Automated Theorem Proving by Resolution for Finitely-Valued Logics Based on Distributive Lattices with Operators

Viorica Sofronie-Stokkermans
Max-Planck-Institut für Informatik,
Im Stadtwald, D-66123 Saarbrücken, Germany
email: sofronie@mpi-sb.mpg.de

To appear in Multiple-Valued Logic - An International Journal

Abstract

In this paper we present a method for automated theorem proving in many-valued logics whose algebra of truth values is a finite distributive lattice with operators. This class of many-valued logics includes many logics that occur in a natural way in applications. The method uses the Priestley dual of the algebra of truth values instead of the algebra itself; this dual is used as a finite set of possible worlds. We first present a procedure that constructs, for every formula ϕ in the language of such a logic, a set Φ of signed clauses such that ϕ is a theorem if and only if Φ is unsatisfiable. Compared to related approaches, the method presented here leads in many cases to a reduction of the number of clauses that are generated, especially when the set of truth values is not linearly ordered. We then discuss several possibilities for checking the unsatisfiability of Φ, among which a version of signed hyperresolution, and give several examples.

Keywords: Automated theorem proving, resolution, many-valued logics, distributive lattices, Priestley duality.

Presented by Reiner Hähnle. Submitted February 12, 1999; Revised October 4, 1999; Revised version sent to the managing editor of "Multiple-Valued Logic" on December 7, 1999.

1 Introduction

Recently, a growing number of new logics has found applications in computer science, linguistics and related areas. Therefore, there is a strong demand for efficient automatic proof procedures for non-classical logics, especially for many-valued logics.

In classical logic, a common approach to automated theorem proving is resolution and its refinements. In non-classical logic, the specific techniques and the efficiency of the methods depend very much on the particular logic, and on the way this logic is described: by giving its semantics (e.g. a set of truth values with the appropriate truth tables, or, more generally, a class of models), or by means of a logical calculus (e.g. Hilbert-style or Gentzen-style). The simplest example of logics described by their semantics is formed by the many-valued logics, and we will only refer to this kind of logics in what follows. Many-valued logics are sound and complete with respect to a given model, viewed as a set of truth values.

The methods for automated theorem proving currently used in many-valued first-order logics can roughly be classified as *clausal* and *non-clausal*. We start with some words on existing work on *clausal* theorem proving. One of the most general methods for automated theorem proving in finitely-valued logics is that of Baaz and Fermüller [Baa92, BF92, BF95]. It is based on a notion of "signed literal" (extending the notion of literal from classical logic) where truth values are used as signs. The unsatisfiability of such sets of signed literals is checked in [BF95] with a many-valued resolution procedure, which is an extension of classical resolution. The method of Baaz and Fermüller is very general, being suitable for all finitely-valued logics; for clause generation only the truth-tables of the operators and the definition of the quantifiers are used. Results along similar lines appear in the work of Hähnle [Häh93, Häh94, Häh96b], who, instead of single truth values, uses sets of truth values as signs for the literals, and defines various versions of signed resolution. For more details on signed resolution we refer to [BFS99]. We also refer to the work of Kifer and Lozinskii [KL92] and Lu, Murray and Rosenthal [LMR93, LMR98], where signed resolution is discussed, also in the context of annotated logic.

Concerning *non-clausal* methods in many-valued logics, here we only mention tableaux methods such as the method of Surma [Sur84], further developed by Carnielli [Car87], see also [Car91], the method due to Suchoń [Suc74] for the special case of n-valued logics, and the method of Hähnle who defined a generalized notion of signs in tableaux that makes it possible to speak concisely about the truth values a formula can take at a certain stage during the construction of the tableau [Häh90, Häh91, Häh93]. Recently, a strong interdependence between tableaux-based methods and resolution-based methods in many-valued logics emerged. Ideas from tableaux-based theorem proving, such as the use of sets as signs, turned out to be also useful in theorem proving by resolution [Häh94, Häh96b].

The main goal of this paper is to show that, in many cases, the complexity of resolution-based automated theorem proving procedures can be improved by exploiting the finer properties of the finitely-valued logics (in particular the algebraic properties of their algebra of truth values). In what follows, as a measure for the complexity of the resolution procedure, we consider:

- the number of clauses generated in the process of translation to clause form,
- the number of clauses generated in the process of actual proving by resolution.

We show that if the set of truth values is a finite distributive lattice with unary operators then improvements can be made in both these directions. The main idea is to use, instead of the algebra A of truth values, its Priestley dual $D(A)$ (seen as a particular kind of Kripke-style model). If A is finite, $D(A)$ is in bijective correspondence with the set of join-irreducible elements of A, hence has fewer elements than A. Based on this, we give a method that for every formula ϕ in the language of such a logic constructs a set Φ of signed clauses such that ϕ is a theorem if and only if Φ is unsatisfiable. First, as expected, it turns out that in many of the cases we consider fewer clauses will be generated than with the very general procedure described in [BF95] or with general procedures using sets-as-signs. Moreover our results show that only signs of a certain type are needed in the clause form, namely only signs of the type $\uparrow i$ and complements thereof, where i is a *join-irreducible* element of the lattice of truth values, and $\uparrow i = \{j \in A \mid j \geq i\}$. Second, the fact that the signs of the clauses all have the special form mentioned above, allows a reduction of the number of clauses that may be generated in the resolution process compared to other approaches that use sets-as-signs, and even with approaches such as [KL92] where the signs are $\uparrow i$ or complements thereof, where i is an *arbitrary* element of the lattice of truth values. Moreover, we show that certain refinements of classical resolution such as hyperresolution (which have been extended to regular logics in [Häh94, Häh96b]) can be extended to sets of signed clauses of the type studied in this paper.

The assumption that the set of truth values is a distributive lattice with operators is reasonable for many practical applications: in many cases the algebras of truth values associated to non-classical logics are of this type. Several such logics (SH_n-logics, P_{mn}-logics, SHK_n-logics) will be mentioned as examples in this paper. These quite general classes of logics contain as subclasses some logics which are interesting in applications, such as the Kleene logics (SHK_3-logic) or a (well-behaved) fragment of relevance logic (P_{20}-logic). Moreover, Iturrioz [Itu95] noticed that SH_n-logics are instances of L_{t_f}-logics, which were introduced by Rasiowa (see e.g. [Ras88]) to formalize the reasoning of a poset of intelligent agents (for further details see also [Nou97]).

The paper is structured as follows. In Section 2 we briefly present the fundamental notions used in the paper. We continue with a presentation of the Priestley representation for bounded distributive lattices with operators (abbreviated DLO) in Section 3. In Section 4 we use these results to define models based on the Priestley dual of the algebra of truth values (for finitely-valued logics based on DLO). In Section 5 we present an approach to automated theorem proving by resolution based on this idea. The methods are illustrated in Section 6 by several examples: SH_n-logics, SHK_n-logics, and P_{mn}-logics. In Section 7 comparisons with other existing approaches to automated theorem proving in non-classical logics are made. Section 8 contains some concluding remarks and plans for future work.

2 Preliminaries

This section contains a brief presentation of the main notions needed in this paper.

2.1 Partially-ordered sets, lattices, universal algebra

In what follows we assume known standard notions, such as partially-ordered set, order-filter (or upwards-closed set) and order-ideal (or downwards-closed set) in a partially-ordered set. For the definitions and further information we refer to [DP90] or to [SS00a]. If $\mathbf{P} = (P, \leq)$ is a partially-ordered set, we denote by $\mathcal{O}(\mathbf{P})$ the set of order-filters of $\mathbf{P}$; if $i \in P$, we will use the notations $\uparrow i := \{j \in P \mid j \geq i\}$, and $\downarrow i := \{j \in P \mid i \geq j\}$. A *lattice* is a partially-ordered set $\mathbf{L} = (L, \leq)$ with the property that every two elements $x, y \in L$ have a supremum (denoted $x \vee y$) and an infimum (denoted by $x \wedge y$) in L. Alternatively, a non-empty set L together with two binary operations $\vee$ and $\wedge$ on L is called *lattice* if $\vee$ and $\wedge$ are associative, commutative and idempotent and satisfy the absorption laws. A *distributive lattice* is a lattice that satisfies either of the distributive laws. We say that a lattice $\mathbf{L}$ has a *first (smallest) element* if there is an element $0 \in L$ such that $0 \leq x$ for every $x \in L$. A lattice $\mathbf{L}$ has a *last (greatest) element* if there is an element $1 \in L$ such that $x \leq 1$ for every $x \in L$. A lattice having both a first and a last element is called *bounded*. If not specified otherwise, in what follows we will denote the first element in a lattice (if any) by 0 and the last element (if any) by 1. Let $\mathbf{L} = (L, \leq)$ be a lattice. An element x of L is called *join-irreducible* if $x \neq 0$ (in case L has a first element) and whenever $x = y \vee z$, either $x = y$ or $x = z$. *Meet-irreducible* elements are defined dually. An *ideal* of $\mathbf{L}$ is a non-empty downwards-closed set closed under join, and a *filter* is a non-empty upwards-closed set closed under meet. An ideal I of $\mathbf{L}$ is said to be *prime* if $I \neq L$ and for every $x, y \in L$, if $x \wedge y \in I$ then $x \in I$ or $y \in I$. A filter F of $\mathbf{L}$ is said to be *prime* if $F \neq L$ and for every $x, y \in L$, if $x \vee y \in F$ then $x \in F$ or $y \in F$.

For the basic notions of universal algebra needed in what follows, we refer for instance to [BS81]. For every signature Σ and every arity function $a : \Sigma \to \mathbb{N}$, a Σ-algebra is a structure $(A, \{\sigma_A\}_{\sigma \in \Sigma})$, where for every $\sigma \in \Sigma$, $\sigma_A : A^{a(\sigma)} \to A$. If the signature Σ is assumed to be known we will sometimes use the notation A for the Σ-algebra $(A, \{\sigma_A\}_{\sigma \in \Sigma})$. We say that a Σ-algebra has a *distributive lattice reduct* if there exist operations $\vee, \wedge$ in Σ such that $(A, \vee_A, \wedge_A)$ is a distributive lattice. We say that a Σ-algebra has a *bounded distributive lattice reduct* if it has a distributive lattice reduct, and, moreover, there exist constants $0, 1 \in \Sigma$ such that 0_A is a first element in A and 1_A is a last element in A. The absolutely free algebra over a given signature Σ freely generated by a set X will be called the algebra of terms in X over Σ and will be denoted $\mathsf{Term}_\Sigma(X)$. If $t_1, t_2 \in \mathsf{Term}_\Sigma(X)$, and A is a Σ-algebra, we say that A satisfies the identity $t_1 = t_2$ (denoted in what follows by $A \overset{a}{\models} t_1 = t_2$) if for every assignment $f : X \to A$, $\overline{f}(t_1) = \overline{f}(t_2)$, where $\overline{f} : \mathsf{Term}_\Sigma(X) \to A$ is the unique morphism of Σ-algebras that extends f.

2.2 Finitely-valued logics

Since in this paper we mention both propositional and first-order many-valued logics, we now briefly present a semantical definition of first-order many-valued logics having a set A of truth values. We closely follow Baaz and Fermüller [BF95]. In Section 4 we will define a semantics that uses the Priestley dual $D(A)$ of A in the case when A is a bounded distributive lattice with operators.

Syntax: Let $\mathbf{L} = (X, O, P, \Sigma, \{Q_1, \dots, Q_k\})$ be a first-order language consisting of a (countably) infinite set X of variables, a set O of function symbols, a set P of predicate symbols, a set Σ of logical operators, and a finite set of (one-place) quantifiers $Q_1, \dots, Q_k$. Terms, ground terms, atomic formulae and formulae are defined in the usual way. We will denote by $\mathsf{Term}_O(X)$ (resp. Term^0_O) the set of all terms (resp. ground terms) of $\mathcal{L}$, by $\mathsf{At}_{\mathbf{L}}(X)$ the set of atomic formulae in the language $\mathbf{L}$, and by $\mathsf{Fma}(\mathbf{L})$ the set of all formulae in the language $\mathbf{L}$.

Semantics[1]**:** Let A be a finite set of truth values. We associate truth functions with logical operators and quantifiers as follows:

- to every $\sigma \in \Sigma$ with arity n we associate a truth function $\sigma_A : A^n \to A$,
- to every quantifier Q we associate a truth function $\overline{Q} : \mathcal{P}(A) \backslash \{\emptyset\} \to A$.

These truth functions can be regarded as truth tables in A for the logical connectives and the quantifiers.

Definition 1 *A many-valued logic with language* $\mathbf{L}$ *and set of truth values* A *is a pair* $\mathcal{L} = (\mathbf{L}, (A, \{\sigma_A\}_{\sigma \in \Sigma}, \{\overline{Q_i}\}_{i=1,\dots,k}))$ *consisting of a first-order language* $\mathbf{L} = (X, O, P, \Sigma, \{Q_1, \dots, Q_k\})$ *and a set of truth values* A *endowed with truth functions for all logical operators and quantifiers in* $\mathbf{L}$.

In what follows let $\mathcal{L}$ be a many-valued logic with language $\mathbf{L} = (X, O, P, \Sigma, \{Q_1, \dots, Q_k\})$ and set of truth values A.

Definition 2 (Frame, Interpretation) *A* frame *for* $\mathcal{L}$ *is a pair* (D, I) *where:*

[1] The idea to generalize the algebraic treatment of formulae of propositional calculi to formulae of predicate calculi goes back to the work of Mostowski [Mos48], who introduced an algebraic interpretation of formulae of intuitionistic calculi (the quantifiers $\forall$ and $\exists$ are interpreted as greatest lower bound resp. least upper bound). The approach in [BF95] extends Mostowski's in that arbitrary quantifiers (not only $\forall$ and $\exists$) are considered, and they are interpreted as maps from $\mathcal{P}(A) \backslash \{\emptyset\}$ to A.

(1) D is a non-empty set, the domain, *and*

(2) I is a signature interpretation, i.e. a function assigning a function $I(f) : D^n \to D$ to every n-ary function symbol $f \in O$, and a function $I(R) : D^n \to A$ to every n-ary predicate symbol $R \in P$.

An interpretation $\mathcal{I}$ *for* $\mathcal{L}$ *(or* interpretation for $\mathbf{L}$ in A*) is a triple (D, I, d) where (D, I) is a frame and d is a* variable assignment $d : X \to D$.

Every interpretation $\mathcal{I} = (D, I, d)$ extends in a canonical way to terms, and induces a valuation function on formulae, $v_{\mathcal{I}} : \mathsf{Fma}(\mathbf{L}) \to A$, as follows:

(1) $v_{\mathcal{I}}(x) = d(x)$ for all variables $x \in X$,

(2) $v_{\mathcal{I}}(f(t_1, \ldots, t_n)) = I(f)(v_{\mathcal{I}}(t_1), \ldots, v_{\mathcal{I}}(t_n))$ for all n-ary $f \in O$, $n \geq 0$,

(3) $v_{\mathcal{I}}(R(t_1, \ldots, t_n)) = I(R)(v_{\mathcal{I}}(t_1), \ldots, v_{\mathcal{I}}(t_n))$ for all n-ary $R \in P$, $n \geq 0$,

(4) $v_{\mathcal{I}}(\sigma(\phi_1, \ldots, \phi_n)) = \sigma_A(v_{\mathcal{I}}(\phi_1), \ldots, v_{\mathcal{I}}(\phi_n))$ for all n-ary $\sigma \in \Sigma$,

(5) $v_{\mathcal{I}}((Qx)\phi) = \overline{Q}(\{w \mid \exists d \in D \text{ s.t. } v_{\mathcal{I}_{x,d}}(\phi) = w\})$ for all quantifiers Q, where $\mathcal{I}_{x,d}$ is identical to $\mathcal{I}$ except for assigning d to the variable x.

Assume that a subset A_d of A of *designated truth values* for the logic $\mathcal{L}$ is additionally specified. A formula ϕ is valid in a logic $\mathcal{L}$ (with set A_d of designated truth values) if and only if $v_{\mathcal{I}}(\phi) \in A_d$ for all interpretations $\mathcal{I}$ for the language of $\mathcal{L}$ in A. A formula ϕ is *satisfiable* in $\mathcal{L}$ if and only if there is an interpretation $\mathcal{I}$ with $v_{\mathcal{I}}(\phi) \in A_d$.

