

A Decision Procedure for Monotone Functions over Lattices^{*}

Domenico Cantone¹ and Calogero G. Zarba²

¹ Dipartimento di Matematica e Informatica
Università degli Studi di Catania, Italy
`cantone@dmf.unict.it`

² Computer Science Department
Stanford University, USA
`zarba@theory.stanford.edu`

Abstract. This paper presents a practical decision procedure for the unquantified theory of lattices with monotone functions. Specifically, it considers the unquantified language **Lmf** with the predicates $=$ and \leq and with the operators *inf* and *sup* over terms which may involve also uninterpreted function symbols. Additional predicates expressing increasing and decreasing monotonicity of functions are allowed as well as a predicate \preceq for pointwise functions comparison.

For a restricted collection of conjunctions, denoted **Lmf**^{*}, we give a quadratic satisfiability test. We also describe a nondeterministic quadratic reduction of the satisfiability problem for **Lmf**-formulae to the one for **Lmf**^{*}, which allows to prove the \mathcal{NP} -completeness of the former problem.

Key words. Satisfiability decision problem, proof verification, constraints in lattice theory.

1 Introduction

Lattices are partial orders in which every pair of elements has a least upper bound and a greatest lower bound. They have several applications in mathematics and computer science, including model checking [5], knowledge representation [7], and partial order programming [6].

In this paper we introduce the unquantified language **Lmf** (*Lattices with Monotone Functions*) for expressing constraints over lattices and monotone functions. The language contains the equality predicate $=$, an ordering predicate \leq , and the operators *inf* and *sup*. The language also allows for uninterpreted unary functions, and has predicates for expressing increasing and decreasing

^{*} This research has been partially supported by MURST Grant prot. 2001017741 under project “*Ragionamento su aggregati e numeri a supporto della programmazione e relative verifiche.*”

monotonicity of functions, as well as a predicate \preceq for pointwise functions comparison.

We then present a practical decision procedure for **Lmf**, and we prove that the decision problem for **Lmf** is \mathcal{NP} -complete.

The satisfiability problem for unquantified formulae in the more restricted language obtained by dropping the operators *inf* and *sup* and the predicate \preceq from **Lmf** was studied in [1]. The decision procedure given there was based on a nondeterministic quadratic reduction to the \mathcal{NP} -complete satisfiability problem for the unquantified fragment of set theory denoted **MLS** [2, 4], thus yielding a much less practical decision test than the one presented in this paper. We recall that the decision problem for the fully quantified theory of lattices is undecidable, as proved by Tarski in [8].

The paper is structured as follows. In Section 2 we define the syntax and semantics of the unquantified language **Lmf**, and we provide a nondeterministic quadratic reduction of the decision problem for **Lmf** to the decision problem for a restricted class of conjunctions denoted **Lmf**^{*}. In Section 3 we give a quadratic satisfiability test for **Lmf**^{*}, and we prove its correctness. In Section 4 we prove that the decision problem for **Lmf** is \mathcal{NP} -complete. Finally, in Section 5 we conclude the paper with directions for future research.

2 The theory of lattices with monotone functions

We present the syntax and semantics of an unquantified theory of lattices with monotone functions and discuss some elementary reductions of its satisfiability problem.

2.1 Syntax

The language **Lmf** (*Lattices with Monotone Functions*) is the unquantified language containing an enumerable collection of variables x, y, z, \dots , the constants m and M (intended to denote the minimum and the maximum of the lattice), the equality predicate symbol $=$, the binary predicate symbol \leq , the binary function symbols *inf* and *sup* (intended to denote the greatest lower bound and the least upper bound, respectively), an enumerable collection of unary function symbols f, g, \dots , the unary predicates *inc* and *dec* (intended to state increasing and decreasing monotonicity of functions, respectively), and the binary predicate \preceq (intended to compare functions pointwise).

The terms of **Lmf** are defined in the standard way, namely:

- any variable or one of the constants m and M is a term;
- if t is a term and f is a function symbol, then $f(t)$ is a term;
- if t_1 and t_2 are terms, then $\text{inf}(t_1, t_2)$ and $\text{sup}(t_1, t_2)$ are terms.

The atomic formulae of **Lmf** are the following:

$$t_1 = t_2, \quad t_1 \leq t_2, \quad \text{inc}(f), \quad \text{dec}(f), \quad f \preceq g, \quad (1)$$

where t_1, t_2 stand for terms of **Lmf** and f, g stand for function symbols.

