p-$\sim$-compatibility of $(S_\sim, {}^{\cdot})$ we have $s_w \in \dot{s}_v$ showing $g_w \in \overset{*}{g}_v$. If instead $v \notin w$ is a literal of p, then $g_w \notin \overset{*}{g}_v$, because otherwise $(g_w)_* \in \dot{s}_v$, i.e., $s_w \in \dot{s}_v$, which contradicts the hypothesis of p-$\sim$-compatibility of $(S_\sim, {}^{\cdot})$. This completes the proof that $(S_p, {}^*)$ is p-$\sim_p$-compatible, and the lemma is established. ■

From the preceding two lemmas the following corollary is immediately derived.

COROLLARY 2. (Completeness). *Let p be a normalized conjunction of MLSF. If p is satisfiable then $\sim_p$ is p-compatible and it admits a p-$\sim_p$-compatible DAG.*

Moreover, there is an algorithm to test whether $\sim_p$ is p-compatible and it admits a p-$\sim_p$-compatible DAG.

In order to show the soundness of the test given in the preceding corollary we recall the definition and some properties of syllogistic schemes (cf. [4]).

DEFINITION 4. Let $X = \{x_0, x_1, \ldots, x_I\}$ and $Y = \{y_0, y_1, \ldots, y_J\}$ be collections of set variables such that

(i) X and Y have no variables in common;
(ii) there is a bijection °: $Y \to Pow(X) \setminus \{\emptyset\}$ (so that $|Y| = 2^{|X|} - 1$).

Moreover, let $\sim$ be an equivalence relation over X whose equivalence classes are

$$\{x_{00}, x_{01}, \ldots, x_{0L_0}\}, \{x_{10}, x_{11}, \ldots, x_{1L_1}\}, \ldots, \{x_{N0}, x_{N1}, \ldots, x_{NL_N}\},$$

with $x_{n0} \prec x_{n1} \prec \cdots \prec x_{nL_n}$ for $n = 0, 1, \ldots, N$, and where we assume $x_{00} \prec x_{10} \prec \cdots \prec x_{N0}$ ($\prec$ is a fixed ordering of the collection of all variables). Let us put $s_n = x_{n0}$ for $n = 0, 1, \ldots, N$ and let $S = \{s_0, s_1, \ldots, s_N\}$ be the set of representatives of the equivalence classes of $\sim$. Also suppose that S is given a DAG structure $(S, {}^\bullet)$. Let $F \subseteq Z \subseteq Y$ be such that

(a) $\mathring{z} \subseteq S$, for all z in Z;
(b) there are no distinct v, w in S such that
  (b1) $v \in \dot{s}$ if and only if $w \in \dot{s}$, for all s in S, and
  (b2) $v \in \mathring{z}$ if and only if $w \in \mathring{z}$, for all z in Z
  hold together.

Let δ, ρ_{ZF} denote the formulae

$$\delta \;=_{\mathrm{Def}} \&_{j \leqslant J}(y_j \neq \emptyset \;\&\; \&_{i \leqslant I}\, x_i \notin y_j \;\&\; \&_{j < k \leqslant J}\, y_j \cap y_k = \emptyset),$$

$$\rho_{ZF} \;=_{\mathrm{Def}} \&_{n \leqslant N}\left(s_n = x_{n1} = \cdots = x_{nL_n} = \bigcup_{\substack{z \in Z \\ s_n \in \mathring{z}}} z \cup \bigcup_{\substack{r \in S \\ s_n \in \dot{r}}} \{r\}\right) \&$$
$$(\&_{f \in F}\, Finite(f)) \;\&\; (\&_{z \in Z \setminus F}\, \neg Finite(z)).$$

We put $\sigma_{ZF} =_{\mathrm{Def}} \delta \;\&\; \rho_{ZF}$ and call σ_{ZF} a *syllogistic scheme* over X (relative to $X, \sim, {}^\bullet$).

In [4] it is shown that syllogistic schemes partition the class of all possible assignments over X. Other important properties of syllogistic schemes are stated in the following proposition (for a proof see [4]).

PROPOSITION 1. *Let σ_{ZF} be a syllogistic scheme relative to $X, \sim, {}^\bullet$. The following properties hold*:

(a) *σ_{ZF} is solvable*;
(b) *for any solution M of σ_{ZF} we have*:
  (b1) *for all x, y in X, $x^M = y^M$ iff $x \sim y$,*
  (b2) *for all s_0, s_1 in X, $s_0^M \in s_1^M$ iff $s_1 \in \dot{s}_0$,*
  (b3) *for all s in S, s^M is infinite iff $s \in \mathring{z}$ for some z in $Z \setminus F$.*

The following lemma gives the soundness of the satisfiability test whose completeness has already been established in Corollary 2.