In many concrete many-valued logics, the algebra of truth values A has a bounded lattice reduct, hence it contains an element 1, which is its greatest element with respect to the lattice ordering. This is the situation we will analyze in this paper. In this context, we will use a slightly more restrictive notion of validity, resp. satisfiability: we will assume that $A_d = \{1\}$.

Assume that $(A, \{\sigma_A\}_{\sigma \in \Sigma})$ is an algebra with a bounded lattice reduct, and let 1 be its greatest element.

Definition 3 (Validity, Satisfiability) *A formula ϕ is valid in $\mathcal{L}$ if and only if $v_{\mathcal{I}}(\phi) = 1$ for all interpretations $\mathcal{I}$ for the language of $\mathcal{L}$ in A. A formula ϕ is* satisfiable *in $\mathcal{L}$ if and only if there is an interpretation $\mathcal{I}$ for the language of $\mathcal{L}$ in A with $v_{\mathcal{I}}(\phi) = 1$.*

Note that the propositional case is encompassed by the definitions above: if all predicate symbols are supposed to be 0-ary, then they can be regarded as propositional variables. Let $\mathcal{L}$ be a logic with a set Var of propositional variables (0-ary predicate symbols). If the set Σ of logical connectives is considered known, instead of $\mathsf{Fma}(\mathbf{L})$ we will use the notation $\mathsf{Fma}(\mathsf{Var})$. $\mathsf{Fma}(\mathsf{Var})$ is the absolutely free Σ-algebra freely generated by Var. Interpretations are in this case assignments $I : \mathsf{Var} \to A$; the induced valuation functions $v_{\mathcal{I}} : \mathsf{Fma}(\mathsf{Var}) \to A$ are obtained by the universality property of $\mathsf{Fma}(\mathsf{Var})$. It is easy to see that a formula ϕ is valid in a propositional logic $\mathcal{L}$ if and only if the equation $\phi = 1$ (where ϕ is regarded as a term) is satisfied by $(A, \{\sigma_A\}_{\sigma \in \Sigma})$ (seen as a Σ-algebra).

3 Priestley representation

Priestley representation. The Priestley representation theorem states that every bounded distributive lattice A is isomorphic to the lattice of clopen (i.e. both closed and open) order filters of the ordered topological space $D(A)$ having as points the prime filters of A, ordered by inclusion, and the topology generated by the sets of the form $X_a = \{F \mid F$ prime filter $a \in F\}$ and their complements as a subbasis. The set of prime filters of A will be also denoted $D(A)$. In what follows, we will refer to $D(A)$ as the Priestley dual of A. The Priestley representation has been extended to distributive pseudocomplemented lattices and Heyting algebras (cf. e.g. [Pri84], [Gol89]). The dual spaces are in this case Priestley spaces $(X, \leq, \tau)$ satisfying the condition that for every clopen order-filter U, $X \setminus \downarrow U$ is clopen.

Finite distributive lattices. For a finite distributive lattice A, its Priestley dual $D(A)$ is of the form $\{\uparrow b \mid b \in A$, join-irreducible$\}$, with the order given by set inclusion, and the discrete topology. Thus, A is isomorphic to the lattice of order-filters of $D(A)$; the map $\eta_A : A \to \mathcal{O}(D(A))$ defined for every $a \in A$ by

$$\eta_A(a) = \{\uparrow b \in D(A) \mid a \in \uparrow b\}$$

is an isomorphism of bounded distributive lattices. (This result is also known as Birkhoff's representation theorem for finite distributive lattices.)

Other operations such as pseudocomplementation[2] and relative pseudocomplementation[3] can be treated in a similar way. For every $U, U_1, U_2 \in \mathcal{O}(D(A))$ the pseudocomplement of U, $\neg U$, resp. the pseudocomplement of U_1 relative to U_2, $U_1 \Rightarrow U_2$, can be defined on $\mathcal{O}(D(A))$ by:

$$\neg U = \{\alpha \in D(A) \mid \forall \beta (\text{if } \beta \geq \alpha \text{ then } \beta \notin U)\};$$

$$U_1 \Rightarrow U_2 = \{\alpha \in D(A) \mid \forall \beta \geq \alpha, \text{ if } \beta \in U_1 \text{ then } \beta \in U_2\}.$$

It can be shown that if $(A, \vee, \wedge, 0, 1)$ is a finite bounded distributive lattice and $\neg$ and $\Rightarrow$ a pseudocomplementation resp. a relative pseudocomplementation on A, then $\eta_A(a \Rightarrow b) = \eta_A(a) \Rightarrow \eta_A(b)$, and $\eta_A(\neg a) = \neg \eta_A(a)$, i.e. $\eta_A : (A, \vee, \wedge, \Rightarrow, \neg, 0, 1) \to (\mathcal{O}(D(A)), \cup, \cap, \Rightarrow, \neg, \emptyset, D(A))$ is a $\{\vee, \wedge, \Rightarrow, \neg, 0, 1\}$-isomorphism. For further details we refer e.g. to [DP90], [Gol89].

3.1 Extension to distributive lattices with operators

The Priestley representation can be extended to distributive lattices endowed with operators which preserve part of the lattice structure. The definitions of the operators we will consider are given below.

Definition 4 *Let A be an algebra with a bounded distributive lattice reduct.*
A lattice antimorphism *on A is a map $k : A \to A$ such that $k(0) = 1, k(1) = 0, k(a_1 \vee a_2) = k(a_1) \wedge k(a_2)$ and $k(a_1 \wedge a_2) = k(a_1) \vee k(a_2)$.*
A join hemimorphism *on A is a map $f : A^n \to A$ such that for every $i, 1 \leq i \leq n$,*

$(Jh1)$ $f(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) = 0,$
$(Jh2)$ $f(a_1, \dots, a_{i-1}, b_1 \vee b_2, a_{i+1}, \dots, a_n) = f(a_1, \dots, a_{i-1}, b_1, a_{i+1}, \dots, a_n) \vee f(a_1, \dots, a_{i-1}, b_2, a_{i+1}, \dots, a_n).$

A meet hemimorphism *on A is a map $g : A^n \to A$ such that for every $i, 1 \leq i \leq n$,*

$(Mh1)$ $g(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n) = 1,$
$(Mh2)$ $g(a_1, \dots, a_{i-1}, b_1 \wedge b_2, a_{i+1}, \dots, a_n) = g(a_1, \dots, a_{i-1}, b_1, a_{i+1}, \dots, a_n) \wedge g(a_1, \dots, a_{i-1}, b_2, a_{i+1}, \dots, a_n).$

A join hemiantimorphism *on A is a map $f : A^n \to A$ such that for every $i, 1 \leq i \leq n$,*

$(Ja1)$ $f(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n) = 0,$

[2] The *pseudocomplement* of an element $a \in L$ (denoted by $\neg a$) is the largest element of $\{c \in L \mid a \wedge c = 0\}$ (if any). In a finite lattice every element has a pseudocomplement.

[3] Given $a, b \in L$, the *pseudocomplement of a relative to b* (denoted by $a \Rightarrow b$) is the largest element of $\{c \in L \mid a \wedge c \leq b\}$ (if any). In a finite lattice such a largest element does always exist.

(*Ja*2) $f(a_1, \ldots, a_{i-1}, b_1 \wedge b_2, a_{i+1}, \ldots, a_n) = f(a_1, \ldots, a_{i-1}, b_1, a_{i+1}, \ldots, a_n) \vee f(a_1, \ldots, a_{i-1}, b_2, a_{i+1}, \ldots, a_n)$.

A meet hemiantimorphism *on* A *is a map* $g : A^n \to A$ *such that for every* $i, 1 \leq i \leq n$,

(*Ma*1) $g(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) = 1$,

(*Ma*2) $g(a_1, \ldots, a_{i-1}, b_1 \vee b_2, a_{i+1}, \ldots, a_n) = g(a_1, \ldots, a_{i-1}, b_1, a_{i+1}, \ldots, a_n) \wedge g(a_1, \ldots, a_{i-1}, b_2, a_{i+1}, \ldots, a_n)$.

Definition 5 *Let* $(X, \leq)$ *be a partially ordered set and let* $R \subseteq X^{n+1}$ *be a* $(n+1)$*-ary relation on* X. R *is an* increasing relation *if it has the property that for all* $\overline{x} \in X^n$ *and every* $y, z \in X$, *if* $R(\overline{x}, y)$ *and* $y \leq z$ *then* $R(\overline{x}, z)$. R *is a* decreasing relation *if it has the property that for all* $\overline{x} \in X^n$ *and every* $y, z \in X$, *if* $R(\overline{x}, y)$ *and* $z \leq y$ *then* $R(\overline{x}, z)$.

Goldblatt [Gol89] extended the Priestley representation for bounded distributive lattices to bounded distributive lattices having as operators join and meet hemimorphisms. These results were extended in [SS00a] to larger classes of operators, including lattice antimorphisms, and join resp. meet antihemimorphisms. For the sake of simplicity, and since the logics studied in this paper have finite sets of truth values, in Sections 3.1.1 and 3.1.2 we will illustrate how this can be done for *finite distributive lattices with operators.*

In what follows, Σ will be a signature containing operation symbols corresponding to the classes mentioned above. In order to distinguish these classes, we will write $\Sigma = Lh \cup La \cup Jh \cup Mh \cup Ja \cup Ma$, where Lh, La, Jh, Mh, Ja, and Ma may be empty. Let DLO_Σ be the class of all $\{0, 1, \vee, \wedge\} \cup \Sigma$-algebras

$$(A, \vee, \wedge, 0, 1, \{h_A\}_{h \in Lh}, \{k_A\}_{k \in La}, \{f_A\}_{f \in Jh}, \{g_A\}_{g \in Mh}, \{f'_A\}_{f' \in Ja}, \{g'_A\}_{g' \in Ma})$$

such that (i) $(A, \vee, \wedge, 0, 1)$ is a bounded distributive lattice, and (ii) for every $h \in Lh$, $k \in La$, $f \in Jh$, $g \in Mh$, $f' \in Ja$, $g' \in Ma$: h_A is a lattice morphism, k_A is a lattice antimorphism, f_A is a join hemimorphism, g_A is a meet hemimorphism, f'_A is a join hemiantimorphism, and g'_A is a meet hemiantimorphism.

Notation. As a convention, (if not explicitly specified otherwise) the name h will be used for an operation symbol in Lh, k for an operation symbol in La, f for one in Jh, g for one in Mh, f' for one in Ja, and g' for one in Ma. Similarly, on the corresponding Priestley spaces, we will denote by H (resp. K) the map corresponding to operation symbols in Lh (resp. La), by R (resp. R') the relation corresponding to a symbol in Jh (resp. Ja), and by Q (resp. Q') the relation corresponding to a symbol in Mh (resp. Ma).

3.1.1 Canonical definition of maps and relations on $D(A)$

Every operator on a lattice A which is a lattice (anti)homomorphism or a join or meet (anti)hemimorphism defines in a canonical way a map or, respectively, relation on $D(A)$, as shown in the following theorem.

Theorem 1 (cf. [SS00a]) *Let A be a bounded distributive lattice. Then:*

(1) *If* $h : A \to A$ *is a lattice homomorphism, then the map* $D(h) : D(A) \to D(A)$, *defined for every* $F \in D(A)$ *by* $D(h)(F) = h^{-1}(F)$, *is order-preserving.*

If $k : A \to A$ *is a lattice antimorphism then the map* $D(k) : D(A) \to D(A)$, *defined for every* $F \in D(A)$ *by* $D(k)(F) = A \backslash k^{-1}(F)$, *is order-reversing.*

(2) If $f : A^n \to A$ is a join hemimorphism, $f' : A^n \to A$ a join hemiantimorphism, $g : A^n \to A$ a meet hemimorphism, and $g' : A^n \to A$ a meet hemiantimorphism, then $D(f), D(f')$ are increasing and $D(g), D(g')$ are decreasing relations, where $D(f), D(f'), D(g), D(g') \subseteq D(A)^{n+1}$ are defined by:

$$D(f)(F_1, \ldots, F_n, F_{n+1}) \text{ iff for all } a_1, \ldots, a_n, \text{ if } a_1 \in F_1, \ldots, a_n \in F_n \text{ then } f(a_1, \ldots, a_n) \in F_{n+1};$$

$$D(f')(F_1, \ldots, F_n, F_{n+1}) \text{ iff for all } a_1, \ldots, a_n, \text{ if } a_1 \notin F_1, \ldots, a_n \notin F_n \text{ then } f'(a_1, \ldots, a_n) \in F_{n+1};$$

$$D(g)(F_1, \ldots, F_n, F_{n+1}) \text{ iff for all } a_1, \ldots, a_n, \text{ if } g(a_1, \ldots, a_n) \in F_{n+1} \text{ then there exists } i \in \{1, \ldots, n\} \text{ with } a_i \in F_i;$$

$$D(g')(F_1, \ldots, F_n, F_{n+1}) \text{ iff for all } a_1, \ldots, a_n, \text{ if } g'(a_1, \ldots, a_n) \in F_{n+1} \text{ then there exists } i \in \{1, \ldots, n\} \text{ with } a_i \notin F_i.$$

In particular, if $A \in \mathsf{DLO}_\Sigma$, then each of the operators in Σ on A induces a function resp. relation on $D(A)$. If $h \in Lh$, then $D(h_A)$ is an order-preserving map; if $k \in La$, then $D(k_A)$ is an order-reversing map; if $f \in Jh \cup Ja$, then $D(f_A)$ is an increasing relation, and if $g \in Mh \cup Ma$, then $D(g_A)$ is a decreasing relation.

3.1.2 Canonical definition of operators on $\mathcal{O}(D(A))$

We saw that the operations on $D(A)$ defined in a canonical way from the operators on A are: order-preserving maps (Lh), order-reversing maps (La), increasing relations (Jh, Ja), and decreasing relations (Mh, Ma). We now show how, given a finite partially ordered set $\mathbf{X} = (X, \leq)$, relations on X satisfying the conditions above induce in a canonical way operators on $\mathcal{O}(\mathbf{X})$. If $\mathbf{X} = D(A)$ we obtain, in particular, a way of defining such operators on $\mathcal{O}(D(A))$.

Theorem 2 (cf. [SS00a]) *Let $\mathbf{X} = (X, \leq)$ be a finite partially-ordered set. Then:*

(1) If $H : X \to X$ is an order-preserving map, then $h_{\mathcal{O}(X)} : \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$, defined for every $U \in \mathcal{O}(\mathbf{X})$ by $h_{\mathcal{O}(X)}(U) = H^{-1}(U)$ is a lattice homomorphism.

If $K : X \to X$ is an order-reversing map, then $k_{\mathcal{O}(X)} : \mathcal{O}(\mathbf{X}) \to \mathcal{O}(\mathbf{X})$, defined for every $U \in \mathcal{O}(\mathbf{X})$ by $k_{\mathcal{O}(X)}(U) = X \backslash K^{-1}(U)$ is a lattice antimorphism.

(2) If $R \subseteq X^{n+1}$ is an increasing relation and $Q \subseteq X^{n+1}$ a decreasing relation, then the maps $f_{\mathcal{O}(X)}, f'_{\mathcal{O}(X)}, g_{\mathcal{O}(X)}, g'_{\mathcal{O}(X)} : \mathcal{O}(\mathbf{X})^n \to \mathcal{O}(\mathbf{X})$, defined by:

$$f_{\mathcal{O}(X)}(U_1, \ldots, U_n) = \{x \in X \mid \exists x_1 \in U_1, \ldots, x_n \in U_n \text{ s.t. } R(x_1, \ldots, x_n, x)\};$$

$$f'_{\mathcal{O}(X)}(U_1, \ldots, U_n) = \{x \in X \mid \exists x_1 \notin U_1, \ldots, x_n \notin U_n \text{ s.t. } R(x_1, \ldots, x_n, x)\};$$

$$g_{\mathcal{O}(X)}(U_1, \ldots, U_n) = \{x \in X \mid \forall x_1 \in X, \ldots, x_n \in X (Q(x_1, \ldots, x_n, x) \to \exists i, x_i \in U_i)\};$$

$$g'_{\mathcal{O}(X)}(U_1, \ldots, U_n) = \{x \in X \mid \forall x_1 \in X, \ldots, x_n \in X (Q(x_1, \ldots, x_n, x) \to \exists i, x_i \notin U_i)\}$$

have the following properties: $f_{\mathcal{O}(X)}$ is a join hemimorphism, $f'_{\mathcal{O}(X)}$ is a join hemiantimorphism, $g_{\mathcal{O}(X)}$ is a meet hemimorphism, and $g'_{\mathcal{O}(X)}$ is a meet hemiantimorphism.