Finally, the formulae of **Lmf** are the propositional combinations of atomic formulae of **Lmf**, by means of the propositional connectives $\neg, \wedge, \vee, \rightarrow$.

2.2 Semantics

The intended semantics for **Lmf** is based upon *complete lattices*. Among the equivalent definitions of lattices, we adopt the following one.

Definition 1. A LATTICE is a pair $\langle \mathbf{L}, \leq \rangle$ such that

- \mathbf{L} is non-empty set, called the SUPPORT of the lattice;
- \leq is a partial order over \mathbf{L} ;
- every pair of elements of \mathbf{L} has both a greatest lower bound and a least upper bound.

A lattice $\langle \mathbf{L}, \leq \rangle$ is COMPLETE if every subset of \mathbf{L} has both a greatest lower bound and a least upper bound.

Remark 1. A complete lattice has both minimum and maximum.

Definition 2. An **Lmf**-INTERPRETATION \mathcal{A} is a pair $\langle \mathcal{L}_{\mathcal{A}}, (\cdot)^{\mathcal{A}} \rangle$ such that

- $\mathcal{L}_{\mathcal{A}} = \langle \mathbf{A}, \leq \rangle$ is a complete lattice with support \mathbf{A} ;
- $(\cdot)^{\mathcal{A}}$ is a mapping which interprets
 - each variable x of **Lmf** as an element $x^{\mathcal{A}}$ in \mathbf{A} ;
 - the constants m and M as the minimum and maximum elements of $\mathcal{L}_{\mathcal{A}}$, respectively;
 - the binary predicate symbol \leq as the partial ordering \leq of $\mathcal{L}_{\mathcal{A}}$;
 - the binary function symbols \inf and \sup as the greatest lower bound (glb) and the least upper bound (lub) with respect to the partial order \leq ;
 - each unary function symbol f as a map $f^{\mathcal{A}} : \mathbf{A} \rightarrow \mathbf{A}$;
 - each atomic formula $\text{inc}(f)$ as the truth value **true**, provided that the map $f^{\mathcal{A}}$ is increasing in \mathbf{A} , with respect to \leq ;
 - each atomic formula $\text{dec}(f)$ as the truth value **true**, provided that the map $f^{\mathcal{A}}$ is decreasing in \mathbf{A} , with respect to \leq ;
 - each atomic formula $f \preceq g$ as the truth value **true**, provided that $f^{\mathcal{A}}(a) \leq g^{\mathcal{A}}(a)$ holds, for every $a \in \mathbf{A}$.

Given a formula Φ of **Lmf**, we denote by $\Phi^{\mathcal{A}}$ the truth value of Φ under the **Lmf**-interpretation \mathcal{A} .

If $\Phi^{\mathcal{A}} = \mathbf{true}$, then \mathcal{A} is called a MODEL of Φ .

Remark 2. As will appear clear later, for our decidability purposes we could have used a less restricted semantics, based on lattices with minimum and maximum, rather than on complete lattices. We preferred to use complete lattices in view of a future extension of **Lmf** with set variables X, Y, Z, \dots and, among others, atomic formulae of the form $x = \inf(X)$ and $x = \sup(X)$.

2.3 The satisfiability problem for **Lmf**

The satisfiability problem for **Lmf** is the problem of establishing for any given formula Φ of **Lmf** whether it has a model, that is, whether there exists an **Lmf**-interpretation \mathcal{A} such that $\Phi^{\mathcal{A}}$ is true. Such a problem can be reduced to simpler ones, as the following considerations show.

We begin by observing that using disjunctive normal form, the general satisfiability problem for **Lmf** can be reduced to the satisfiability problem for conjunctions of literals of **Lmf**, namely atomic **Lmf**-formulae of the form (1) or their negations.

As a second step, **Lmf**-literals can be put in *flat form*. This can be achieved by suitably renaming terms by newly introduced variables: for instance, a literal of the form $f_1(f_2(x)) \neq g_1(g_2(g_3(y)))$ can be reduced by such technique to the equisatisfiable conjunction

$$z_1 = f_2(x) \wedge z_2 = f_1(z_1) \wedge w_1 = g_3(y) \wedge w_2 = g_2(w_1) \wedge w_3 = g_1(w_2) \wedge z_2 \neq w_3,$$

where z_1, z_2, w_1, w_2, w_3 are new variables. Notice that the complexity of the above flattening process is linear.