LEMMA 6 (Soundness). *Let p be a normalized conjunction of MLSF and let X be the collection of all variables occurring in p. If there exists a p-compatible equivalence relation $\sim$ which admits a p-$\sim$-compatible DAG $(S, {}^{\bullet})$, then p is satisfiable*

Proof. We will show that from the hypothesis it follows that there exists a syllogistic scheme σ_{ZF} relative to X, $\sim$, ${}^{\bullet}$ whose models correctly model p. This, in view of Proposition 1, gives the result. We construct such a syllogistic scheme as follows.

For each pair s_0, s_1 of distinct variables in S for which $\{x\colon s_0 \in \dot{x}\} = \{x\colon s_1 \in \dot{x}\}$, we consider an acceptable place $P_{s_0 s_1}$ of p, $\sim$ such that

$$s_0 \in P_{s_0 s_1} \text{iff } s_1 \notin P_{s_0 s_1}.$$

Moreover, for each s in S for which $\neg$*Finite* v occurs in p with $s \sim v$, we consider an acceptable place P_s such that $s \in P_s$ and P_s does not contain any variable w with *Finite* w occurring in p. Note that the existence of such places $P_{s_0 s_1}$ and P_s is insured by the p-compatibility of $\sim$ (cf. Definition 1). Let Y be a set of variables, disjoint from X, and let ${}^{\circ}$ be a bijection from Y onto $Pow(X)\backslash\{\emptyset\}$. We denote by Z the set of all y in Y with $\mathring{y} = P \cap S$, where P is a place of the type $P_{s_0 s_1}$ or of the type P_s. Moreover we denote by F the set of all z in Z with $\mathring{z} = P \cap S$ where P is *not* of the form P_s. It is an easy matter to verify that σ_{ZF} is a syllogistic scheme over X, $\sim$, ${}^{\bullet}$.

In order to complete the proof of the lemma it only remains to show that every model of σ_{ZF} is a model of p. Let, therefore, M be a model of σ_{ZF}. Proposition 1(b) implies at once that M correctly models all literals in p of type $x = y$, $x \neq y$, $x \in y$, $x \notin y$, *Finite* x, and $\neg$*Finite* x. If $v = w_1 \cup w_2$ occurs in p, then supposing $v \sim s_v$, $w_1 \sim s_{w_1}$, and $w_2 \sim s_{w_2}$, we have (cf. Definition 4)

$$\begin{aligned} v^M = s_v^M &= \bigcup_{\substack{z \in Z \\ s_v \in \mathring{z}}} z^M \cup \bigcup_{\substack{r \in S \\ s_v \in r}} \{r^M\} \\ &= \left(\bigcup_{\substack{z \in Z \\ s_{w_1} \in \mathring{z}}} z^M \cup \bigcup_{\substack{r \in S \\ s_{w_1} \in \dot{r}}} \{r^M\} \right) \cup \left(\bigcup_{\substack{z \in Z \\ s_{w_2} \in \mathring{z}}} z^M \cup \bigcup_{\substack{r \in S \\ s_{w_2} \in r}} \{r^M\} \right) \\ &= s_{w_1}^M \cup s_{w_2}^M = w_1^M \cup w_2^M, \end{aligned}$$

since, by the p-$\sim$-compatibility of $(S, {}^{\bullet})$, sets $\dot{r}$ and $\mathring{z}$ are of the form $P \cap S$ for some acceptable place P of p, $\sim$, so that $s_v \in P$ if and only if $s_{w_1} \in P$ or $s_{w_2} \in P$. This shows that M is a model of the literals in p of type $v = w_1 \cup w_2$. Analogously it can be shown that the remaining clauses of type $v = w_1 \backslash w_2$ are also modeled correctly, thus showing that M is model of p. This completes the proof of the lemma. ■

Combining together Corollary 2 and Lemma 6 we re-discover that the class MLSF of formulae has a decidable satisfiability problem:

COROLLARY 3. *An algorithm for testing a normalized conjunction p of MLSF for satisfiability is*:

(1) *construct the equivalence relation* $\sim_p$;

(2) *check whether* $\sim_p$ *is p-compatible*;

(3) *check whether* $\sim_p$ *admits a* p-$\sim_p$*-compatible DAG.*

3. On Complexity Issues

In this section we address some complexity issues concerning the decision problem in set theory; in particular we prove that the class of $(\forall)_0$-simple prenex formulae introduced in [2] with the name 'purely universal simple prenex formulae' and studied in [9] and [8] has a decision problem which is NP-complete, when the length of the quantifier prefixes cannot exceed a *fixed* number l.