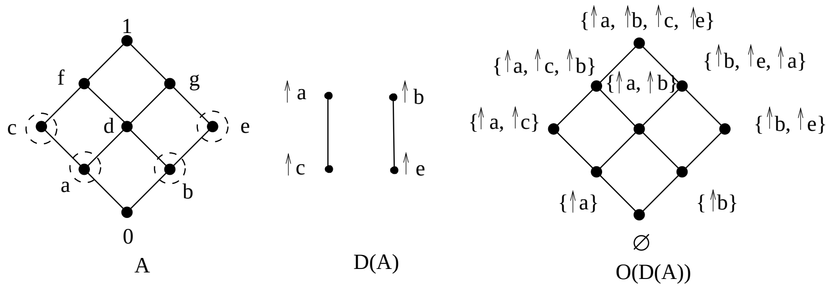

Figure 1: Example 1

3.1.3 Priestley representation for finite distributive lattices with operators

Let A be a finite distributive lattice in DLO_Σ, and let $D(A)$ be its Priestley dual on which functions and relations are defined as described in Section 3.1.1. We will denote by $\mathcal{O}(D(A))$ the algebra $(\mathcal{O}(D(A)), \cup, \cap, \emptyset, D(A), \{h_{\mathcal{O}(D(A))}\}_{h\in\Sigma})$, where for every $h \in \Sigma$, $h_{\mathcal{O}(D(A))}$ is defined from $D(h_A)$ as explained in Section 3.1.2. With the above notation, the following holds.

Theorem 3 (cf. [SS00a]) *For every finite $A \in \mathsf{DLO}_\Sigma$, $\mathcal{O}(D(A)) \in \mathsf{DLO}_\Sigma$, and the map $\eta_A : A \to \mathcal{O}(D(A))$, defined by $\eta_A(a) = \{\uparrow b \in D(A) \mid a \in \uparrow b\}$ is an isomorphism of $\{\vee, \wedge, 0, 1\} \cup \Sigma$-algebras.*

We illustrate the ideas above by two examples.

Example 1 *Let A be the finite lattice represented in the left side of Figure 1, with an additional operator $\sim : A \to A$, a lattice antimorphism defined by:*

$$\sim 0 = 1,\ \sim 1 = 0,\ \sim a = f,\ \sim f = a,\ \sim b = g,\ \sim g = b,\ \sim c = c,\ \sim e = e,\ \sim d = d.$$

The Priestley dual of A, $D(A)$, is also represented in Figure 1. $D(A)$ has as elements the principal filters generated by the join-irreducible elements of A (encircled in Figure 1) with the order given by the inclusion. It is additionally endowed with a map $D(\sim)$ induced by $\sim$, defined by $D(\sim)(F) = A \backslash \sim^{-1}(F)$, i.e. such that

$$D(\sim)(\uparrow a) = \uparrow e,\ D(\sim)(\uparrow e) = \uparrow a,\ D(\sim)(\uparrow b) = \uparrow c,\ D(\sim)(\uparrow c) = \uparrow b.$$

$D(\sim)$ induces a map $\sim_{\mathcal{O}(D(A))}$ on $\mathcal{O}(D(A))$, defined for every $U \in \mathcal{O}(D(A))$ by $\sim_{\mathcal{O}(D(A))}(U) = D(A) \backslash D(\sim)^{-1}(U)$. It can easily be checked that the algebra $\mathcal{O}(D(A))$ obtained this way is isomorphic to A.

Example 2 *Consider the lattice A represented in Figure 2, on which an additional operator $\neg$ is defined, such that for every $a \in A$, $\neg a$ is the largest element of A such that $a \wedge \neg a = 0$ (the pseudocomplement of a). Thus:*

$$\neg 0 = 1, \neg a = b, \neg b = a, \neg c = \neg 1 = 0.$$

The join-irreducible elements of A are encircled in Figure 2. The Priestley dual $D(A)$ of A, is represented in Figure 2; so is $\mathcal{O}(D(A))$. Note that a pseudocomplementation on $\mathcal{O}(D(A))$, $\neg_{\mathcal{O}(D(A))}$, can be defined for every $U \in \mathcal{O}(D(A))$ by $\neg_{\mathcal{O}(D(A))} U = \{\alpha \in D(A) \mid \forall \beta \geq \alpha, \beta \notin U\}$. It can easily be checked that the algebra $\mathcal{O}(D(A))$ obtained this way is isomorphic to A.

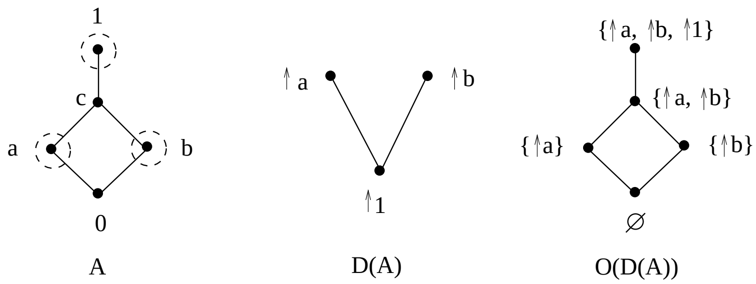

Figure 2: Example 2

4 Models based on the Priestley dual

Let $\mathcal{L}$ be a finitely-valued logic whose algebra A of truth values is a distributive lattice with operators in the classes discussed above. In this section we present a way of defining a notion of satisfiability for formulae of $\mathcal{L}$ with respect to the Priestley dual of A, $D(A)$, regarded as a special kind of Kripke-style frame.

We start with propositional logics and then also consider first-order finitely-valued logics.

In what follows $\Sigma = Lh \cup La \cup Jh \cup Mh \cup Ja \cup Ma$, where Lh, La, Jh, Mh, Ja, and Ma (possibly empty) correspond to the classes of operators mentioned above.

4.1 Propositional logic

Let $\mathcal{L}$ be a many-valued propositional logic with a set Var of propositional variables and logical operators[4] in $\{\vee, \wedge\} \cup \Sigma$, and satisfying the property:

(P) $\mathcal{L}$ has a finite set A of truth values; and $(A, \vee_A, \wedge_A, 0_A, 1_A, \{\sigma_A\}_{\sigma \in \Sigma}) \in \mathsf{DLO}_\Sigma$.

We now show that the Priestley representation theorem for A helps us to define a satisfiability relation with respect to the relational space $D(A)$, which can be regarded as a special kind of Kripke-style frame.

Note first that, since A is finite, there exists an isomorphism $\eta_A : A \to \mathcal{O}(D(A))$. For defining Kripke-style models based on $D(A)$, we impose the condition that all meaning functions are hereditary, i.e. have as values order-filters of $D(A)$ (for details about the reasons leading to this assumption see [SS97, SS00b]). Let $m : \mathsf{Var} \to \mathcal{O}(D(A))$ be a meaning function, and let $\overline{m} : \mathsf{Fma}(\mathsf{Var}) \to \mathcal{O}(D(A))$ be the unique extension of m to a morphism of $\{\vee, \wedge, 0, 1\} \cup \Sigma$-algebras (such an extension exists and is unique by the universality property of $\mathsf{Fma}(\mathsf{Var})$). It can be seen that $\overline{m} = \eta_A \circ \overline{\eta_A^{-1} \circ m}$, where $\overline{\eta_A^{-1} \circ m}$ is the unique morphism of $\{\vee, \wedge, 0, 1\} \cup \Sigma$-algebras extending $\eta_A^{-1} \circ m : \mathsf{Var} \to A$ to $\mathsf{Fma}(\mathsf{Var})$.

$$\begin{tikzcd}
\mathsf{Var} \arrow[r, "m"] \arrow[dr] & \mathcal{O}(D(A)) \arrow[r, shift left, "\eta_A^{-1}"] & A \arrow[l, shift left, "\eta_A"] \\
& \mathsf{Fma}(\mathsf{Var}) \arrow[u, "\overline{m}"] \arrow[ur, swap, "\overline{\eta_A^{-1} \circ m}"] &
\end{tikzcd}$$

[4]The operators in Σ may for instance correspond to versions of non-classical negation, or to operators modeling knowledge, belief, necessity, or possibility, that generalize similar operators from modal logic.

Definition 6 *Let $\phi \in \mathsf{Fma}(\mathsf{Var})$ and $\alpha \in D(A)$. Let $m : \mathsf{Var} \to \mathcal{O}(D(A))$ be a meaning function, and let $\overline{m} : \mathsf{Fma}(\mathsf{Var}) \to \mathcal{O}(D(A))$ be its extension to $\mathsf{Fma}(\mathsf{Var})$. We define:*

(1) $D(A) \overset{r}{\models}_{m,\alpha} \phi$ if and only if $\alpha \in \overline{m}(\phi)$,

(2) $D(A) \overset{r}{\models}_{m} \phi$ if and only if $\overline{m}(\phi) = D(A)$,

(3) $D(A) \overset{r}{\models} \phi$ if and only if $D(A) \overset{r}{\models}_{m} \phi$ for every $m : \mathsf{Var} \to \mathcal{O}(D(A))$.

Proposition 4 *Let $\mathcal{L}$ be a propositional logic satisfying condition (P). Then for every formula ϕ of $\mathcal{L}$, ϕ is valid in $\mathcal{L}$ if and only if $D(A) \overset{r}{\models} \phi$.*

Proof: By the definition of $\overset{r}{\models}$, $D(A) \overset{r}{\models} \phi$ if and only if $\mathcal{O}(D(A)) \overset{a}{\models} \phi = 1$. The conclusion now follows immediately using (P), Definition 3, and the fact that there exists an isomorphism $\eta_A : A \to \mathcal{O}(D(A))$. □

Example 3 *Let $\mathcal{L}$ be a propositional logic with set of propositional variables Var, logical connectives $\{\vee, \wedge, \sim\}$ and having as algebra of truth values the finite distributive lattice considered in Example 1 and represented in Figure 1. Let $\psi = \sim\phi \in \mathsf{Fma}(\mathsf{Var})$, and let $m : \mathsf{Var} \to \mathcal{O}(D(A))$ be a meaning function and $\alpha \in D(A)$. Then:*

$$\begin{array}{lll}
D(A) \overset{r}{\models}_{m,\alpha} \sim\phi & \textit{iff} & \alpha \in \overline{m}(\sim\phi) \\
& \textit{iff} & \alpha \in \sim_{\mathcal{O}(D(A))} (\overline{m}(\phi)) = D(A) \backslash D(\sim)^{-1}(\overline{m}(\phi)) \\
& \textit{iff} & \alpha \notin D(\sim)^{-1}(\overline{m}(\phi)) \\
& \textit{iff} & D(\sim)(\alpha) \notin \overline{m}(\phi) \\
& \textit{iff} & D(A) \not\models_{m,D(\sim)(\alpha)} \phi.
\end{array}$$

Example 4 *Let $\mathcal{L}$ be a propositional logic with set of propositional variables Var, logical connectives $\{\vee, \wedge, \neg\}$, and having as algebra of truth values the finite distributive lattice A considered in Example 2 and represented in Figure 2. Let $\psi = \neg\phi \in \mathsf{Fma}(\mathsf{Var})$, $m : \mathsf{Var} \to \mathcal{O}(D(A))$ a meaning function, and $\alpha \in D(A)$. Then:*

$$\begin{array}{lll}
D(A) \overset{r}{\models}_{m,\alpha} \neg\phi & \textit{iff} & \alpha \in \overline{m}(\neg\phi) \\
& \textit{iff} & \alpha \in \neg_{\mathcal{O}(D(A))}(\overline{m}(\phi)) = \{\gamma \in D(A) \mid \forall\beta \geq \gamma, \beta \notin \overline{m}(\phi)\} \\
& \textit{iff} & \text{for every } \beta \in D(A),\ \alpha \leq \beta \text{ implies } \beta \notin \overline{m}(\phi) \\
& \textit{iff} & \text{for every } \beta \in D(A),\ \alpha \leq \beta \text{ implies } D(A) \overset{r}{\not\models}_{m,\beta} \phi.
\end{array}$$

4.2 First-order logic

Let $\mathbf{L} = (X, O, P, \Sigma \cup \{\vee, \wedge\}, \{\forall, \exists\})$ be a first-order language, consisting of a (countably) infinite set X of variables, a set O of function symbols, and a set P of predicate symbols, where $\Sigma \cup \{\vee, \wedge\}$ is the set of logical operators and $\{\forall, \exists\}$ is the set of quantifiers. Let $(A, \vee_A, \wedge_A, 0_A, 1_A, \{\sigma_A\}_{\sigma \in \Sigma})$ be a finite algebra in DLO_Σ.

Let $\mathcal{L}$ be the first-order logic with the language $\mathbf{L}$, having A as a set of truth values, such that for every $\sigma \in \Sigma \cup \{\vee, \wedge\}$, $\sigma_A : A^{a(\sigma)} \to A$ is the truth table of σ in A. Moreover, assume that the truth tables of the operators $\forall$ and $\exists$ are given by generalized conjunctions, resp. disjunctions, i.e. $\overline{\forall}, \overline{\exists} : \mathcal{P}(A) \backslash \{\emptyset\} \to A$ are defined for every non-empty $W \subseteq A$ by

$$\overline{\forall}(W) = \bigwedge_{w \in W} w; \ \overline{\exists}(W) = \bigvee_{w \in W} w.$$

Frames and interpretations of $\mathbf{L}$ in A are defined as in Definition 2. Every interpretation $\mathcal{I} = (D, I, d)$ induces a valuation function $v_\mathcal{I} : \mathsf{Fma}(\mathbf{L}) \to A$ as explained in Section 2.2.

The fact that A and $\mathcal{O}(D(A))$ are isomorphic suggests that, also in this case, one can introduce an alternative notion of interpretation, namely as a triple $\mathcal{M} = (D, M, d)$, where $d : X \to D$, $M(f) : D^n \to D$ for every $f \in O$ of arity n, and $M(R) : D^n \to \mathcal{O}(D(A))$ for $R \in P$ of arity n. Because of the isomorphism between A and $\mathcal{O}(D(A))$ these two notions of interpretation are equivalent. Interpretations in $\mathcal{O}(D(A))$ extend to formulae in a similar way as interpretations in A; hence we can express notions as validity and satisfiability using the Priestley dual $D(A)$ of A, as in the case of propositional logics. Thus, the following result follows easily.

Proposition 5 *A formula ϕ is valid in $\mathcal{L}$ if and only if $v_{\mathcal{M}}(\phi) = D(A)$ for every interpretation $\mathcal{M} = (D, M, d)$ in $\mathcal{O}(D(A))$.*

Proof: By Definition 3, a formula ϕ is valid in $\mathcal{L}$ if and only if $v_{\mathcal{I}}(\phi) = 1$ for all interpretations $\mathcal{I}$ for the language of $\mathcal{L}$ with algebra of truth values A. Assume that ϕ is valid in $\mathcal{L}$, and let $\mathcal{M} = (D, M, d)$ be an interpretation in $\mathcal{O}(D(A))$. Let $\mathcal{I} = (D, I, d)$ be the interpretation in A obtained from $\mathcal{M}$ by defining, for every $R \in P$, $I(R) = \eta_A^{-1}(M(R))$. Then $v_{\mathcal{I}}(\phi) = 1$. From the definition of $\mathcal{I}$ it follows therefore that $v_{\mathcal{M}}(\phi) = \eta_A(v_{\mathcal{I}}(\phi)) = D(A)$. Conversely, assume that $v_{\mathcal{M}}(\phi) = D(A)$ for every interpretation $\mathcal{M} = (D, M, d)$ in $\mathcal{O}(D(A))$. Let $\mathcal{I} = (D, I, d)$ be an arbitrary interpretation in A, and let $\mathcal{M} = (D, M, d)$ be the interpretation in $\mathcal{O}(D(A))$ obtained from $\mathcal{I}$ by defining, for every $R \in P$, $M(R) = \eta_A(I(R))$. From the assumption, $v_{\mathcal{M}}(\phi) = D(A)$. From the definition of $\mathcal{M}$ it follows therefore that $v_{\mathcal{I}}(\phi) = \eta_A^{-1}(v_{\mathcal{M}}(\phi)) = 1$. □

5 Automated theorem proving

In this section we present a method for automated theorem proving based on the results presented in Section 3 and Section 4. The method was inspired by the work of Hähnle on automated theorem proving in regular logics [Häh94, Häh96b]. It extend those results to more general situations where the algebra of truth values is not linearly ordered. The main idea of the method we present here is to use signed literals, where the signs are elements of $D(A)$, corresponding to prime filters of truth values, instead of truth values (as done in [BF92, BF95]) or arbitrary sets of truth values (as done in [Häh94, Häh96b]). The idea of using "valuations in $\{0, 1\}$" instead of values appears already for instance in [Sco73] for Łukasiewicz logics.