Finally, negative literals of type

$$\begin{array}{lll} x_1 \neq f(x_2), & x_1 \neq \inf(x_2, x_3), & x_1 \neq \sup(x_2, x_3), \\ \neg \text{inc}(f), & \neg \text{dec}(f), & f \not\leq g \end{array} \quad (2)$$

can be replaced by equisatisfiable conjunctions whose negative literals are only of the types $x \neq y$ or $\neg(x \leq y)$. For instance, the literal $x_1 \neq \inf(x_2, x_3)$ is equisatisfiable with the conjunction $w = \inf(x_2, x_3) \wedge w \neq x_1$, where w is a new variable, and the literal $\neg \text{inc}(f)$ is equisatisfiable with the conjunction $w_1 = f(z_1) \wedge w_2 = f(z_2) \wedge z_1 \leq z_2 \wedge \neg(w_1 \leq w_2)$, where w_1, w_2, z_1, z_2 are new variables. Analogous considerations hold for the remaining four types of negative literals. Again, the above process can be performed in linear time.

In conclusion, it turns easily out that the satisfiability problem for **Lmf** can be reduced to the satisfiability problem for conjunctions of **NORMALIZED Lmf-LITERALS**, namely literals of the following types

$$\begin{array}{llll} x = y, & x \neq y, & x \leq y, & \neg(x \leq y), \\ x = \inf(y, z), & x = \sup(y, z), & x = f(y), & \\ \text{inc}(f), & \text{dec}(f), & f \leq g, & \end{array} \quad (3)$$

where x, y, z range over the variables and constant symbols of **Lmf** and f, g range over the function symbols of **Lmf**.

Notice that as a by-product of the preceding discussion we have the following result.

Lemma 1. *For any conjunction of **Lmf**-literals, one can construct in linear time an equisatisfiable conjunction of normalized **Lmf**-literals.*

Finally, we further restrict our attention to conjunctions Ψ of normalized **Lmf**-literals which satisfy the following closure conditions (C1), (C2), and (C3); we call such formulae **CLOSED CONJUNCTIONS OF NORMALIZED Lmf-LITERALS** and denote their collection by **Lmf***.

*Closure conditions for **Lmf**^{*}-conjunctions*

- (C1) Ψ contains occurrences of both constants m and M .
- (C2) Ψ contains at least one literal of the form $x = f(m)$ and one literal of the form $x' = f(M)$, for each function symbol f in Ψ .
- (C3) For each literal of type $f \preceq g$ in Ψ and for each variable y , the conjunction Ψ contains a literal of the form $x = f(y)$ whenever it contains a literal of the form $x' = g(y)$, and conversely.

It is easy to see that given any conjunction Φ of normalized **Lmf**-literals, by a simple quadratic completion process one can construct an equisatisfiable closed conjunction Ψ of normalized **Lmf**-literals. Thus, we have:

Lemma 2. *For any conjunction of normalized **Lmf**-literals, one can construct in quadratic time an equisatisfiable closed conjunction of normalized **Lmf**-literals.*

Summing up, we have proved:

Theorem 1. *The satisfiability problem for **Lmf** is equivalent to the satisfiability problem for **Lmf**^{*}.*

In the next section we will provide a quadratic satisfiability test for **Lmf**^{*}.

3 A satisfiability test for **Lmf**^{*}

Our satisfiability test is based on the collection of saturation rules listed in Table 1.

Definition 3. *A collection H of normalized **Lmf**-literals is said to be SATURATED (with respect to the rules of Table 1) if*

- *for each rule \mathcal{R} of Table 1 with premisses, the conclusions of \mathcal{R} belong to H whenever its premisses belong to H ;*
- *the literals*

$$x = x, \quad x \leq x, \quad m \leq x, \quad x \leq M$$

belong to H , for each variable or constant x occurring in H .¹

Given an **Lmf**^{*}-conjunction Ψ , we define the CLOSURE of Ψ as the minimal saturated collection of normalized **Lmf**-literals containing the literals of Ψ . It is not hard to see that closures can be calculated by the quadratic procedure in Table 2.

Notice that procedure $\text{Closure}(\cdot)$ does not introduce any new variable during its computation. Therefore, if its input **Lmf**^{*}-conjunction Ψ contains p distinct variables and constant symbols and q distinct function symbols, then procedure $\text{Closure}(\cdot)$ adds $\mathcal{O}(p^2 + q^2)$ new literals to the closure of Ψ .

Closures play a particularly important rôle in our satisfiability test for **Lmf**^{*}-conjunctions; this just consists, as shown in Table 3, in checking whether the closure of the **Lmf**^{*}-conjunction to be tested for satisfiability contains a pair of complementary literals.