By the *decision problem* for a given class $\mathscr{C}$ of set theoretic formulae we mean the problem of establishing whether or not given any formula ψ in the $\mathscr{C}$ there are sets (in a *standard* or *naïve* model of set theory) that satisfy ψ, when substituted for its free variables.

The class of $(\forall)_0$-simple prenex formulae introduced here has a form which is slightly different from the definition given in [2] but the two definitions can be easily seen to be equivalent by using the normal form theorem presented in [9].

DEFINITION 5. A formula φ is in the class of $(\forall)_0$-simple prenex formulae if

(i) φ is of the form

$$\&_{i=1,\ldots,n}\varphi_i$$

where

$$\varphi_i = (\forall x_{h_1} \in y_{h_1}) \cdots (\forall x_{h_{m_i}} \in y_{h_{m_i}})(l_1^i \vee \cdots \vee l_{k_i}^i)$$

with l_j^i a literal, $1 \leqslant j \leqslant k_i$, and $m_i \geqslant 0$ (if $m_i = 0$, φ_i is unquantified), and

(ii) the *maximum nesting level* of each variable in every φ_i is one, i.e., in any φ_i no x_{h_s} is a y_{h_t}, for any s and t (see [2, 9]).

Moreover, a $(\forall)_0$-simple prenex formula φ is said to be *l-bounded* if the length of the quantifier prefixes in all of its conjuncts φ_i does not exceed l. We will denote by $(\forall)_0^l$-s.p. the class of l-bounded $(\forall)_0$-simple prenex formulae. □

EXAMPLES. The following are all $(\forall)_0^1$-simple prenex formulae:

- $(\forall x \in y)(x \in z)$,
  expressing that y is included in z: $y \subseteq z$.
- $(\forall x \in y)(x \in z)$ & $(\forall x \in y)(x \notin w)$ & $(\forall x \in z)(x \in w \vee x \in y)$,
  expressing that y is the set difference $z \backslash w$: $y = z \backslash w$
- $(\forall x \in y)(x \in z \vee x \in w)$ & $(\forall x \in z)(x \in y)$ & $(\forall x \in w)(x \in y)$,
  expressing that y is the set union $z \cup w$: $y = z \cup w$
- $(\forall x \in y)(x = z)$ & $z \in y$,
  expressing that y is the singleton containing z: $y = \{z\}$.

The following is **not** a $(\forall)_0$-simple prenex formula:

- $(\forall x \in y)(\forall z \in x)(z \in y)$,
  expressing that y is a *transitive* set.

The previous examples also show that the theory MLSS, i.e. the class of unquantified formulae in the language involving $\cap, \setminus, \cup, \{\bullet, \dots, \bullet\}, \in \subseteq, =$, and the propositional connectives is a subclass of the class of $(\forall)_0^1$-simple prenex formulae.

3.1. THE $(\forall)_0^l$-SIMPLE PRENEX DECISION PROBLEM IS NP-HARD

In this section we prove that the decision problem for the class of unquantified formulae including only the membership predicate $\in$ and no other extra-logical symbol is NP hard. We will denote such a language by T_0. The NP-hardness of the $(\forall)_0$-s.p. decision problem will then follow immediately.

We prove that the decision problem for T_0 is NP-hard by showing that the well-known satisfiability problem SAT for propositional calculus can be polynomially reduced to it (see [1]).

Let $\mathscr{P}$ be a formula of propositional calculus in conjunctive normal form, that is:

$$\mathscr{P} = \mathscr{P}_1 \,\&\, \cdots \,\&\, \mathscr{P}_n$$

where $\mathscr{P}_1, \dots, \mathscr{P}_n$ are all *disjunctions* of literals. Let $X_1, \dots, X_m$ be the propositional variables occurring in $\mathscr{P}$ and let $z, x_1, \dots, x_m$ be distinct set variables.