Let $\mathcal{L}$ be a first-order logic with language $\mathbf{L} = (X, O, P, \Sigma \cup \{\vee, \wedge\}, \{\forall, \exists\})$ and algebra of truth values A (a finite algebra in DLO_Σ). In what follows, we consider signed formulae of the type:

$$\boxed{\alpha}\ \phi^t \text{ (“}\phi\text{ is true at }\alpha\text{”), and } \boxed{\alpha}\ \phi^f \text{ (“}\phi\text{ is false at }\alpha\text{”),}$$

where $\phi \in \mathsf{Fma}(\mathbf{L})$ and $\alpha \in D(A)$.

Definition 7 *Let $\mathcal{M} = (D, M, d)$ be an interpretation of $\mathbf{L}$ in $\mathcal{O}(D(A))$. Let $\phi \in \mathsf{Fma}(\mathbf{L})$ and $\alpha \in D(A)$. We say that $\boxed{\alpha}\ \phi^t$ holds in $\mathcal{M}$ (for short: $\boxed{\alpha}\ \phi^t$ in $\mathcal{M}$) if $\alpha \in v_{\mathcal{M}}(\phi)$. Similarly, $\boxed{\alpha}\ \phi^f$ in $\mathcal{M}$ if $\alpha \notin v_{\mathcal{M}}(\phi)$.*

Note that if $\alpha \leq \beta \in D(A)$ and $\boxed{\alpha}\ \phi^t$ in $\mathcal{M}$ then $\boxed{\beta}\ \phi^t$ in $\mathcal{M}$.

Lemma 6 *The formula ϕ is valid in $\mathcal{L}$ if and only if there is no interpretation $\mathcal{M} = (D, M, d)$ of $\mathbf{L}$ in $\mathcal{O}(D(A))$ such that there exists a minimal element $\alpha \in D(A)$ with $\boxed{\alpha}\ \phi^f$ in $\mathcal{M}$.*

Proof: It follows from the fact that ϕ is not valid if and only if there exists an interpretation $\mathcal{M} = (D, M, d)$ such that $v_{\mathcal{M}}(\phi) \neq D(A)$. From the fact that, for every interpretation $\mathcal{M}$ in $\mathcal{O}(D(A))$, $v_{\mathcal{M}}(\phi) \subseteq D(A)$ is an order-filter and finite, it follows that ϕ is not valid if and only if there exists an interpretation $\mathcal{M} = (D, M, d)$ and an element $\alpha \in D(A)$ which is *minimal* in $D(A)$ such that $\alpha \notin v_{\mathcal{M}}(\phi)$. □

In order to avoid possible confusion with the notation for the lattice operators, in what follows classical conjunction and disjunction between the signed formulae defined above will be denoted by $\bigwedge\!\!\!\bigwedge$ resp. $\bigvee\!\!\!\bigvee$. With this notation, the last condition in Lemma 6 can be expressed by $\bigvee\!\!\!\bigvee_{\alpha \in D(A),\ minimal} \boxed{\alpha}\ \phi^f$ in $\mathcal{M}$.

In order to define an automated theorem proving procedure, we need a method for translation into clause form, and a resolution procedure. Such methods will be presented in the subsequent subsections.

5.1 Translation to clause form

We present a *structure-preserving* transformation method inspired by the structure-preserving transformation method to clause form in first-order logic of Tseitin [Tse70]. For further details on such transformation methods and applications to classical and many-valued logics we refer to [Ede92], [BF95] and [Häh94].

5.1.1 Signed literals and signed clauses

Definition 8 (Literal, Clause, Signed CNF) *Let L be an atomic formula and let $\alpha \in D(A)$. Then $\boxed{\alpha}\ L^t$ is a* positive literal *with sign α, and $\boxed{\alpha}\ L^f$ is a* negative literal *with sign α. A set of (positive or negative) signed literals is called a* (signed) clause. *A clause is called* negative (positive) *if it only contains negative (positive) literals. A* formula in signed conjunctive normal form (CNF) *is a finite set of signed clauses. In the first-order case we require that the clauses in a signed CNF formula have disjoint variables.*

Definition 9 (Satisfiability) *An interpretation $\mathcal{M} = (D, M, d)$ of $\mathbf{L}$ in $\mathcal{O}(D(A))$* satisfies *a positive literal $\boxed{\alpha}\ L^t$ if L is true in $\mathcal{M}$ at α, i.e. $\alpha \in v_{\mathcal{M}}(L)$. $\mathcal{M}$* satisfies *a negative literal $\boxed{\alpha}\ L^f$ if L is false in $\mathcal{M}$ at α, i.e. $\alpha \notin v_{\mathcal{M}}(L)$. $\mathcal{M}$* satisfies *a signed clause if it satisfies at least one of the literals that occur in it.*

A signed clause is satisfiable if and only if there exists an interpretation $\mathcal{M} = (D, M, d)$ in $\mathcal{O}(D(A))$ that satisfies at least one of its literals. A formula Φ in signed CNF is satisfiable if and only if all clauses in Φ are simultaneously satisfiable by the same interpretation.

5.1.2 Structure-preserving translation to signed clause form

Let ϕ be a formula. The main idea of structure-preserving methods for transformation to clause form is to introduce for every non-atomic subformula ψ of ϕ a new atomic formula of the form $P_\psi(\overline{x})$, where P_ψ is a new predicate symbol and $\overline{x}$ are the free variables in ψ.

Definition 10 *Let $\mathcal{L}$ be a logic with language $\mathbf{L}$ and let $\phi \in \mathsf{Fma}(\mathbf{L})$. The language $\mathbf{L}^*$ is an extension of $\mathbf{L}$ obtained by introducing a new predicate symbol P_ψ with arity n for every non-atomic subformula ψ of ϕ with free variables $\{x_1, \ldots, x_n\}$.*

In what follows, let $\mathcal{L} = (\mathbf{L}, A)$ be a logic with language $\mathbf{L}$ and algebra of truth values A. We study the way interpretations of $\mathbf{L}$ in A can be extended when the language is extended as explained in Definition 10.

Definition 11 *Let $\mathcal{I} = (D, I, d)$ be an interpretation of* $\mathbf{L}$ *in A. Let ϕ be a formula of $\mathcal{L}$ and let* $\mathbf{L}^*$ *the extension of* $\mathbf{L}$ *as defined above. The extension $\mathcal{I}^* = (D^*, I^*, d^*)$ to the language* $\mathbf{L}^*$ *is defined inductively as follows:*

- *$D^* = D$; $d^* = d : X \to D$; $I^*(f) = I(f)$ for every function symbol $f \in O$; $I^*(R) = I(R)$ for every predicate symbol $R \in P$.*
- *For every non-atomic subformula $\psi = \sigma(\psi_1, \dots, \psi_m)$ of ϕ with free variables $\{x_1, \dots, x_n\}$, $I^*(P_\psi) : D^n \to A$ is inductively defined for all $d_1, \dots, d_n \in D$ by $I^*(P_\psi)(d_1, \dots, d_n) = \sigma_A(I^*(P_{\psi_1})(\overline{d}^1), \dots, I^*(P_{\psi_m})(\overline{d}^m))$*

  (where for every $i \in \{1, \dots, m\}$, $\overline{d}^i$ is the subsequence of $(d_1, \dots, d_n)$ corresponding to the free variables $\overline{x}^i$ in ψ_i).
- *For every non-atomic subformula $\psi = (Qx)\psi_1(x, x_1, \dots, x_n)$ of ϕ with free variables $\{x_1, \dots, x_n\}$, $I^*(P_\psi) : D^n \to A$ is defined for all $d_1, \dots, d_n \in D$ by $I^*(P_\psi)(d_1, \dots, d_n) = \overline{Q}(\{w \mid \exists d \in D, I^*(P_{\psi_1})(d, d_1, \dots, d_n) = w\})$.*

Lemma 7 *Let $\mathcal{I} = (D, I, d)$ be an interpretation of* $\mathbf{L}$ *in A. Let ϕ be a formula in $\mathcal{L}$,* $\mathbf{L}^*$ *the extension of* $\mathbf{L}$ *associated with ϕ as in Definition 10, and $\mathcal{I}^*$ the extension of $\mathcal{I}$ to* $\mathbf{L}^*$*. Then $v_{\mathcal{I}^*_{\{x_i/d_i, 1 \leq i \leq n\}}}(P_\psi(x_1, \dots, x_n)) = v_{\mathcal{I}_{\{x_i/d_i, 1 \leq i \leq n\}}}(\psi)$ for every subformula ψ of ϕ with free variables $x_1, \dots, x_n$, and for all $d_1, \dots, d_n \in D$, where $\mathcal{I}^*_{\{x_i/d_i, 1 \leq i \leq n\}}$ is identical to $\mathcal{I}^*$ except for assigning d_i to the variable x_i for $1 \leq i \leq n$.*

Proof: It easily follows by structural induction on the structure of ψ, taking into account the definition of $\mathcal{I}^*$. □

Lemma 8 *Let ϕ be a formula of $\mathcal{L}$, and let* $\mathbf{L}^*$ *be the extension of* $\mathbf{L}$ *associated with ϕ as in Definition 10. The formula ϕ is valid in $\mathcal{L}$ if and only if for every interpretation $\mathcal{I}^* = (D, I^*, d)$ of* $\mathbf{L}^*$ *in A that satisfies conditions (a) and (b) below, $P_\phi(x_1, \dots, x_n)$ is true in $\mathcal{I}^*$, where $x_1, \dots, x_n$ are the free variables of ϕ.*

(a) *For every subformula $\psi = \sigma(\psi_1, \dots, \psi_m)$ of ϕ with free variables $x_1, \dots, x_n$, $I^*(P_\psi)(d_1, \dots, d_n) = \sigma_A(I^*(P_{\psi_1})(\overline{d}^1), \dots, I^*(P_{\psi_m})(\overline{d}^m))$ for all $d_1, \dots, d_n \in D$, (where for every $i \in \{1, \dots, m\}$, $\overline{d}^i$ is the subsequence of $(d_1, \dots, d_n)$ corresponding to the free variables $\overline{x}^i$ in ψ_i).*

(b) *For every subformula $\psi = (Qx)\psi_1(x, x_1, \dots, x_n)$, with free variables $x_1, \dots, x_n$, $I^*(P_\psi)(d_1, \dots, d_n) = \overline{Q}(\{w \mid \exists d \in D, I^*(P_{\psi_1})(d, d_1, \dots, d_n) = w\})$, for all $d_1, \dots, d_n \in D$.*

Proof: Assume that ϕ is valid in $\mathcal{L}$. Let $\mathcal{I}^* = (D, I^*, d)$ be an arbitrary interpretation of $\mathbf{L}^*$ such that the conditions (a) and (b) hold. Then $\mathcal{I}^* = (D, I^*, d)$ is the extension of its restriction $\mathcal{I} = (D, I, d)$ to the language of $\mathcal{L}$, hence, by Lemma 7, $v_{\mathcal{I}^*}(P_\phi(x_1, \dots, x_n)) = v_{\mathcal{I}}(\phi)$. Since ϕ is valid, we know that $v_{\mathcal{I}}(\phi) = 1$, hence $v_{\mathcal{I}^*}(P_\phi(x_1, \dots, x_n)) = 1$, i.e. $P_\phi(x_1, \dots, x_n)$ is true in $\mathcal{I}^*$. Conversely, assume that for every interpretation $\mathcal{I}^*$ of $\mathbf{L}^*$ that satisfies conditions (a) and (b), $P_\phi(x_1, \dots, x_n)$ is true in $\mathcal{I}^*$. Let $\mathcal{I} = (D, I, d)$ be an interpretation for the language of $\mathcal{L}$. It is easy to see that the extension $\mathcal{I}^*$ of $\mathcal{I}$ to the language obtained by introducing a new predicate symbol for every subformula ψ of ϕ satisfies (a) and (b). It follows then that $P_\phi(x_1, \dots, x_n)$ is true in $\mathcal{I}^*$. But $v_{\mathcal{I}^*}(P_\phi(x_1, \dots, x_n)) = v_{\mathcal{I}}(\phi)$. Thus, ϕ is true in $\mathcal{I}$. □

Proposition 9 *The formula ϕ is valid in $\mathcal{L}$ if and only if there exists no interpretation $\mathcal{M} = (D, M, d)$ of the extension* $\mathbf{L}^*$ *of* $\mathbf{L}$ *in $\mathcal{O}(D(A))$ such that:*

(N) $\bigvee_{\alpha \in D(A),\ minimal} \boxed{\alpha}\ P_\phi(x_1, \ldots, x_n)^f$ *in* $\mathcal{M}$*;*

$$(\sigma) \begin{cases} (\boxed{\alpha}\ P_\psi(x_1, \ldots, x_n)^t \vee \boxed{\alpha}\ \sigma(P_{\psi_1}(\overline{x}^1), \ldots, P_{\psi_m}(\overline{x}^m))^f) \\ \qquad\qquad\qquad \wedge \\ (\boxed{\alpha}\ P_\psi(x_1, \ldots, x_n)^f \vee \boxed{\alpha}\ \sigma(P_{\psi_1}(\overline{x}^1), \ldots, P_{\psi_m}(\overline{x}^m))^t) \end{cases}$$

in $\mathcal{M}$ *for every* $\alpha \in D(A)$

for all subformulae $\psi = \sigma(\psi_1, \ldots, \psi_m)$ *of* ϕ*, and every instantiation in* D *of the free variables* $x_1, \ldots, x_n$ *of* ψ*;*

$$(Q) \begin{cases} (\boxed{\alpha}\ P_\psi(x_1, \ldots, x_n)^t \vee \boxed{\alpha}\ (Qx)P_{\psi_1}(x, x_1, \ldots, x_n)^f) \\ \qquad\qquad\qquad \wedge \\ (\boxed{\alpha}\ P_\psi(x_1, \ldots, x_n)^f \vee \boxed{\alpha}\ (Qx)P_{\psi_1}(x, x_1, \ldots, x_n)^t) \end{cases}$$

in $\mathcal{M}$ *for every* $\alpha \in D(A)$

for all subformulae of ϕ *of the form* $\psi = (Qx)\psi_1(x, x_1, \ldots, x_n)$ *and every instantiation in* D *of the free variables* $x_1, \ldots, x_n$ *of* ψ*.*

Proof: Taking into account the bijective correspondence between interpretations $\mathcal{I} = (D, I, d)$ based on A and interpretations $\mathcal{M} = (D, M, d)$ based on $\mathcal{O}(D(A))$ it easily follows from Lemma 6 and Lemma 8 that the formula ϕ is valid if and only if there exists no interpretation $\mathcal{M} = (D, M, d)$ based on $\mathcal{O}(D(A))$ of the extension $\mathbf{L}^*$ of the language of $\mathcal{L}$ such that $P_\phi(x_1, \ldots, x_n)$ is false in $\mathcal{M}$ at some $\alpha \in D(A)$, provided the renaming is made correctly, i.e.