¹ Such condition amounts to the saturatedness of H with respect to the rules with no premisses [=1], [\leq .1], [\leq .4], and [\leq .5].

<i>=-rules</i>		
$\frac{}{x = x} [= .1]$	$\frac{x = y}{y = x} [= .2]$	$\frac{x = y}{\ell_y^x} [= .3]$
<i>≤-rules</i>		
$\frac{}{x \leq x} [\leq .1]$	$\frac{x \leq y}{x = y} [\leq .2]$	$\frac{x \leq y}{x \leq z} [\leq .3]$
$\frac{}{m \leq x} [\leq .4]$	$\frac{}{x \leq M} [\leq .5]$	
<i>inf-rules</i>		
$\frac{x = \inf(y, z)}{x \leq y}$ [I.1]	$\frac{x = \inf(y, z)}{x \leq z}$	$\frac{x = \inf(y, z)}{w \leq y}$
		$\frac{w \leq z}{w \leq x}$ [I.2]
<i>sup-rules</i>		
$\frac{x = \sup(y, z)}{y \leq x}$ [S.1]	$\frac{x = \sup(y, z)}{z \leq x}$	$\frac{x = \sup(y, z)}{y \leq w}$
		$\frac{z \leq w}{x \leq w}$ [S.2]
<i>Functions rules</i>		
$\frac{x = f(y)}{x' = f(y')}$ [f.1]	$\frac{inc(f)}{x \leq y}$	$\frac{dec(f)}{x \leq y}$
	$\frac{z = f(x)}{w = f(y)}$ [f.2]	$\frac{z = f(x)}{w = f(y)}$ [f.3]
	$\frac{w = f(y)}{z \leq w}$	$\frac{w = f(y)}{w \leq z}$
<i>⊑-rules</i>		
$\frac{f \preceq g}{g \preceq h}$ [⊑.1]	$\frac{f \preceq g}{x = f(y)}$	$\frac{f \preceq g}{z = g(y)}$
	$\frac{g \preceq h}{f \preceq h}$	$\frac{z = g(y)}{x \leq z}$ [⊑.2]

In the above rules, the symbols x, x', y, y', z, w stand for variables or constant symbols, whereas f, g stand for function symbols. In addition, ℓ stands for a normalized **Lmf**-literal and ℓ_y^x denotes the result of substituting in ℓ all occurrences of the symbol x by the symbol y .

Table 1. Saturation rules for **Lmf**^{*}-conjunctions.

Closure(Ψ)

*Comment: Ψ is an **Lmf**^{*}-conjunction.*

$H :=$ collection of the literals in Ψ ;

for each variable or constant symbol x in Ψ **do**

$H := H \cup \{x = x, x \leq x, m \leq x, x \leq M\}$;

while H is not saturated **do**

- let \mathcal{R} be an **Lmf**^{*}-saturation rule such that $\emptyset \neq \mathcal{P}_{\mathcal{R}} \subseteq H$
but $\mathcal{C}_{\mathcal{R}} \not\subseteq H$, where $\mathcal{P}_{\mathcal{R}}$ and $\mathcal{C}_{\mathcal{R}}$ are respectively the
premisses and the conclusions of \mathcal{R} ;

$H := H \cup \mathcal{C}_{\mathcal{R}}$;

return(H);

Table 2. Saturation function for **Lmf**^{*}-conjunctions.

Lmf^{*}-Satisfiability-Test(Ψ)

*Comment: Ψ is an **Lmf**^{*}-conjunction.*

$H :=$ Closure(Ψ);

if H contains a pair of complementary literals $\ell, \neg\ell$ **then**

return “ Ψ is unsatisfiable”

else

return “ Ψ is satisfiable”

Table 3. A satisfiability test for **Lmf**^{*}-conjunctions.

3.1 An example

We illustrate the complete decision process on a simple example. Let

$$\Phi =_{Def} (f \preceq g \wedge inc(f) \wedge dec(g) \wedge f(x) \neq g(x)) \rightarrow f(m) \neq g(m).$$

To prove that Φ is true under all **Lmf**-interpretation, we can show that its negation $\neg\Phi$ is unsatisfiable. Plainly, $\neg\Phi$ is equivalent to the conjunction

$$\Phi_1 =_{Def} f \preceq g \wedge inc(f) \wedge dec(g) \wedge f(x) \neq g(x) \wedge f(m) = g(m).$$

By normalizing Φ_1 , we obtain the equisatisfiable conjunction

$$\Phi_2 =_{Def} f \preceq g \wedge inc(f) \wedge dec(g) \wedge y_1 = f(x) \wedge y_2 = g(x) \wedge y_1 \neq y_2 \wedge \\ z_1 = f(m) \wedge z_2 = g(m) \wedge z_1 = z_2.$$

By adding to Φ_2 suitable literals, we further obtain an **Lmf**^{*}-conjunction Φ_3 which is equisatisfiable with Φ_2 .