In order to associate to $\mathscr{P}$ a set formula φ of T_0, we substitute in $\mathscr{P}$ each propositional variable X_i by the atomic set formula $z \in x_i$. Thus, for instance, if

$$\mathscr{P} = (X_1 \vee \neg X_2) \,\&\, (\neg X_1 \vee X_3),$$

then

$$\varphi = (z \in x_1 \vee \neg z \in x_2) \,\&\, (\neg z \in x_1 \vee z \in x_3)$$

or, equivalently,

$$\varphi = (z \in x_1 \vee z \notin x_2) \,\&\, (z \notin x_1 \vee z \in x_3).$$

We claim that the propositional formula $\mathscr{P}$ is satisfiable if and only if the T_0-formula φ is satisfiable in the standard model of set theory. Indeed, if $\mathscr{A}$ is any truth assignment which satisfies $\mathscr{P}$, then the set assignment

$$Mz = \emptyset$$
$$Mx_i = \begin{cases} \{\emptyset\} & \text{if } \mathscr{A}(X_i) = \mathit{true} \\ \emptyset & \text{if } \mathscr{A}(X_i) = \mathit{false} \end{cases} \tag{1}$$

is easily seen to satisfy φ.

Conversely, given any set model M of φ, then the truth assignment

$$\mathscr{A}(X_i) = \begin{cases} \textit{true} & \text{if } Mz \in Mx_i \\ \textit{false} & \text{if } Mz \notin Mx_i \end{cases}$$

obviously satisfies $\mathscr{P}$.

By extending the language T_0 with the boolean set operators $\cup$ and $\backslash$, we can further restrict to just *conjunctions* of atoms of type $z \in t$, where t is any set term involving $\cup$ and $\backslash$ only. In fact, given $\mathscr{P}$ and $z, x_1, \ldots, x_m$ as above, we introduce an additional new set variable u (standing for a 'universe'). Then, to each conjunct

$$X_{i_1} \vee \cdots \vee X_{i_h} \vee \neg X_{i_{h+1}} \vee \cdots \vee \neg X_{i_k}$$

of $\mathscr{P}$ we associate the following atomic set formula

$$z \in x_{i_1} \cup \cdots \cup x_{i_h} \cup (u \backslash x_{i_{h+1}}) \cup \cdots \cup (u \backslash x_{i_k}). \tag{2}$$

By letting φ to be the conjunction of all the atoms (2) associated with each conjunct of $\mathscr{P}$, much as previously it can be shown that $\mathscr{P}$ is propositionally satisfiable if and only if φ is satisfiable by a set model. But in this case (1) needs to be defined also over the new variable u. So we put $Mu = \{\emptyset\}$.

Finally, by using a simple normalization process of the kind described in [3], one can easily prove that the satisfiability problem for the class of *conjunctions* of atoms of the simple types

$$x \in y, \qquad x = y \cup z, \qquad x = y \backslash z,$$

where x, y and z stand for variables, is also NP-hard.

3.2. THE $(\forall)_0^l$-SIMPLE PRENEX DECISION PROBLEM IS IN NP

Let l be a *fixed* nonnegative integer. In this section we prove that the $(\forall)_0^l$-s.p. satisfiability problem is in NP by exhibiting a nondeterministic algorithm which tests satisfiability in polynomial time in the size of the input.

Let

$$\varphi = \varphi_1 \mathbin{\&} \cdots \mathbin{\&} \varphi_n$$

be a $(\forall)_0^l$-s.p. formula having free variables $x_1, \ldots, x_m$. Assume that φ is satisfiable and let M be a set model of φ. Then a graph $G_{\varphi,M} = (V, E)$ (which will be referred to as the *model-graph* of φ with respect to the model M) can be associated with the collection of sets $Mx_1, \ldots, Mx_m$ in the following way.

First, the set of nodes V of $G_{\varphi,M}$ is defined by

$$V = \{v_i : 1 \leqslant i \leqslant m\} \cup \{u_{i,j} : 1 \leqslant i, j \leqslant m, i \neq j\}.$$

We associate the set Mx_i with the node v_i, $1 \leqslant i \leqslant m$, whereas to each node of type $u_{i,j}$ we associate an element of $Mx_i \backslash Mx_j$, provided that $Mx_i \backslash Mx_j$ is nonempty (otherwise $u_{i,j}$ has no corresponding set). Notice that the number of nodes in $G_{\varphi,M}$ is $\mathcal{O}(n^2)$.