- for all subformulae $\psi = \sigma(\psi_1, \ldots, \psi_m)$ and every instantiation in D of the free variables $x_1, \ldots, x_n$ of ψ, $P_\psi(x_1, \ldots, x_n)$ is true at α in $\mathcal{M}$ if and only if $\sigma(P_{\psi_1}(\overline{x}^1), \ldots, P_{\psi_m}(\overline{x}^m))$ is true at α in $\mathcal{M}$ (where for every $i \in \{1, \ldots, m\}$, $\overline{x}^1$ is the subsequence of $(x_1, \ldots, x_n)$ consisting of those free variables occurring in ψ_i); and
- for all subformulae $\psi = (Qx)\psi_1(x, x_1, \ldots, x_n)$ and every instantiation in D of the free variables $x_1, \ldots, x_n$ of ψ, $P_\psi(x_1, \ldots, x_n)$ is true at α in $\mathcal{M}$ if and only if $(Qx)P_{\psi_1}(x, x_1, \ldots, x_n)$ is true at α in $\mathcal{M}$.

The result now easily follows from the fact that for every two formulae ϕ_1, ϕ_2, every $\alpha \in D(A)$, and every interpretation $\mathcal{M}$, the sentence "*ϕ_1 is true at α in $\mathcal{M}$ if and only if ϕ_2 is true at α in $\mathcal{M}$*" can alternatively be expressed by

$$(\boxed{\alpha}\ \phi_1^t \vee \boxed{\alpha}\ \phi_2^f) \wedge (\boxed{\alpha}\ \phi_1^f \vee \boxed{\alpha}\ \phi_2^t) \text{ in } \mathcal{M}.$$

$\square$

Remark 10 The formulae (σ) or (Q) used in Proposition 9 for expressing the fact that the renaming of subformulae is correct correspond to conjunctions of implications of the form $(\boxed{\alpha}\ \phi_1^t \to \boxed{\alpha}\ \phi_2^t) \wedge (\boxed{\alpha}\ \phi_2^t \to \boxed{\alpha}\ \phi_1^t)$. In classical logic, considerations about the polarity of the subformulae may show that only one of the implications is actually necessary. This is also the case when using a relational translation for automated theorem proving in modal logics; similar considerations based on a notion of polarity are also made in the context of regular logics by Hähnle in [Häh94]. The nature of the operators we allow on the algebra of truth values, in particular the existence of the antimorphisms, makes it impossible to use a priori a notion of polarity for keeping only one of the implications in the formulae expressing correctness of renaming, as done in modal logic. This may however be possible in particular cases. Issues related to polarity are further discussed in Section 5.2.

In order to obtain a signed conjunctive normal form starting from the conjunction of the formulae (N), (σ) and (Q) in Proposition 9, we have to analyze the way in which expressions of the form $\boxed{\alpha}\psi^t$ and $\boxed{\alpha}\psi^f$ can be decomposed further, where $\psi = \sigma(P_{\psi_1}(\overline{x}^1), \dots, P_{\psi_m}(\overline{x}^m))$ or $\psi = (Qx)P_{\psi_1}(x, x_1, \dots, x_n)$. This last step in the translation to clause form depends on the properties of the operations. In what follows, by interpretation in A, resp. $\mathcal{O}(D(A))$ we mean interpretation of the extension $\mathbf{L}^*$ of the language of $\mathcal{L}$ in A, resp. $\mathcal{O}(D(A))$.

Lemma 11 *The following holds, for any atomic formulae $L, L_1, L_2, \dots, L_n$, in any interpretation $\mathcal{M}$ in $\mathcal{O}(D(A))$, and for every $\alpha \in D(A)$:*

(∨) $\boxed{\alpha}\,(L_1 \vee L_2)^t$ *if and only if* $(\boxed{\alpha}\,L_1^t) \Cup (\boxed{\alpha}\,L_2^t)$;
(∧) $\boxed{\alpha}\,(L_1 \wedge L_2)^t$ *if and only if* $(\boxed{\alpha}\,L_1^t) \Cap (\boxed{\alpha}\,L_2^t)$.

Assume that the logic additionally contains operators like lattice morphisms and antimorphisms, join and meet hemimorphisms and antimorphisms. For such an operator f, $D(f)$ denotes the corresponding function or relation in $D(A)$, as defined in Section 3.1.1.

(Lh) Let $h \in Lh$ be an operator such that $h_A : A \to A$ is a lattice morphism. Then $\boxed{\alpha}\,h(L)^t$ if and only if $\boxed{D(h_A)(\alpha)}\,L^t$.

(La) Let $k \in La$ be an operator such that $k_A : A \to A$ is a lattice antimorphism. Then $\boxed{\alpha}\,k(L)^t$ if and only if $\boxed{D(k_A)(\alpha)}\,L^f$.

(Jh) Let $f \in Jh$ be an operator such that $f_A : A^n \to A$ is a join hemimorphism. Then $\boxed{\alpha}\,f(L_1, \dots, L_n)^t$ if and only if $\bigvee\!\!\!\!\bigvee\{\boxed{\alpha_1}\,L_1^t \Cap \dots \Cap \boxed{\alpha_n}\,L_n^t \mid (\alpha_1, \dots, \alpha_n, \alpha) \in D(f_A)\}$.

(Mh) Let $g \in Mh$ be an operator such that $g_A : A^n \to A$ is a meet hemimorphism. Then $\boxed{\alpha}\,g(L_1, \dots, L_n)^t$ if and only if $\bigwedge\!\!\!\!\bigwedge\{\boxed{\alpha_1}\,L_1^t \Cup \dots \Cup \boxed{\alpha_n}\,L_n^t \mid (\alpha_1, \dots, \alpha_n, \alpha) \in D(g_A)\}$.

(Ja) Let $f' \in Ja$ be an operator such that $f'_A : A^n \to A$ is a join hemiantimorphism. Then $\boxed{\alpha}\,f'(L_1, \dots, L_n)^t$ if and only if $\bigvee\!\!\!\!\bigvee\{\boxed{\alpha_1}\,L_1^f \Cap \dots \Cap \boxed{\alpha_n}\,L_n^f \mid (\alpha_1, \dots, \alpha_n, \alpha) \in D(f'_A)\}$.

(Ma) Let $g' \in Ma$ be an operator such that $g'_A : A^n \to A$ is a meet hemiantimorphism. Then $\boxed{\alpha}\,g'(L_1, \dots, L_n)^t$ if and only if $\bigwedge\!\!\!\!\bigwedge\{\boxed{\alpha_1}\,L_1^f \Cup \dots \Cup \boxed{\alpha_n}\,L_n^f \mid (\alpha_1, \dots, \alpha_n, \alpha) \in D(g'_A)\}$.

Proof: The cases $(\vee, t), (\vee, f), (\wedge, t), (\wedge, f)$ follow easily.

(Lh) $v_{\mathcal{M}}(h(L)) = h_{\mathcal{O}(D(A))}(v_{\mathcal{M}}(L)) = \{\alpha \in D(A) \mid D(h_A)(\alpha) \in v_{\mathcal{M}}(L)\}$.
Hence, $\alpha \in v_{\mathcal{M}}(h(L))$ if and only if $D(h_A)(\alpha) \in v_{\mathcal{M}}(L)$.

(La) follows similarly from the definition of $k_{\mathcal{O}(D(A))}$, for $k \in La$, and the fact that $v_{\mathcal{M}}(k(L)) = k_{\mathcal{O}(D(A))}(v_{\mathcal{M}}(L)) = \{\alpha \in D(A) \mid D(h_A)(\alpha) \notin v_{\mathcal{M}}(L)\}$.

(Jh)
$$\begin{aligned} v_{\mathcal{M}}(f(L_1, \dots, L_n)) &= f_{\mathcal{O}(D(A))}(v_{\mathcal{M}}(L_1), \dots, v_{\mathcal{M}}(L_n)) = \\ &= \{x \in D(A) \mid \exists x_1, \dots x_n, \text{ with } x_1 \in v_{\mathcal{M}}(L_1), \dots, \\ &\qquad x_n \in v_{\mathcal{M}}(L_n) \text{ and } (x_1, \dots, x_n, x) \in D(f_A)\}. \end{aligned}$$
The conclusion follows immediately.

(Mh) follows similarly from the definition of $g_{\mathcal{O}(D(A))}$ and the fact that $v_{\mathcal{M}}(g(L_1, \dots, L_n)) = g_{\mathcal{O}(D(A))}(v_{\mathcal{M}}(L_1), \dots, v_{\mathcal{M}}(L_n))$.

The results for the cases (Ja) and (Ma) can be proved similarly. □

For the situation when $\psi = (Qx)\psi_1(x)$, where $Q \in \{\forall, \exists\}$, we have the following results (for related results we refer to [Häh96a, Häh98]).

Lemma 12 *Let $\mathcal{I} = (D, I, d)$ be an interpretation in A, let $L(x, x_1, \ldots, x_n)$ be an atomic formula with free variables $x, x_1, \ldots, x_n$, and let $a \in A$ be a join-irreducible element. The following properties hold:*

($\forall$1) *$v_{\mathcal{I}}(\forall x L(x, x_1, \ldots, x_n)) \geq a$ if and only if $v_{\mathcal{I}_{x/d_x}}(L(x, x_1, \ldots, x_n)) \geq a$ for every instantiation $d_x \in D$ of x.*

($\forall$2) *$v_{\mathcal{I}}(\forall x L(x, x_1, \ldots, x_n)) \not\geq a$ if and only if $v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) \not\geq a$ for some instantiation $d_0 \in D$ of x, depending on $d(x_1), \ldots, d(x_n)$.*

($\exists$1) *$v_{\mathcal{I}}(\exists x L(x, x_1, \ldots, x_n)) \geq a$ if and only if $v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) \geq a$ for some instantiation $d_0 \in D$ of x, depending on $d(x_1), \ldots, d(x_n)$.*

($\exists$2) *$v_{\mathcal{I}}(\exists x L(x, x_1, \ldots, x_n)) \not\geq a$ if and only if $v_{\mathcal{I}_{x/d_x}}(L(x, x_1, \ldots, x_n)) \not\geq a$ for every instantiation $d_x \in D$ of x.*

Proof: ($\forall$1) and ($\forall$2) follow immediately, taking into account that

$$v_{\mathcal{I}}(\forall x L(x, x_1, \ldots, x_n)) = \bigwedge\{w \mid \exists d_0 \in D \ s.t.\, v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w\}.$$

($\exists$1) For the direct implication, assume that $v_{\mathcal{I}}(\exists x L(x, x_1, \ldots, x_n)) = \bigvee\{w \mid \exists d_0 \in D \ s.t.\ v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w\} \geq a$. From the fact that A is distributive and for every L the set $\{w \mid \exists d_0 \in D \ s.t.\, v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w\}$ is finite, it follows that $a = \bigvee\{w \mid \exists d_0 \in D \ s.t.\, v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w\} \wedge a = \bigvee\{w \wedge a \mid \exists d_0 \in D \ s.t.\, v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w\}$. By the fact that a is join-irreducible it then follows that $a = w \wedge a$ for some w such that there exists $d_0 \in D$ with $v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w$, hence, $v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) \geq a$ for some $d_0 \in D$ (depending on $d(x_1), \ldots, d(x_n)$).

The converse follows immediately since if $v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) \geq a$ for some $d_0 \in D$, then $v_{\mathcal{I}}(\exists x L(x)) = \bigvee\{w \mid \exists d_0 \in D \ s.t.\, v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) = w\} \geq a$.

($\exists$2) follows immediately from ($\exists$1). □

Lemma 13 *Let $\mathcal{M} = (D, M, d)$ be an interpretation in $\mathcal{O}(D(A))$, and let $\mathcal{I} = (D, I, d)$ be the interpretation in A associated to $\mathcal{M}$ (obtained by composition with η_A^{-1} when necessary). Let $\alpha = \uparrow a \in D(A)$, with $a \in A$ join-irreducible. Then for every $\psi \in \mathsf{Fma}(\mathbf{L})$, $\boxed{\alpha}\ \psi^t$ in $\mathcal{M}$ if and only if $v_{\mathcal{I}}(\psi) \geq a$.*

Proof: Let $v_{\mathcal{M}} : \mathsf{Fma}(\mathbf{L}) \to \mathcal{O}(D(A))$ and $v_{\mathcal{I}} : \mathsf{Fma}(\mathbf{L}) \to A$ the valuation functions induced by $\mathcal{M}$ and $\mathcal{I}$ respectively. For every $\psi \in \mathsf{Fma}(\mathbf{L})$, $v_{\mathcal{M}}(\psi) = \eta_A(v_{\mathcal{I}}(\psi)) = \{\uparrow a \mid v_{\mathcal{I}}(\psi) \in \uparrow a\} = \{\uparrow a \mid v_{\mathcal{I}}(\psi) \geq a\}$. Thus, $\alpha = \uparrow a \in v_{\mathcal{M}}(\psi)$ if and only if $v_{\mathcal{I}}(\psi) \geq a$, i.e. $\boxed{\alpha}\ \psi^t$ in $\mathcal{M}$ if and only if $v_{\mathcal{I}}(\psi) \geq a$. □

Proposition 14 *Let $\mathcal{M} = (D, M, d)$ be an interpretation in $\mathcal{O}(D(A))$ and let $L(x, x_1, \ldots, x_n)$ an atomic formula with free variables $x, x_1, \ldots, x_n$. Let $\alpha \in D(A)$ be arbitrary but fixed. The following properties hold:*

(1) $\boxed{\alpha}\ (\forall x L(x, x_1, \ldots, x_n))^t$ in $\mathcal{M}$ iff $\boxed{\alpha}\ (L(x, x_1, \ldots, x_n))^t$ in $\mathcal{M}$ for every instantiation of x.

(2) $\boxed{\alpha}\ (\forall x L(x, x_1, \ldots, x_n))^f$ in $\mathcal{M}$ iff $\boxed{\alpha}\ (L(f(x_1, \ldots, x_n), x_1, \ldots, x_n))^f$ in $\mathcal{M}$ (where f is a new function symbol).

(3) $\boxed{\alpha}\ (\exists x L(x, x_1, \ldots, x_n))^t$ in $\mathcal{M}$ iff $\boxed{\alpha}\ (L(f(x_1, \ldots, x_n), x_1, \ldots, x_n))^t$ in $\mathcal{M}$ (where f is a new function symbol).

(4) $\boxed{\alpha}\ (\exists x L(x, x_1, \ldots, x_n))^f$ in $\mathcal{M}$ iff $\boxed{\alpha}\ (L(x, x_1, \ldots, x_n))^f$ in $\mathcal{M}$ for every instantiation of x.

Proof: Let $\alpha \in D(A)$ be arbitrary but fixed. We know that $\alpha = \uparrow a$, where $a \in A$ is join-irreducible. Let $\mathcal{I} = (D, I, d)$ be the interpretation in A associated to $\mathcal{M}$. By Lemma 13, for every formula $\psi \in \mathsf{Fma}(\mathbf{L})$, $\boxed{\alpha}\ \psi^t$ in $\mathcal{M}$ if and only if $v_{\mathcal{I}}(\psi) \geq a$.

(1) $\boxed{\alpha}\ (\forall x L(x, x_1, \ldots, x_n))^t$ in $\mathcal{M}$ if and only if $v_{\mathcal{I}}(\forall x L(x, x_1, \ldots, x_n)) \geq a$, which, by Lemma 12 happens if and only if $v_{\mathcal{I}_{x/d_x}}(L(x, x_1, \ldots, x_n)) \geq a$ for every instantiation $d_x \in D$ of x. Thus, $\boxed{\alpha}\ (\forall x L(x, x_1, \ldots, x_n))^t$ in $\mathcal{M}$ if and only if for every instantiation of x, $\boxed{\alpha}\ (L(x, x_1, \ldots, x_n))^t$ in $\mathcal{M}$.

(2) $\boxed{\alpha}\ (\forall x L(x, x_1, \ldots, x_n))^f$ in $\mathcal{M}$ if and only if $v_{\mathcal{I}}(\forall x L(x, x_1, \ldots, x_n)) \not\geq a$. By Lemma 12 this happens if and only if $v_{\mathcal{I}_{x/d_0}}(L(x, x_1, \ldots, x_n)) \not\geq a$ for some instantiation $d_0 \in D$ of x, depending on the values of the other free variables in L. (Then $f(x_1, \ldots, x_n)$ must be interpreted as d_0.)