Next, let $H = \text{Closure}(\Phi_3)$. We prove that H is unsatisfiable by showing that it contains a pair of complementary literals. Notice that H must contain the following literals, among others:

- | | |
|--|--|
| (a) $m \leq x$ (by rule $[\leq.4]$) | (f) $z_1 \leq y_2$ (by rule $[\leq.3]$) |
| (b) $z_1 \leq y_1$ (by rule $[f.2]$) | (g) $y_2 = z_1$ (by rule $[\leq.2]$) |
| (c) $y_1 \leq y_2$ (by rule $[\leq.2]$) | (h) $y_1 \leq z_1$ (by rule $[= .3]$) |
| (d) $y_2 \leq z_2$ (by rule $[f.3]$) | (i) $y_1 = z_1$ (by rule $[\leq.2]$) |
| (e) $y_2 \leq z_1$ (by rule $[= .3]$) | (j) $y_1 = y_2$ (by rule $[= .3]$) |

Thus, H contains the pair of complementary literals $y_1 \neq y_2$ (since it belongs to Φ_2) and $y_1 = y_2$, so that it is unsatisfiable. It follows that Φ_3 , Φ_2 , and Φ_1 are unsatisfiable as well, and therefore our initial **Lmf**-formula Φ must be true under all **Lmf**-interpretations.

3.2 Correctness

To prove the correctness of the procedure **Lmf*-Satisfiability-Test**, it is enough to show that for each **Lmf***-conjunction Ψ we have

soundness: if Ψ is satisfiable, then $\text{Closure}(\Psi)$ is satisfiable;

completeness: if $\text{Closure}(\Psi)$ does not contain any pair of complementary literals, then it is satisfiable.

Soundness Concerning soundness, we have a slightly stronger result:

Lemma 3 (Soundness). *Let Φ be a conjunction of normalized **Lmf**-literals and let \mathcal{A} be an **Lmf**-interpretation. Then \mathcal{A} satisfies Φ if and only if it satisfies $\text{Closure}(\Phi)$.*

Proof. Plainly, if an **Lmf**-interpretation \mathcal{A} satisfies $\text{Closure}(\Phi)$ then it satisfies Φ , since all the literals of Φ are contained in $\text{Closure}(\Phi)$.

On the other hand, it can easily be shown by a simple inspection of the saturation rules of Table 1 that if an **Lmf**-interpretation \mathcal{A} satisfies Φ , then it must inductively satisfy all literals which are added to the closure of Φ .

Completeness Let Ψ be an **Lmf***-conjunction and let H be the closure of Ψ . Let us assume that H does not contain any pair of complementary literals ℓ , $\neg\ell$. Let V be the collection of variables and constant symbols occurring in H .

We define an **Lmf**-interpretation \mathcal{A} as follows. We let $A = V/\sim$, where \sim is the equivalence relation induced by the literals of the form $x = y$ in H , and for each x in V , we denote by $[x]$ the equivalence class of x relative to \sim and put $x^{\mathcal{A}} = [x]$.

Clearly, all literals in H of the form $x = y$ and $x \neq y$ are satisfied by \mathcal{A} .

Next, we interpret \leq as follows: for $[x], [y] \in A$, we put

$$[x] \leq^{\mathcal{A}} [y] \text{ if and only if the literal } x \leq y \text{ occurs in } H.$$

Such a definition is well-given, since if $x \leq y$ is in H , $x' \in [x]$, and $y' \in [y]$, then by the $=$ -rules it follows that $x' \leq y'$ is in H too.

By saturation with respect to the \leq -rules, $\leq^{\mathcal{A}}$ is a partial order with minimum $[m]$ and maximum $[M]$. Since A is finite, the lattice induced by $\langle A, \leq^{\mathcal{A}} \rangle$ is complete.

Plainly, all literals in H of the form $x \leq y$ and $\neg(x \leq y)$ are now satisfied by \mathcal{A} .