Next we define the set E of edges of $G_{\varphi,M}$. Given any two nodes a, b in V, the edge (a, b) is in E if and only if

(a) the nodes a and b have sets associated to them, and

(b) $A \in B$ holds, where A is the set associated with a and B is the set associated with b.

REMARK 1. The graph $G_{\varphi,M}$ is acyclic, since $\in$ is acyclic. ■

Intuitively, the graph $G_{\varphi,M}$ stores all relevant information of the model M with respect to the class $(\forall)_0^l$-s.p. This is best seen in the following construction.

Let $w_1, \ldots, w_s$ be the nodes associated with sets in $\{Mx_1, \ldots, Mx_m\}$ and let $w'_1, \ldots, w'_t$ be the remaining nodes. Let $I_1, \ldots, I_t$ be t sets having the same finite rank p, with $p > |V|$, and such that $I_i \neq I_j$ if and only if the set associated with w'_i is different from the set associated with w'_j.

Next, we inductively define a map $\star$ on the set V by setting

$$w_i^\star = \{z^\star : z \in V \;\&\; (z, w_i) \in E\}$$
$$(w'_j)^\star = \{I_j\} \cup \{z^\star : z \in V \;\&\; (z, w'_j) \in E\}.$$

The map $\star$ has the following properties:

LEMMA 7. *For all $a, b \in V$,*

$(=)$ $\quad a^\star = b^\star \leftrightarrow A = B$, and

$(\in)$ $\quad a^\star \in b^\star \leftrightarrow A \in B$,

where A and B are the sets corresponding to the nodes a and b, respectively.

Proof of $(=)$. Let $a, b \in V$ and let their corresponding sets be A and B, respectively. If $A = B$, then by definition $a^\star = b^\star$. To show the converse, assume by contradiction that a is an E-minimal node for which there exists a node b such that $A \neq B$ and $a^\star = b^\star$.

We need to consider the following four cases.

(1) a and b are both nodes of type w, i.e. they are associated with Mx_i, Mx_j for some i and j;

(2) a is of type w and b is of type w';

(3) a is of type w' and b is of type w;

(4) a and b are both of type w'.

Case 1. Since $A = Mx_i \neq Mx_j = B$, then either $Mx_i \backslash Mx_j$ or $Mx_j \backslash Mx_i$ is nonempty, and therefore either $u_{i,j}$ or $u_{j,i}$ has a corresponding set associated with it. Let us assume that the node $u_{i,j}$ is associated with the set $U_{i,j}$. Thus $U_{i,j} \in Mx_i = A$, from which $(u_{i,j})^\star \in a^\star$, and $U_{i,j} \notin Mx_j = B$. But $a^\star = b^\star$, so that $(u_{i,j})^\star \in b^\star$. Observe that all the elements in $b^\star$ are of type $z^\star$ for some $z \in V$, since b is of type w (see the definition of $\star$ in the case of nodes of type w). Hence there must exist a node $z \in V$ different from $u_{i,j}$ and such that $z^\star = (u_{i,j})^\star$, $(z, b) \in E$, and $Z \neq U_{i,j}$, where Z is the set associated with the node z. But this contradicts the E-minimality of a, as the node $u_{i,j}$ E-precedes a.

The case in which the node $u_{j,i}$ has a set associated with it is completely analogous.

Cases 2 and 3 can be proved by observing that $(w')^\star$ has elements of rank exactly p, whereas $\star$ interprets every node of type w with sets that can have no element of rank p.

Case 4 follows easily by observing that distinct nodes of type w' have distinct corresponding sets I's.

Proof of $(\in)$. Again, let $a, b \in V$. If $A \in B$, then $(a, b) \in E$, so that $a^\star \in b^\star$ follows from the definition of $\star$. Conversely, if $a^\star \in b^\star$ whereas by contradiction $A \notin B$, then there must exist a node $z \in V$ such that $(z, b) \in E$ and $z^\star = a^\star$. But then $Z \in B$, where Z is the set corresponding to the node z, contradicting $(=)$, since plainly $Z \neq A$. ■

Finally, we define the set assignment $M^\star$ over φ by putting

$$M^\star x_i = v_i^\star.$$

From Lemma 7, the assignment $M^\star$ is a model of φ. Indeed, assume by way of contradiction that the assignment $M^\star$ does not satisfy φ. Since φ has the form $\varphi = \varphi_1 \,\&\cdots\&\, \varphi_n$, it follows that there is a $\bar{\imath} \in \{1, \ldots, n\}$ such that $M^\star$ satisfies $\neg\varphi_{\bar{\imath}}$. Therefore, the sets $M^\star x_i$, $i = 1, \ldots, m$, together with some of their elements of type $(w')^\star$ would satisfy the negation of the matrix of $\varphi_{\bar{\imath}}$, no individual I_j being involved. But then, from $(=)$ and $(\in)$ of Lemma 7, the corresponding sets in the model M would also satisfy the negation of the matrix of $\varphi_{\bar{\imath}}$. In particular, M would satisfy $\neg\varphi_{\bar{\imath}}$, which is a contradiction.