The assertions (3) and (4) follow analogously. $\square$

We can use these remarks to obtain a signed conjunctive normal form for ϕ. The rules for the operators $\vee$, $\wedge$, the lattice morphisms and the lattice antimorphisms can be schematically represented as follows (where SL is a disjunction – possibly empty – of signed literals, and L, L_1, L_2 are atomic formulae):

$$(\vee) : (\vee, t) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ (L_1 \vee L_2)^t}{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_1^t \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_2^t} \qquad (\vee, f) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ (L_1 \vee L_2)^f}{(SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_1^f) \mathrel{\wedge\!\!\wedge} (SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_2^f)}$$

$$(\wedge) : (\wedge, t) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ (L_1 \wedge L_2)^t}{(SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_1^t) \mathrel{\wedge\!\!\wedge} (SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_2^t)} \qquad (\wedge, f) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ (L_1 \wedge L_2)^f}{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_1^f \mathrel{\vee\!\!\vee} \boxed{\alpha}\ L_2^f}$$

$$(Lh) : (h,\ t) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ h(L)^t}{SL \mathrel{\vee\!\!\vee} \boxed{D(h_A)(\alpha)}\ L^t} \qquad (h,\ f) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ h(L)^f}{SL \mathrel{\vee\!\!\vee} \boxed{D(h_A)(\alpha)}\ L^f}$$

$$(La) : (k,\ t) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ k(L)^t}{SL \mathrel{\vee\!\!\vee} \boxed{D(k_A)(\alpha)}\ L^f} \qquad (k,\ f) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ k(L)^f}{SL \mathrel{\vee\!\!\vee} \boxed{D(k_A)(\alpha)}\ L^t}$$

The rules for join and meet hemimorphisms and hemiantimorphisms are more complicated. We illustrate them by two simple examples, namely a unary join hemimorphism $\diamond : A \to A$, and a unary meet hemimorphism $\square : A \to A$. The rules (Jh) and (Mh) then specialize to:

$$\diamond : (\diamond, t) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ \diamond(L)^t}{SL \mathrel{\vee\!\!\vee} \bigvee\!\!\!\bigvee_{(\alpha_1, \alpha) \in D(\diamond)}^{\alpha_1} \boxed{\alpha_1}\ L^t} \qquad (\diamond, f) : \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ \diamond(L)^f}{\bigwedge\!\!\!\bigwedge_{(\alpha_1, \alpha) \in D(\diamond)}^{\alpha_1} \left(SL \mathrel{\vee\!\!\vee} \boxed{\alpha_1}\ L^f\right)}$$

$$\square : (\square, t) \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ \square(L)^t}{\bigwedge\!\!\!\bigwedge_{(\alpha_1, \alpha) \in D(\square)}^{\alpha_1} \left(SL \mathrel{\vee\!\!\vee} \boxed{\alpha_1}\ L^t\right)} \qquad (\square, f) : \quad \frac{SL \mathrel{\vee\!\!\vee} \boxed{\alpha}\ \square(L)^f}{SL \mathrel{\vee\!\!\vee} \bigvee\!\!\!\bigvee_{(\alpha_1, \alpha) \in D(\square)}^{\alpha_1} \boxed{\alpha_1}\ L^f}$$

The rules for the quantifiers $\forall$ and $\exists$ are given below (f_i are Skolem functions):

$$\forall : (\forall, t) \quad \frac{SL \Cup \boxed{\alpha}\ (\forall x)L(x, \overline{y})^t}{SL \Cup \boxed{\alpha}\ L(x, \overline{y})^t} \qquad (\forall, f) : \quad \frac{SL \Cup \boxed{\alpha}\ (\forall x)L(x, \overline{y})^f}{SL \Cup \boxed{\alpha}\ L(f_i(\overline{y}), \overline{y})^f}$$

$$\exists : (\exists, t) \quad \frac{SL \Cup \boxed{\alpha}\ (\exists x)L(x, \overline{y})^t}{SL \Cup \boxed{\alpha}\ L(f_i(\overline{y}), \overline{y})^t} \qquad (\exists, f) : \quad \frac{SL \Cup \boxed{\alpha}\ (\exists x)L(x, \overline{y})^f}{SL \Cup \boxed{\alpha}\ L(x, \overline{y})^f}$$

The next proposition gives an estimate of the number of clauses generated from a given formula ϕ (we assume that only unary meet and join hemimorphisms and hemiantimorphisms occur in ϕ).

Proposition 15 *Let n be the number of elements of $D(A)$. Let ϕ be a formula, let $SF(\phi)$ be the set of non-atomic subformulae of ϕ, and let $l = card(SF(\phi))$.*

(1) If ϕ only contains the connectives $\wedge, \vee$ and $\{g_i \mid i \in I\}$ such that for every $i \in I$, g_i is interpreted as a lattice morphism or antimorphism on A, then the number of clauses generated from ϕ is in $O(l \cdot n)$.

(2) If ϕ only contains the connectives $\wedge, \vee$ and $\{g_i \mid i \in I\}$ such that for every $i \in I$, g_i is interpreted as a lattice morphism or antimorphism, or as a unary join or meet hemimorphism or hemiantimorphism on A, then the number of clauses generated from ϕ is in $O(l \cdot n^2)$.

Proof: By Proposition 9, the number of clauses generated from a given formula ϕ is $1 + N_\sigma + N_Q$, where N_σ (resp. N_Q) is the number of clauses in the clause form of the conjunction of formulae in (σ) (resp. (Q)). Note first that since all the rules for quantifiers have branching factor 1, N_Q is in $O(n \cdot l)$.

(1) Assume that ϕ only contains the connectives $\vee, \wedge$ and lattice morphisms and/or antimorphisms. Then, for every $\alpha \in D(A)$ and every non-atomic subformula $\psi = \sigma(\psi_1, \dots, \psi_m)$ of ϕ, the formulae $(\boxed{\alpha}\ P_\psi(\overline{x})^f \Cup \boxed{\alpha}\ \sigma(P_{\psi_1}(\overline{x}^1), \dots, P_{\psi_m}(\overline{x}^m))^t)$ and $(\boxed{\alpha}\ P_\psi(\overline{x})^t \Cup \boxed{\alpha}\ \sigma(P_{\psi_1}(\overline{x}^1), \dots, P_{\psi_m}(\overline{x}^m))^f)$ generate together at most 3 clauses. Thus, the number of clauses generated from ϕ has as upper bound $1 + 3nl$, so it is in $O(l \cdot n)$.

(2) If the formula ϕ additionally contains unary join and meet hemimorphisms (resp. hemiantimorphisms) then every subformula ψ of ϕ starting with such an operator generates, for every $\alpha \in D(A)$, at most $n + 1$ clauses. Thus, the number of clauses generated from ϕ has in this case as upper bound $1 + (n + 1)nl$, so it is in $O(l \cdot n^2)$. $\square$

Thus, for logics that only have operators whose interpretations in A are conjunction, disjunction, lattice morphisms and antimorphisms, and unary join and meet hemi(anti)morphisms the method described above generates few clauses: the number is at most quadratic in the number of the dual of the algebra of truth values. From Lemma 11 it is however easy to see that, by distributivity, the rules for join and meet hemi(anti)morphisms of arity greater than or equal to 2 may introduce a number of clauses that is exponential in the number of truth values. Often, this number can be reduced according to special properties of the particular examples which are considered. Another possibility[5] of reducing the number of clauses in our approach would require quantification over the elements of $D(A)$, according to the definition of the operators in $D(A)$.

[5]We thank Hans-Jürgen Ohlbach for this remark.

Example 5 *Let $\mathcal{L}$ be a first-order logic with set of logical connectives $\{\vee, \wedge, \sim\}$, where $\vee$ and $\wedge$ are binary and $\sim$ is unary, and quantifiers $\forall, \exists$. Assume that $\mathcal{L}$ has as algebra of truth values the 9-element algebra A described in Example 1. Let $\phi = \forall x(p_1(x) \vee \sim p_2(x))$ be a formula in the logic $\mathcal{L}$. We will now show how a set Φ of signed clauses can be obtained, such that ϕ is valid in $\mathcal{L}$ if and only if Φ is unsatisfiable.*

Step 1: *For every non-atomic subformula of ϕ we introduce a new predicate symbol. This yields the following new atomic formulae:*

$$P_{\forall x(p_1(x) \vee \sim p_2(x))},\ P_{p_1(x) \vee \sim p_2(x)}(x),\ P_{\sim p_2(x)}(x).$$

Step 2: *By Lemma 6 and Proposition 9, ϕ is valid if and only if the following conjunction of formulae is unsatisfiable:*

$$\begin{cases} \boxed{\uparrow c} P^f_{\forall x(p_1(x)\vee\sim p_2(x))} & \Cup & \boxed{\uparrow e} P^f_{\forall x(p_1(x)\vee\sim p_2(x))}, & \\ \boxed{\alpha} P^t_{\forall x(p_1(x)\vee\sim p_2(x))} & \textit{iff} & \boxed{\alpha} (\forall x P_{p_1(x)\vee\sim p_2(x)}(x))^t & \textit{for every } \alpha \in D(A), \\ \boxed{\alpha} P_{p_1(x)\vee\sim p_2(x)}(x)^t & \textit{iff} & \boxed{\alpha} (p_1(x) \vee P_{\sim p_2(x)}(x))^t & \textit{for every } \alpha \in D(A), \\ \boxed{\alpha} P_{\sim p_2(x)}(x)^t & \textit{iff} & \boxed{\alpha} (\sim p_2(x))^t & \textit{for every } \alpha \in D(A). \end{cases}$$

The conjunction above is not yet in clause form. It can be brought into clause form by applying the rules presented in Lemma 11 and Lemma 12. By Proposition 9, Lemma 11 and Lemma 12, ϕ is valid if and only if the set of clauses

$$\Phi = \{C_1\} \cup \bigcup_{\alpha \in D(A)} \{C_{21\alpha}, C_{22\alpha}, C_{311\alpha}, C_{312\alpha}, C_{32\alpha}, C_{41\alpha}, C_{42\alpha}\}$$

is unsatisfiable, where:

$C_1 = \{\boxed{\uparrow c} P^f_{\forall x(p_1(x), \sim p_2(x))}, \boxed{\uparrow e} P^f_{\forall x(p_1(x)\vee\sim p_2(x))}\}.$ *($\uparrow c$ and $\uparrow e$ are minimal in $D(A)$, cf. Figure 1 in Example 1.)*

$C_{21\alpha} = \{\boxed{\alpha} P^f_{\forall x(p_1(x)\vee\sim p_2(x))}, \boxed{\alpha} P_{p_1(x)\vee\sim p_2(x)}(x)^t\},$ *for every $\alpha \in D(A)$, where x is a free variable.*

$C_{22\alpha} = \{\boxed{\alpha} P^t_{\forall x(p_1(x)\vee\sim p_2(x))}, \boxed{\alpha} P_{p_1(x)\vee\sim p_2(x)}(sk)^f\},$ *for every $\alpha \in D(A)$, where sk is a new Skolem constant.*

$C_{311\alpha} = \{\boxed{\alpha} P_{p_1(x)\vee\sim p_2(x)}(x)^t, \boxed{\alpha}\ p_1(x)^f\},$ *for every $\alpha \in D(A)$, where x is a free variable.*

$C_{312\alpha} = \{\boxed{\alpha} P_{p_1(x)\vee\sim p_2(x)}(x)^t, \boxed{\alpha}\ P_{\sim p_2(x)}(x)^f\},$ *for every $\alpha \in D(A)$, where x is a free variable.*

$C_{32\alpha} = \{\boxed{\alpha} P_{p_1(x)\vee\sim p_2(x)}(x)^f, \boxed{\alpha}\ p_1(x)^t, \boxed{\alpha}\ P_{\sim p_2(x)}(x)^t\},$ *for every $\alpha \in D(A)$, where x is a free variable.*

$C_{41\alpha} = \{\boxed{\alpha} P_{\sim p_2(x)}(x)^t, \boxed{D(\sim)(\alpha)}\ p_2(x)^t\},$ *for every $\alpha \in D(A)$, where x is a free variable.*

$C_{42\alpha} = \{\boxed{\alpha} P_{\sim p_2(x)}(x)^f, \boxed{D(\sim)(\alpha)}\ p_2(x)^f\},$ *for every $\alpha \in D(A)$, where x is a free variable.*

Note that in the clause forms obtained by this method the literals can be considered to be labeled by a pair (α, v) consisting of an element $\alpha = \uparrow i \in D(A)$, and a classical truth value $v \in \{t, f\}$. Thus, all labels can be considered as being of the form $(\uparrow i, t)$ or $(\uparrow i, f)$, where i is a join-irreducible element of A.

5.2 Theorem Proving

Checking the satisfiability of the sets of signed clauses obtained after the clause form transformation described in Section 5.1 does not present a problem.

In [SS98a], where we focus on theorem proving for sets of signed clauses, we note that the translation to clause form described above is in fact a translation to (many-sorted) classical logic ($\boxed{\alpha}\ L^t$ stands for $\mathsf{holds}(L, \alpha)$, and $\boxed{\alpha}\ L^f$ for $\neg\mathsf{holds}(L, \alpha)$). Then, as shown in [SS98a], the validity of ϕ can be checked by applying classical resolution to the set Φ^c of classical clauses obtained from the signed CNF of ϕ by replacing $\boxed{\alpha}\ L^t$ with $\mathsf{holds}(L, \alpha)$, and $\boxed{\alpha}\ L^f$ with $\neg\mathsf{holds}(L, \alpha)$, to which the set of clauses $\mathsf{Her} = \{\{\neg\mathsf{holds}(x, \alpha), \mathsf{holds}(x, \beta)\} \mid \alpha \leq \beta\}$, expressing the heredity of truth, is adjoined. Therefore, considerations from *classical logic* about the polarity of formulae can, in certain circumstances, be used to reduce the number of clauses generated by the translation to clause form, by only considering one of the implications in the formulae expressing that renaming is done correctly (cf. also Proposition 9 and Remark 10). Moreover the decidability of certain fragments of first-order finitely-valued logic follows immediately from classical decidability results in first-order logic (cf. e.g. [FLTZ93]), as noticed in [SS98a].

Alternatively, versions of *signed resolution* can be used, which generalize existing signed resolution rules for regular logics. We illustrate these ideas by presenting a version of negative hyperresolution for signed clauses, inspired by, and generalizing, the similar method established for regular clauses in [Häh96b].

5.2.1 A many-valued Herbrand universe

Let Φ be a set of signed clauses. In what follows we present an extension of the classical notion of Herbrand interpretation to many-valued logics. We present Herbrand interpretations, both as complete assignments in the algebra A, as done in [BF95], and in $\mathcal{O}(D(A))$.

Definition 12 (Herbrand universe) *The* Herbrand universe $H(\Phi)$ *of* Φ *is the set of ground terms that consist of constants and function symbols occurring in* Φ*. If there is no constant in* Φ *we introduce a special constant symbol to prevent* $H(\Phi)$ *from being empty.*

A *ground instance* C' of a clause (or atom) C in Φ is a substitution instance of C such that terms in $H(\Phi)$ replace the variables of C.

Definition 13 (Herbrand base) *The* Herbrand base $\mathbf{A}(\Phi)$ *of* Φ *is the set of all ground instances of atoms that occur in clauses in* Φ*.*

Definition 14 (Assignment in A, $D(A)$) *An* assignment *in A associates truth values in A with atoms. An assignment in $D(A)$ associates upwards-closed sets of elements in $D(A)$ with atoms.*

Definition 15 (Complete assignment in A, $D(A)$) *A* complete assignment *in A to a set of atoms $\mathbf{K}$ is a set of literals $\{P^{v(P)} \mid P \in \mathbf{K}\}$, where $v : \mathbf{K} \to A$. A* complete assignment *in $D(A)$ to a set of atoms $\mathbf{K}$ is a set of the form $\{\ \boxed{\alpha}\ P^t \mid P \in \mathbf{K}, \alpha \in m(P)\} \cup \{\ \boxed{\alpha}\ P^f \mid P \in \mathbf{K}, \alpha \notin m(P)\}$ where $m : \mathbf{K} \to \mathcal{O}(D(A))$.*

Definition 16 (Herbrand interpretation) *An Herbrand interpretation (abbreviated H-*interpretation*) of a set of clauses Φ is a complete assignment to $\mathbf{A}(\Phi)$.*

An H-interpretation M satisfies a clause set Φ if and only if for every $C \in \Phi$, all ground instances C' of C are such that $C' \cap M \neq \emptyset$. Φ is *H-unsatisfiable* if there is no H-interpretation that satisfies Φ.