Next, we interpret the operators inf and sup in $\langle A, \leq^{\mathcal{A}} \rangle$ by putting for $[x], [y] \in A$

$$\begin{aligned} inf^{\mathcal{A}}([x], [y]) &= \text{greatest lower bound of } [x] \text{ and } [y] \text{ in } \langle A, \leq^{\mathcal{A}} \rangle \\ sup^{\mathcal{A}}([x], [y]) &= \text{least upper bound of } [x] \text{ and } [y] \text{ in } \langle A, \leq^{\mathcal{A}} \rangle. \end{aligned}$$

Plainly, all literals in H of the form $x = inf(y, z)$ and $x = sup(y, z)$ are now satisfied by \mathcal{A} .

Finally, we extend \mathcal{A} to unary function symbols occurring in H . Let us first put:

$$\begin{aligned} INC(H) &= \{f \mid inc(f) \text{ occurs in } H\}, \\ DEC(H) &= \{f \mid dec(f) \text{ occurs in } H\}, \\ MON(H) &= INC(H) \cup DEC(H). \end{aligned}$$

Then, for each function symbol $f \in MON(H)$, we put

$$f^{\mathcal{A}}([x]) = \begin{cases} \text{lub}\{[z] \mid z = f(y) \text{ and } y \leq x \text{ are in } H\}, & \text{if } f \in INC(H) \\ \text{lub}\{[z] \mid z = f(y) \text{ and } x \leq y \text{ are in } H\}, & \text{if } f \in DEC(H) \end{cases}$$

(Thanks to rules $[\leq .4]$, $[\leq .5]$, $[f.2]$, and $[f.3]$, when both $inc(f)$ and $dec(f)$ occur in H , we have

$$\{[z] \mid z = f(y) \text{ and } y \leq x \text{ are in } H\} = \{[z] \mid z = f(y) \text{ and } x \leq y \text{ are in } H\},$$

so that the ambiguity of the above definition for $f \in INC(H) \cap DEC(H)$ is only apparent.)

Additionally, for each function symbol f occurring in H but such that $f \notin MON(H)$, we put

$$f^{\mathcal{A}}([x]) = \begin{cases} [y] & \text{if } y = f(x) \text{ is in } H \\ \text{lub}\{g^{\mathcal{A}}([x]) \mid g \preceq f \text{ is in } H \text{ and } g \in MON(H)\} & \text{otherwise,} \end{cases}$$

where we agree that $\text{lub } \emptyset = [m]$. (By rule $[f.1]$, if both $y = f(x)$ and $y' = f(x)$ are in H , then $[y] = [y']$, so that even the latter definition is not ambiguous.)

It is not hard to check that the interpretation \mathcal{A} so defined also satisfies all the literals in H involving function symbols, namely those of type $x = f(y)$, $inc(f)$, $dec(f)$, and $f \preceq g$. Such verifications are based on the fact that the set H is saturated with respect to the rules in Table 1 and the closure conditions (C1)–(C3) of the initial **Lmf**^{*}-conjunction Ψ .

Just to exemplify the kind of reasoning involved, we will limit to verify that \mathcal{A} models correctly a literal in H of type $f \preceq g$ in the case in which H contains also the literal $inc(f)$. For this purpose, we will assume that the interpretation \mathcal{A} models correctly all literals in H of type $y = h(z)$, $inc(h)$, and $dec(h)$.

Thus, let the literals $f \preceq g$ and $inc(f)$ occur in H and let $[x] \in A$. We need to show that $f^{\mathcal{A}}([x]) \leq g^{\mathcal{A}}([x])$. We distinguish the following cases:

Case (a): the literal $\text{inc}(g)$ occurs in \mathbf{H} . In this case we have:

$$\begin{aligned} f^{\mathcal{A}}([x]) &= \text{lub}\{[z] \mid z = f(y) \text{ and } y \leq x \text{ are in } \mathbf{H}\} \\ g^{\mathcal{A}}([x]) &= \text{lub}\{[z] \mid z = g(y) \text{ and } y \leq x \text{ are in } \mathbf{H}\}. \end{aligned}$$

Moreover, by the closure condition (C3) and rule $[\preceq.2]$, there exist elements $a_1, \dots, a_k, b_1, \dots, b_k \in A$ such that

- $\{[z] \mid z = f(y) \text{ and } y \leq x \text{ are in } \mathbf{H}\} = \{a_1, \dots, a_k\}$,
- $\{[z] \mid z = g(y) \text{ and } y \leq x \text{ are in } \mathbf{H}\} = \{b_1, \dots, b_k\}$, and
- $a_i \leq^{\mathcal{A}} b_i$, for $i = 1, \dots, k$.