Having proved that $M^\star$ is a model of φ, it follows that the following nondeterministic algorithm can decide in polynomial time whether a $(\forall)_0^l$-s.p. formula is satisfiable or not.

Step 1. Guess the graph $G_{\varphi,M} = (V, E)$ (this amounts to guessing the set E of edges only, since V is determined).

Step 2. Guess which nodes in V besides v_i, $i = 1, \ldots, m$, are of type w (i.e., guess the nodes associated with sets in $\{Mx_1, \ldots, Mx_m\}$).

Step 3. Define $\star$ (notice that one does not need to specify the internal structure of the sets I's).

Step 4. Check that any conjunct in φ is satisfied by the assignment $M^\star x_i = v_i^\star$.

Clearly, steps 1–3 can be executed in nondeterministic polynomial time. To show that step 4 takes also polynomial time in the length of the input formula φ, it is enough to observe the following fact:

- for any $a, b \in V$, the verification of $a^\star = b^\star$ or $a^\star \in b^\star$ can be done in polynomial time;
- to check whether $M^\star$ satisfies each conjunct φ_i of φ one has to verify only polynomially many atomic formulae of types $a^\star = b^\star$ or $a^\star \in b^\star$, since the number of quantifiers in φ_i is at most l.

We have just shown that the model graph technique applied to the class of $(\forall)_0^l$-s.p. formulae yields a nondeterministic polynomial decision test. This is in fact a very favorable situation.

The results in [9] are examples of cases in which the resulting decision procedures derived have a super-exponential time complexity. A possible explanation of this fact is that the languages considered in [9] are expressive enough to force certain variables to be modeled only by sets of bounded and finite rank. This prevents us from dealing with individuals (i.e., sets of type I) just symbolically, as in steps 3 and 4 of the above nondeterministic algorithm. In those cases, it appears that one has to guess exactly the sets corresponding to the so-called *trapped* variables, causing an inevitable combinatorial explosion.

It would be interesting to find different approaches to the definition of $\star$ that could solve this problem without the complexity of the resulting algorithm blowing up.

References

1. Aho, A. V., Hopcroft, J. E., and Ullman, J. D., *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass. (1983).
2. Breban, M., Ferro, A., Omodeo, E. G., and Schwartz, J. T., 'Decision procedures for elementary sublanguages of set theory. II. Formulas involving restricted quantifiers, together with ordinal, integer, map, and domain notions', *Comm. Pure App. Math.* **34**, 177–195 (1981).
3. Cantone, D., 'A decision procedure for a class of unquantified formulae of set theory involving the powerset and singleton operators', PhD Thesis, New York Univ. – GSAS, Courant Institute of Mathematical Sciences, New York (1987).
4. Cantone, D., Ghelfo, S., and Omodeo, E. G., 'The automation of syllogistic. I. Syllogistic normal forms', *J. Symbol. Comp.* **6**(1); 83–98 (1988).
5. Ghelfo, S. and Omodeo, E. G., 'Towards practical implementations of syllogistic', in *EUROCAL '85, European Conf. on Computer Algebra – Proceedings*, Vol. 2, pp. 40–48, Springer-Verlag (1985).
6. Mendelson, E., *Introduction to Mathematical Logic*, Van Nostrand-Reinhold, Princeton, New Jersey (1964).
7. Omodeo, E. G., 'Decidability and proof procedures for set theory with a choice operator', PhD Thesis, New York Univ. – GSAS, Courant Institute of Mathematical Sciences, New York (1984).
8. Parlamento, F. and Policriti, A., 'The logically simplest form of the infinity axiom; *Proc. AMS* **103**(1), (May 1988).
9. Parlamento, F. and Policriti, A., 'Decision procedures for elementary sublanguages of set theory. XIII. Model graphs, reflection and decidability', *J. Symb. Comp.*, – *Special issue* (to appear). *New Trends in Automated Mathematical Reasoning – Proceedings.*