Proposition 16 *A set of clauses Φ is unsatisfiable if and only if it is H-unsatisfiable.*

Proof: *(Sketch)* Every H-interpretation corresponds to a frame, as defined in Definition 2. Assume that Φ is H-satisfiable, i.e. that there exists an H-interpretation M that satisfies Φ. We can therefore construct an interpretation in A, $\mathcal{H} = (\mathsf{Term}^0_{\mathsf{O}}, I, d)$, that satisfies Φ (or alternatively an interpretation in $\mathcal{O}(D(A))$, $\mathcal{H}_M = (\mathsf{Term}^0_{\mathsf{O}}, M, d)$). Conversely, let $\mathcal{I} = (D, I, d)$ be an arbitrary interpretation. $\mathcal{I}$ induces an H-interpretation in A, $I_{\mathcal{I}} = \{P^w \mid v_{\mathcal{I}}(P) = w, P \in \mathbf{A}(\Phi)\}$ (resp. an H-interpretation in $D(A)$, $M_{\mathcal{I}} = \{\boxed{\alpha}\, P^t \mid \alpha \in \eta_A(v_{\mathcal{I}}(P)), P \in \mathbf{A}(\Phi)\} \cup \{\boxed{\alpha}\, P^f \mid \alpha \notin \eta_A(v_{\mathcal{I}}(P)), P \in \mathbf{A}(\Phi)\}$). $\square$

5.2.2 A hyperresolution procedure

We showed that, given a formula $\phi \in \mathsf{Fma}(\mathbf{L})$, the clause form transformation described in Section 5.1 constructs a set Φ of signed clauses such that ϕ is valid in $\mathcal{L}$ if and only if Φ is unsatisfiable. The literals in Φ contain free variables, and, possibly Skolem constants; and predicate symbols in the extension $\mathbf{L}^*$ of the language of $\mathcal{L}$, as defined in Definition 11.

For proving the unsatisfiability of a set Φ of signed clauses resolution techniques can be used. In this section we present a signed hyperresolution procedure that extends the signed hyperresolution for regular logics due to [Häh94, Häh96b].

Definition 17 (Substitution) *A substitution is a map assigning a term to every variable, which is the identity on all variables except possibly a finite number of them. If τ is a substitution, then we use the following notation:*

(i) if $L(x_1, \dots, x_n)$ is an atomic formula with free variables $x_1, \dots, x_n$, we denote by $\tau(L)$ the atomic formula $L(\tau(x_1), \dots, \tau(x_n))$ obtained from $L(x_1, \dots, x_n)$ by replacing every occurrence of x_i with $\tau(x_i)$ for $i \in \{1, \dots, n\}$;

(ii) if $\boxed{\alpha}\, L^v$ is a signed literal (where $v \in \{t, f\}$), we denote by $\tau(\boxed{\alpha}\, L^v)$ the literal $\boxed{\alpha}\, \tau(L)^v$, obtained by replacing every occurrence of a free variable x in L with $\tau(x)$;

(iii) if $C = \boxed{\alpha_1}\, L_1^{v_1} \Cup \dots \Cup \boxed{\alpha_n}\, L_n^{v_n}$ is a signed clause, we denote by $\tau(C)$ the clause obtained by applying τ to all the signed literals in C.

Note that substitutions do not act on signs, but only on atoms.

Definition 18 (m.g.u) *Let $\boxed{\alpha_1}\, L_1^{v_1}$ and $\boxed{\alpha_2}\, L_2^{v_2}$ be two signed literals. A substitution τ is a* most general unifier *(abbreviated m.g.u. in what follows) of $\boxed{\alpha_1}\, L_1^{v_1}$ and $\boxed{\alpha_2}\, L_2^{v_2}$ if $\alpha_1 = \alpha_2$, $v_1 = v_2$, and τ is a most general unifier of L_1 and L_2 (in the usual sense).*

Definition 19 (Factor) *Let C be a signed clause. If two or more positive literals of C have an m.g.u. τ, then $\tau(C)$ is called a* factor *of C. Likewise, if two or more negative literals of C have an m.g.u. τ, then $\tau(C)$ is called a* factor *of C.*

Definition 20 (Binary Resolvent; Resolvent) *Let C_1 and C_2 be signed clauses with no variable in common. Let $\boxed{\alpha}\, L_1^t$ and $\boxed{\beta}\, L_2^f$ be signed literals occurring in C_1 and C_2, respectively. If $\alpha \leq \beta$ and L_1 and L_2 have an m.g.u. τ, then the clause $C_3 := (\tau(C_1) - \boxed{\alpha}\, \tau(L_1)^t) \cup (\tau(C_2) - \boxed{\beta}\, \tau(L_2)^f)$ is called a* binary resolvent *of C_1 and C_2. A* resolvent *of two clauses C_1 and C_2 is a binary resolvent of (a factor of) C_1 and (a factor of) C_2.*

Definition 21 (Semantic clash; Resolvent by hyperresolution) *A finite set of clauses* $\{E_1, \dots, E_q, N\}, q \geq 1$ *is called a* semantic clash *if and only if* $E_1, \dots, E_q$ *(called* electrons*) and* N *(called* nucleus*) satisfy the following conditions:*

1. $E_1, \dots, E_q$ *are negative clauses;* N *is a non-negative clause;*
2. *Let* $R_1 = N$*. For each* $i = 1, \dots, q$*, there is a resolvent* R_{i+1} *of* R_i *and* E_i*;*
3. R_{q+1} *is a negative clause.*

R_{q+1} *is called a* resolvent by hyperresolution *of the semantic clash* $\{E_1, \dots, E_q, N\}$*.*

The principle of signed negative hyperresolution can be represented schematically as follows:

$$\frac{\left\{\boxed{\beta_1}\ L_1^f\right\} \cup D_1, \dots, \left\{\boxed{\beta_n}\ L_n^f\right\} \cup D_n, \left\{\boxed{\alpha_1}\ L_1^t, \dots, \boxed{\alpha_n}\ L_n^t\right\} \cup E}{D_1 \cup \dots \cup D_n \cup E}$$

provided that $n \geq 1$, $\alpha_i \leq \beta_i$ for all $i = 1, \dots, n$ and $D_1, \dots, D_n, E$ are negative.

Lemma 17 (Ground Clauses) *Let* Φ *be a finite unsatisfiable set of signed ground clauses. Then* $\Box$ *can be derived from* Φ *by a finite number of applications of many-valued negative hyperresolution.*

Proof: In order to prove the completeness of many-valued negative hyperresolution we show that the proof presented in [Häh96b], which was taken virtually unaltered from [AB70], works as well in this case.

If $\Box \in \Phi$ then the conclusion is obvious. Therefore in what follows we will assume that $\Box \notin \Phi$. Note first that, since the clauses in Φ are ground, the Herbrand basis of Φ, $\mathbf{A}(\Phi)$, coincides with the set of all literals of Φ. Let $nl(\Phi)$ be the total number of literals in Φ and $nc(\Phi)$ the total number of clauses in Φ. Since we assumed that $\Box \notin \Phi$, it is obvious that $nl(\Phi) \geq nc(\Phi)$. We will proceed by induction on the difference $k(\Phi) = nl(\Phi) - nc(\Phi)$. We distinguish the following cases:

Case 1: $k(\Phi) = 0$: In this case $nl(\Phi) = nc(\Phi)$. Since $\Box \notin \Phi$, Φ must consist only of unit clauses. We prove that there must exist two clauses $\{\boxed{\alpha}\ L^t\}$ and $\{\boxed{\beta}\ L^f\}$ in Φ such that $\alpha \leq \beta$. Assume that for every two clauses C_1, C_2 in Φ, if $C_1 = \boxed{\alpha}\ L^t$ and $C_2 = \boxed{\beta}\ L^f$ then $\alpha \not\leq \beta$. In this case we can construct a valuation $m : \mathbf{A}(\Phi) \to \mathcal{O}(D(A))$ (and, hence, an H-interpretation) that satisfies Φ. Indeed, for every $L \in \mathbf{A}(\Phi)$ let $m(L) = \uparrow\{\alpha \mid \boxed{\alpha}\ L^t \in \Phi\}$. It is easy to see that m has order-filters as values, and that m satisfies Φ: if $\boxed{\alpha}\ L^t \in \Phi$ then $\alpha \in m(L)$, whereas if $\boxed{\beta}\ L^f \in \Phi$ then $\beta \notin m(L)$ (otherwise there would exist $\alpha \leq \beta$, with $\boxed{\alpha}\ L^t \in \Phi$). Since Φ is unsatisfiable, this is a contradiction. Hence, there exist two clauses $\{\boxed{\alpha}\ L^t\}$ and $\{\boxed{\beta}\ L^f\}$ in Φ such that $\alpha \leq \beta$. Then $\Box$ can be derived from these clauses by negative hyperresolution.

Case 2: $k(\Phi) > 0$ (i.e. $nl(\Phi) > nc(\Phi)$).
Subcase 2a: Assume that all non-positive clauses consist of one literal. Then it follows that all negative literals in Φ appear in unit clauses. We prove that there is a positive clause in Φ which immediately produces the empty clause with suitable negative unit clauses. Let $m : \mathbf{A}(\Phi) \to \mathcal{O}(D(A))$ be defined by $m(L) = D(A)$ for every L that does not occur in a negative clause in Φ, and by $m(L) = \{\alpha \mid$

$\forall\beta(\{\boxed{\beta}\, L^f\} \in \Phi$ implies $\beta \not\geq \alpha)\}$ otherwise. It is easy to see that (i) m has as values order filters, and (ii) m is a valuation that defines an H-interpretation which makes true all negative clauses. Since Φ is unsatisfiable, this H-interpretation cannot make Φ true. Hence, there exists a positive clause $C = \{\, \boxed{\alpha_1}\, L_1^t, \ldots, \boxed{\alpha_n}\, L_n^t\}$ such that m does not make C true. Then for every $i = 1, \ldots, n$, $\alpha_i \notin m(L_i)$, i.e. there exists $C_i = \boxed{\beta_i}\, L_i^f \in \Phi$ with $\alpha_i \leq \beta_i$. It follows therefore $C, C_1, \ldots, C_n$ yield the empty clause by hyperresolution.
Subcase 2b: There exists a non-positive non-unit clause $C = \{\, \boxed{\beta}\, L^f\} \cup D \in \Phi$ with $D \neq \Box$. Let $\Phi' = \Phi \backslash \{C\}$, $\Phi_1 = \Phi' \cup D$, and $\Phi_2 = \Phi' \cup \{\{\, \boxed{\beta}\, L^f\}\}$. Since Φ is unsatisfiable it follows that both Φ_1, Φ_2 are unsatisfiable. Moreover, Φ_1 and Φ_2 contain the same number of clauses as Φ, but they have fewer literals. Hence, $k(\Phi_1) < k(\Phi)$ and $k(\Phi_2) < k(\Phi)$. Since Φ_1 and Φ_2 are both unsatisfiable, by the induction hypothesis, $\Box$ can be deduced from Φ_1 and $\Box$ can be deduced from Φ_2 by hyperresolution. Consider a hyperresolution deduction of $\Box$ from Φ_1. If we replace each occurrence of D by C in this proof then we obtain a valid hyperresolution deduction with last clause $\Box$ or $\{\, \boxed{\beta}\, L^f\}$. If the last clause is $\Box$, we have already a hyperresolution deduction of $\Box$ from Φ. If the last clause is $\{\, \boxed{\beta}\, L^f\}$, then its deduction from Φ can be extended to a deduction of $\Box$ from Φ using the fact that there is a deduction of $\Box$ from Φ_2. $\Box$

Lemma 18 (Lifting Lemma) *If C_1' and C_2' are instances of C_1 and C_2 respectively, and if C' is a resolvent of C_1' and C_2', then there is a resolvent C of C_1 and C_2 such that C' is an instance of C.*

Proof: The proof is an easy adaptation of the proof of the classical lifting lemma (cf. e.g. [CL73]), taking into account the notion of resolvent we used here. $\Box$

The following theorem states the correctness and completeness of the signed hyperresolution procedure:

Theorem 19 *The following hold:*

(1) For any set Φ of signed clauses, if the empty clause can be derived from Φ by signed negative hyperresolution, then Φ is unsatisfiable.

(2) Let Φ be a finite unsatisfiable set of signed clauses. Then $\Box$ can be derived from Φ by a finite number of applications of signed negative hyperresolution.

Proof: (1) Follows from the fact that an H-interpretation that satisfies a set of clauses also satisfies all their factors and their resolvents. But no H-interpretation satisfies the empty clause.

(2) Follows exactly as in the case of classical logic, using Lemma 17 and the lifting lemma, Lemma 18. (The proof of Theorem 6.1 in [CL73], Section 6.4 can be adapted without any problem.) $\Box$

An ordered version of hyperresolution can be defined as well. Assume given an order on atoms which is stable with respect to substitutions. If in the definition of a semantic clash $\{E_1, \ldots, E_q, N\}, q \geq 1$ we additionally impose the condition that, for each $i = 1, \ldots, q$, the resolvent R_{i+1} of R_i and E_i has the property that the literal in E_i which is resolved upon contains the largest atom in $E_i, i = 1, \ldots, q$, we obtain a notion of *ordered semantic clash.* (Note that several signed literals in E_i may contain the largest atom, possibly with different signs.) The proof of Lemma 17 remains essentially the same with this ordered version of hyperresolution: in **Case 2b** one has to additionally assume that the non-positive non-unit clause

$C = \{\boxed{\beta}\ L^f\} \cup D \in \Phi$ is chosen such that $\boxed{\beta}\ L^f$ is minimal among all negative literals in Φ which occur in non-positive non-unit clauses. Under this additional assumption, it is then easy to check that all hyperresolution inferences respect the ordering constraints (negative literals smaller than $\boxed{\beta}\ L^f$ only occur in negative unit clauses, and are resolved immediately, hence do not occur in the resolvents).

5.2.3 Further remarks

Note the similarity of the signed hyperresolution method described in this section and the classical hyperresolution method. The explanation for this relies on the fact that the method for clause-form transformation described in Section 5.1 can actually be seen as a translation to classical logic, as pointed out at the beginning of Section 5.2. This may be better explained by an alternative formulation of the Priestley representation theorem for finite distributive lattices: Every finite bounded distributive lattice A is isomorphic to the lattice of all order-preserving maps from its Priestley dual $D(A)$ into the partially-ordered set with two elements [Pri70, Pri72]. In the presence of additional operators this is true too (the dual $D(A)$ of A is then also endowed with a family of maps and relations that induce corresponding operators on the lattice of all order-preserving maps from $D(A)$ to $\{0,1\}$). $D(A)$ can be thus seen as a Kripke-style frame, where at every possible world the valuations are 2-valued.

This is also the case, in a less obvious way, with the regular logics discussed in [Häh94].

6 Examples

In this section we illustrate the method presented before by means of several examples, such as SH_n-logics, SHK_n-logics, and P_{mn}-logics.

6.1 SH_n-logics

The language of propositional SH_n-logics is a propositional language, whose formulae are built from propositional variables taken from a set Var, with operations $\vee$ (disjunction), $\wedge$ (conjunction), $\Rightarrow$ (intuitionistic implication), $\sim, \neg$ (a De Morgan resp. an intuitionistic negation), and a family $\{S_i \mid i = 1, \ldots, n-1\}$ of unary operations. A Hilbert style axiomatization of propositional SH_n-logics can be found in [Itu82, IO96]; algebraic and relational semantics for SH_n-logics have been studied in, respectively, [Itu83] and [IO96]. For the main definitions we refer to [Itu83, IO96] or to [SS98b].

Iturrioz [Itu82] showed that propositional SH_n-logics are sound and complete with respect to the variety $\mathsf{SH_n}$ of SH_n-algebras, i.e. that a formula ϕ is an SH_n-theorem if and only if it is satisfied by every SH_n-algebra. SH_n-algebras are Heyting algebras endowed with a De Morgan negation and $n-1$ additional operations $S_1, \ldots, S_{n-1}$ that satisfy certain conditions (for the definition see e.g. [Itu83]).