By elementary reasoning in lattice theory, it follows at once that

$$\text{lub}\{a_1, \dots, a_k\} \leq^{\mathcal{A}} \text{lub}\{b_1, \dots, b_k\},$$

i.e., $f^{\mathcal{A}}([x]) \leq g^{\mathcal{A}}([x])$.

Case (b): the literal $\text{dec}(g)$ occurs in \mathbf{H} . Let $y = f(M)$ and $y' = g(M)$ be two literals in \mathbf{H} whose existence is assured by the closure condition (C2). By rule $[\preceq.2]$, \mathbf{H} must contain the literal $y \leq y'$, so that

$$f^{\mathcal{A}}([M]) = [y] \leq^{\mathcal{A}} [y'] = g^{\mathcal{A}}([M]).$$

Therefore, since by rule $[\leq.5]$ the literal $x \leq M$ is in \mathbf{H} , so that $[x] \leq^{\mathcal{A}} [M]$, then by exploiting the fact that the maps $f^{\mathcal{A}}$ and $g^{\mathcal{A}}$ are respectively increasing and decreasing, it follows that

$$f^{\mathcal{A}}([x]) \leq^{\mathcal{A}} f^{\mathcal{A}}([M]) \leq^{\mathcal{A}} g^{\mathcal{A}}([M]) \leq^{\mathcal{A}} g^{\mathcal{A}}([x]).$$

Case (c): neither $\text{inc}(g)$ nor $\text{dec}(g)$ occurs in \mathbf{H} . If \mathbf{H} contains a literal $y = g(x)$ then, by the closure condition (C3) it also contain a literal $y' = f(x)$, so that, by rule $[\preceq.2]$, the literal $y' \leq y$ must also occur in \mathbf{H} . Therefore,

$$f^{\mathcal{A}}([x]) = [y'] \leq^{\mathcal{A}} [y] = g^{\mathcal{A}}([x]).$$

On the other hand, if the term $g(x)$ does not occur in any literal in \mathbf{H} , then we have:

$$g^{\mathcal{A}}([x]) = \text{lub}\{h^{\mathcal{A}}([x]) \mid h \preceq g \text{ is in } \mathbf{H} \text{ and } h \in \text{MON}(\mathbf{H})\},$$

so that $f^{\mathcal{A}}([x]) \leq^{\mathcal{A}} g^{\mathcal{A}}([x])$ follows again.

This concludes the verification that when the literals $f \preceq g$ and $\text{inc}(f)$ occur in \mathbf{H} , then $f \preceq g$ is modeled correctly by the interpretation \mathcal{A} .

Other cases can be proved similarly, yielding

Lemma 4 (Completeness). *Let Ψ be an \mathbf{Lmf}^* -conjunction and let $\mathbf{H} = \text{Closure}(\Psi)$. Then \mathbf{H} is satisfiable if and only if it does not contain any pair of complementary literals $\ell, \neg\ell$.*

Since closures can be computed in quadratic time, we have

Theorem 2. *The satisfiability problem for \mathbf{Lmf}^* is solvable in quadratic time.*

Remark 3. It is to be noticed that as a by-product of the proof of Lemma 4, we actually have a satisfiability test for \mathbf{Lmf}^* -conjunctions which, when run on a satisfiable input conjunction Ψ , it returns an \mathbf{Lmf} -model of Ψ , and not just the answer that Ψ is satisfiable.

In view of Lemmas 1 and 2, we also have:

Theorem 3. *The satisfiability problem for conjunctions of \mathbf{Lmf} -literals is solvable in polynomial time.*

Finally, combining Theorems 1 and 2, we have:

Theorem 4. *The satisfiability problem for \mathbf{Lmf} is solvable.*

4 Complexity

We show that the satisfiability problem for \mathbf{Lmf} is \mathcal{NP} -complete.

Concerning the \mathcal{NP} -hardness, it is enough to show that SAT is polynomial-time reducible to the satisfiability problem for \mathbf{Lmf} .² Let \mathbf{P} be a propositional formula (in conjunctive normal form). To each propositional letter P in \mathbf{P} we associate a distinct variable x_P of the \mathbf{Lmf} -language and define $\mathbf{P}^{\mathbf{Lmf}}$ as the \mathbf{Lmf} -formula obtained by substituting in \mathbf{P} each propositional letter P by the atomic \mathbf{Lmf} -formula $x_P = m$. Plainly, \mathbf{P} is satisfiable by a truth-value assignment if and only if $\mathbf{P}^{\mathbf{Lmf}}$ is satisfiable by an \mathbf{Lmf} -assignment. Moreover, the size of $\mathbf{P}^{\mathbf{Lmf}}$ is of the same order as the size of \mathbf{P} , so that the mapping $\mathbf{P} \mapsto \mathbf{P}^{\mathbf{Lmf}}$ just described is a linear-time reduction of SAT to the satisfiability problem for \mathbf{Lmf} , proving that the latter is \mathcal{NP} -hard.

To establish the membership of the \mathbf{Lmf} -satisfiability problem to \mathcal{NP} , we give the following nondeterministic polynomial test for it. Let Φ be any formula of \mathbf{Lmf} and let $Atoms_\Phi$ be the collection of distinct atomic subformulae of Φ . Nondeterministically, we construct a conjunction Ψ_0 by choosing a literal in $\{At, \neg At\}$, for each atom $At \in Atoms_\Phi$. Next, in linear time, we flatten out the conjunction Ψ_0 and eliminate from it all negative literals of type (2), as shown in Section 2.3. Let Ψ_1 be the conjunction of normalized \mathbf{Lmf} -literals thus obtained. In quadratic time, we can construct an equisatisfiable conjunction $\Psi_2 \in \mathbf{Lmf}^*$, satisfying the closure conditions (C1)–(C3) stated at the end of Section 2.3. Finally, the \mathbf{Lmf} -satisfiability of Ψ_2 can be verified by means of the quadratic satisfiability test described in Section 3. It is not difficult to see that our initial formula Φ is \mathbf{Lmf} -satisfiable if and only if there is a computation of the above procedure which returns a positive answer, so that the \mathbf{Lmf} -satisfiability problem belongs to the class \mathcal{NP} and, therefore, it is \mathcal{NP} -complete.

² We recall that SAT is the well-known satisfiability problem for propositional formulae in conjunctive normal form [3].

5 Conclusion

We presented a practical decision procedure for the unquantified theory of lattices with monotone functions, which can have applications in the assisted verification of mathematical proofs and handling of constraints in advanced declarative programming languages.

Specifically, we considered the unquantified language **Lmf** with the predicates $=$ and \leq , with the operators *inf* and *sup* over terms which may involve also uninterpreted function symbols, with predicates expressing increasing or decreasing monotonicity of functions, and with a predicate \preceq for pointwise functions comparison.

In particular, for a restricted collection of conjunctions, denoted **Lmf**^{*}, we described a quadratic satisfiability test, which yielded immediately a polynomial satisfiability test for conjunctions of normalized **Lmf**-literals. We also provided a nondeterministic quadratic reduction of the satisfiability problem for **Lmf**-formulae to the one for **Lmf**^{*}, which allowed to show the \mathcal{NP} -completeness of the satisfiability problem for **Lmf**-formulae.

Future research will involve the extension of our decision procedure in presence of set variables with the set operators of union, intersection, difference, and singleton, and where the operators *inf* and *sup* are extended also to set variables. We also plan to investigate an extension with non-unary function symbols.

References

1. Domenico Cantone, Alfredo Ferro, Eugenio G. Omodeo, and Jacob T. Schwartz. Decision algorithms for some fragments of analysis and related areas. *Communications on Pure and Applied Mathematics*, 40(3):281–300, 1987.
2. Domenico Cantone, Eugenio G. Omodeo, and Alberto Policriti. The automation of syllogistic. II. Optimization and complexity issues. *Journal of Automated Reasoning*, 6(2):173–187, 1990.
3. Stephen A. Cook. A hierarchy for nondeterministic time complexity. *Journal of Computer and System Sciences*, 7(4):343–353, 1973.
4. Alfredo Ferro, Eugenio G. Omodeo, and Jacob T. Schwartz. Decision procedures for elementary sublanguages of set theory. I. Multi-level syllogistic and some extensions. *Communications on Pure and Applied Mathematics*, 33(5):599–608, 1980.
5. Scott Hazelhurst and Carl-Johan H. Seger. Model checking lattices: Using and reasoning about information orders for abstraction. *Logic Journal of the IGPL*, 7(3):375–411, 1999.
6. Bharat Jayaraman, Mauricio Osorio, and David Plaisted. Theory of partial-order programming. *Science of Computer Programming*, 34(3):207–238, 1999.
7. Frank J. Oles. An application of lattice theory to knowledge representation. *Theoretical Computer Science*, 249(1):163–196, 2000.
8. Alfred Tarski. Undecidability of the theory of lattices and projective geometries. *Journal of Symbolic Logic*, 14:77–78, 1949.