Let $n \geq 2$ and let $L_n = \{0, \frac{1}{n-1}, \ldots, \frac{n-2}{n-1}, 1\}$, where $0 < \frac{1}{n-1} < \cdots < \frac{n-2}{n-1} < 1$. Let $\Rightarrow$ be the Heyting relative pseudocomplementation on L_n, i.e., for every $a, b \in L_n$, $a \Rightarrow b = \begin{cases} 1 & \text{if } a \leq b \\ b & \text{if } a > b \end{cases}$ and $\neg a := a \Rightarrow 0 = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{if } a > 0 \end{cases}$. Additionally, let $\vee$ and $\wedge$ be defined as the maximum, resp. mininum; and $S_i : L_n \to L_n$ be defined by $S_i(\frac{j}{n-1}) = \begin{cases} 1 & \text{if } i + j \geq n, \\ 0 & \text{if } i + j < n, \end{cases}$.

Iturrioz [Itu83] showed that for every $n \geq 2$ the variety $\mathsf{SH_n}$ is generated by the finite algebra S_{n^2} (represented in the left-hand side of Figure 3), defined as follows:

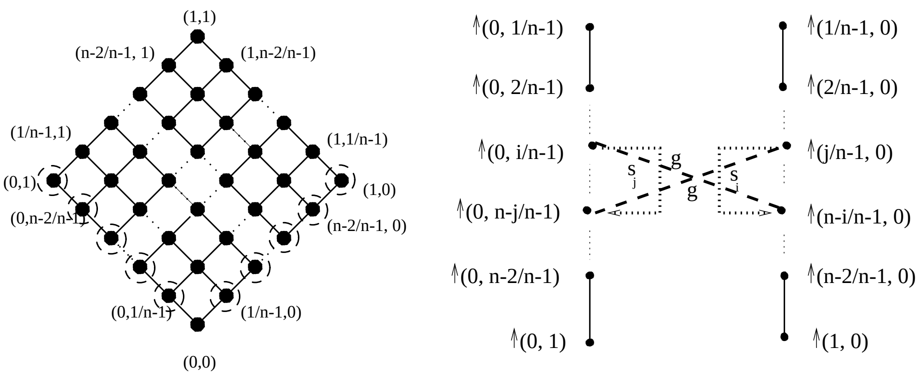

Figure 3: S_{n^2} and its Priestley dual

Definition 22 $S_{n^2} = (L_n \times L_n, \vee, \wedge, \sim, S_1, \ldots, S_{n-1}, \neg, \Rightarrow, 0, 1)$ *where the operations* $0, 1, \vee, \wedge, S_1, \ldots, S_{n-1}, \neg$ *and* $\Rightarrow$ *are defined componentwise from the operations on* L_n *mentioned above, and for every* $(x, y) \in L_n \times L_n$, $\sim(x, y) = (1-y, 1-x)$.

Since the variety $\mathsf{SH_n}$ is generated by S_{n^2}, SH_n-logics can be seen as many-valued logics having the finite algebra S_{n^2} as an algebra of truth values.

Remark. Intuitively, we can think of SH_n-logics as the logics of two agents, both of them n-valued. The logics of each of these agents have as connectives $\mathsf{C}_{L_n} = \{\vee, \wedge, \neg, \Rightarrow, S_1, \ldots, S_{n-1}\}$ (with the truth tables on L_n as defined above). Additionally, a joint weak negation $\sim$ is defined. This is the global aspect that binds the two agents together. In order to determine the truth value of a formula ϕ in this logic, the formula is evaluated recursively by both agents as follows: If the formula is atomic or starts with connectives in the set C_{L_n} then the evaluation is done by each of the agents, individually, and the result of the evaluation is an ordered pair, containing the two evaluations. If the formula is of the form $\phi = \sim\psi$, then the agents evaluate (recursively) the formula ψ, and communicate their evaluation of ψ to each other. Then each of them adopts as an evaluation of $\phi = \sim\psi$ a kind of negation (obtained by subtraction from 1) of the other's evaluation of ψ.

Representation theorems for SH_n-algebras are studied in [Itu83] and [SS97, SS99a]. The Priestley dual of S_{n^2}, $D(S_{n^2})$, represented in the right-hand side of Figure 3, is isomorphic to the ordered set of the join-irreducible elements of S_{n^2} (encircled in Figure 3), i.e.

$$\left\{ \left(0, \frac{i}{n-1}\right) \mid i = 1, \ldots, n-1 \right\} \cup \left\{ \left(\frac{i}{n-1}, 0\right) \mid i = 1, \ldots, n-1 \right\}$$

with the reversed order. These elements correspond to the prime filters $\uparrow(0, \frac{i}{n-1}), i = 1, \ldots, n-1$ resp. $\uparrow(\frac{i}{n-1}, 0), i = 1, \ldots, n-1$. The additional operations $g, s_1, \ldots, s_{n-1}$ are defined by:

$$g\left(\uparrow\left(0, \frac{i}{n-1}\right)\right) = \uparrow\left(\frac{n-i}{n-1}, 0\right), \quad g\left(\uparrow\left(\frac{i}{n-1}, 0\right)\right) = \uparrow\left(0, \frac{n-i}{n-1}\right),$$

$$s_j\left(\uparrow\left(0, \frac{i}{n-1}\right)\right) = \uparrow\left(0, \frac{n-j}{n-1}\right), \quad s_j\left(\uparrow\left(\frac{i}{n-1}, 0\right)\right) = \uparrow\left(\frac{n-j}{n-1}, 0\right).$$

The operations on $D(S_{n^2})$ induce, in their turn, operations $\sim_{\mathcal{O}(D(S_{n^2}))}$ $S_{1_{\mathcal{O}(D(S_{n^2}))}}$, $\ldots$, $S_{n-1_{\mathcal{O}(D(S_{n^2}))}}$ on $\mathcal{O}(D(S_{n^2}))$ as explained in Section 3.1.2. Moreover, a Heyting implication $\Rightarrow$ resp. negation $\neg$ can be defined in a canonical way on $\mathcal{O}(D(S_{n^2}))$ by

$$U_1 \Rightarrow U_2 = \{\alpha \mid \forall \beta \geq \alpha, \text{ if } \beta \in U_1 \text{ then } \beta \in U_2\},$$

$$\neg U = \{\alpha \mid \forall \beta \geq \alpha, \beta \notin U\}.$$

For the sake of simplicity, in what follows we will omit the index $\mathcal{O}(D(S_{n^2}))$ in these operations.

Taking into account these definitions, a satisfiability notion $\overset{r}{\models}$ can be defined also in this case, as explained in Definition 6. $\overset{r}{\models}$ is very similar to the notion of satisfiability with respect to the Kripke-style models[6] defined in [IO96].

The considerations in Section 4.1 lead to the following result.

Corollary 20 *Let ϕ be a formula in the language of the SH_n-propositional logic with variables* Var. *Then $\vdash_{SH_n} \phi$ if and only if $D(S_{n^2}) \overset{r}{\models} \phi$.*

It is easy to see that the general considerations in Section 4.2 can be used to define, for every number n, a many-valued first-order logic having the algebra S_{n^2} as algebra of truth values, where the quantifiers $\forall$ and $\exists$ are induced by the binary connectives $\wedge$ and respectively $\vee$. The method for automated theorem proving described in Section 5 can thus be applied to SH_n-logics. The rules for translation to clause form in the case of SH_n-logics for $\vee, \wedge, \sim, S_1, \ldots, S_{n-1}$ as well as for the quantifiers are given by Lemma 11 and Lemma 12. We present the translation rules for $\Rightarrow, \neg$. For this, we start by proving some general properties.

Lemma 21 *Let L be an atomic formula, $\alpha \in D(S_{n^2})$, and let $m = \mathsf{max}\{\beta \in D(S_{n^2}) \mid \beta \geq \alpha\}$. Then the following hold for every interpretation $\mathcal{M}$ in $\mathcal{O}(D(S_{n^2}))$:*

(1) Every non-empty set $S \subseteq \{\beta \mid \beta \geq \alpha\}$ contains a maximal element $\mathsf{max}(S)$ and a minimal element $\mathsf{min}(S)$. Moreover, (i) $\bigvee_{\beta\in S} \boxed{\beta}\ L^t$ if and only if $\boxed{\mathsf{max}(S)}\ L^t$, and (ii) $\bigvee_{\beta\in S} \boxed{\beta}\ L^f$ if and only if $\boxed{\mathsf{min}(S)}\ L^f$.

(2) If $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = \{\beta \mid \beta \geq \alpha\}$ then (i) $\mathsf{max}(S_1) \neq \mathsf{min}(S_2)$, and (ii) if $\mathsf{max}(S_1) \neq m$ then $\mathsf{min}(S_2) \leq s(\mathsf{max}(S_1))$, where $s(\mathsf{max}(S_1))$ is the immediate successor of $\mathsf{max}(S_1)$ in the ordering on $D(S_{n^2})$.

(3) Let $\alpha \leq \beta < m$. Then $\bigwedge_{S,\mathsf{max}(S)=\beta} \boxed{\mathsf{min}(\uparrow\alpha\backslash S)}\ L^f$ if and only if $\boxed{s(\beta)}\ L^f$.

Proof: (1) The first part follows immediately from the fact that, for every $\alpha \in D(S_{n^2})$, the set $\{\beta \mid \beta \geq \alpha\}$ is finite and totally ordered. In order to prove (i), note that if there exists $\beta \in S$ with $\boxed{\beta}\ L^t$ then also $\boxed{\mathsf{max}(S)}\ L^t$ because $\beta \leq \mathsf{max}(S)$. The converse implication follows from the fact that $\mathsf{max}(S) \in S$. In order to prove (ii), note that if $\boxed{\mathsf{min}(S)}\ L^t$ then for every $\beta \in S$, $\boxed{\beta}\ L^t$. The converse implication follows from the fact that $\mathsf{min}(S) \in S$.

(2) (i) follows immediately from the fact that $S_1 \cap S_2 = \emptyset$. In order to prove (ii), note that if $\mathsf{max}(S_1) \neq m$ then $\gamma = s(\mathsf{max}(S_1)) \notin S_1$, hence it belongs to S_2. Therefore, $\mathsf{min}(S_2) \leq s(\mathsf{max}(S_1))$.

(3) Assume that $\bigwedge_{S,\mathsf{max}(S)=\beta} \boxed{\mathsf{min}(\uparrow\alpha\backslash S)}\ L^f$, i.e. that for every $S \subseteq \uparrow\alpha$ such that $\mathsf{max}(S) = \beta$, $\boxed{\mathsf{min}(\uparrow\alpha\backslash S)}\ L^f$. In particular, let $S = [\alpha, \beta] = \{\gamma \in D(A) \mid$

[6]The link between the algebraic and Kripke-style models for SH_n-logics can be explained using Priestley duality for SH_n-algebras, see e.g. [SS99a].

$\alpha \leq \gamma \leq \beta\}$. Then $\mathsf{max}(S) = \beta$ and $\uparrow\alpha\backslash S = \uparrow s(\beta)$, i.e., $\mathsf{min}(\uparrow\alpha\backslash S) = s(\beta)$. Hence, $\boxed{s(\beta)}\, L^f$. To prove the converse, note that, by (2) (ii), if $\mathsf{max}(S) = \beta$, then $\mathsf{min}(\uparrow\alpha\backslash S) \leq s(\beta)$. This shows that if for some S $\boxed{\mathsf{min}(\uparrow\alpha\backslash S)}\, L^t$, then $\boxed{s(\beta)}\, L^t$. □

Lemma 22 *Let $\alpha \in D(S_{n^2})$ be arbitrary and let $m = \mathsf{max}\{\beta \in D(S_{n^2}) \mid \beta \geq \alpha\}$. Then the following hold for any atomic formulae L, L_1, L_2, and any given interpretation $\mathcal{M}$.*

$(\neg, t)$	$\boxed{\alpha}\,(\neg L)^t$	iff	$\boxed{m}\, L^f$;
$(\neg, f)$	$\boxed{\alpha}\,(\neg L)^f$	iff	$\boxed{m}\, L^t$;
$(\Rightarrow, t)$	$\boxed{\alpha}\,(L_1 \Rightarrow L_2)^t$	iff	for every $\beta \geq \alpha$, $\boxed{\beta}\, L_1^f \Cup \boxed{\beta}\, L_2^t$;
$(\Rightarrow, f)$	$\boxed{\alpha}\,(L_1 \Rightarrow L_2)^f$	iff	$\boxed{m}\, L_1^t$, $\boxed{\alpha}\, L_2^f$, and, for every α_1 with $\alpha \leq \alpha_1 < m$, $\boxed{\alpha_1}\, L_1^t \Cup \boxed{s(\alpha_1)}\, L_2^f$, where $s(\alpha_1)$ is the immediate successor of α_1 in $D(S_{n^2})$.

Proof: The cases $(\neg,\ t)$, $(\neg,\ f)$, and $(\Rightarrow, t)$ are only sketched; only $(\Rightarrow, f)$ is presented in detail.

$(\neg,\ t)$ $\boxed{\alpha}\,(\neg L)^t$ iff $\alpha \in v_{\mathcal{M}}(\neg L) = \neg v_{\mathcal{M}}(L)$ iff $\forall \beta \geq \alpha$, $\beta \notin v_{\mathcal{M}}(L)$ iff $\forall \beta \geq \alpha$, $\boxed{\beta}\, L^f$ in $\mathcal{M}$. From Lemma 21(1) it follows that $\bigwedge_{\beta \geq \alpha} \boxed{\beta}\, L^f$ iff $\boxed{m}\, L^f$.

$(\neg, f)$ $\boxed{\alpha}\,(\neg L)^f$ iff $\alpha \notin v_{\mathcal{M}}(\neg L) = \neg v_{\mathcal{M}}(L)$ iff $\exists \beta \geq \alpha$, $\beta \in v_{\mathcal{M}}(L)$ iff $\exists \beta \geq \alpha$, $\boxed{\beta}\, L^t$. The conclusion follows from Lemma 21(1) .

$(\Rightarrow, t)$ $\boxed{\alpha}\,(L_1 \Rightarrow L_2)^t$ iff $\alpha \in v_{\mathcal{M}}(L_1 \Rightarrow L_2) = v_{\mathcal{M}}(L_1) \Rightarrow v_{\mathcal{M}}(L_2)$ iff $\forall \beta \geq \alpha, \beta \in v_{\mathcal{M}}(L_1)$ implies $\beta \in v_{\mathcal{M}}(L_2)$, iff $\forall \beta \geq \alpha$, $\boxed{\beta}\, L_1^t$ implies $\boxed{\beta}\, L_2^t$.

$(\Rightarrow, f)$ It is easy to see that $\boxed{\alpha}\,(L_1 \Rightarrow L_2)^f$ iff for some $\beta \geq \alpha$, $\boxed{\beta}\, L_1^t$ and $\boxed{\beta}\, L_2^f$. By distributivity, the formula $\bigvee_{\beta \geq \alpha}(\boxed{\beta}\, L_1^t \Cap \boxed{\beta}\, L_2^f)$ can be written as

$$\bigwedge_{\substack{S_1, S_2, S_1 \cap S_2 = \emptyset \\ S_1 \cup S_2 = \{\beta \mid \beta \geq \alpha\}}} \left(\bigvee_{\beta_1 \in S_1} \boxed{\beta_1}\, L_1^t \Cup \bigvee_{\beta_2 \in S_2} \boxed{\beta_2}\, L_2^f \right).$$

Hence, the following holds:

$\boxed{\alpha}\,(L_1 \Rightarrow L_2)^f$ iff $\bigwedge_{\substack{S_1, S_2, S_1 \cap S_2 = \emptyset \\ S_1 \cup S_2 = \{\beta \mid \beta \geq \alpha\}}} \left(\bigvee_{\beta_1 \in S_1} \boxed{\beta_1}\, L_1^t \Cup \bigvee_{\beta_2 \in S_2} \boxed{\beta_2}\, L_2^f \right)$

iff $\boxed{m}\, L_1^t \Cap \boxed{\alpha}\, L_2^f \Cap \bigwedge_{\substack{S_1, S_2 \neq \emptyset, S_1 \cap S_2 = \emptyset \\ S_1 \cup S_2 = \{\beta \mid \beta \geq \alpha\}}} \left(\boxed{\mathsf{max}(S_1)}\, L_1^t \Cup \boxed{\mathsf{min}(S_2)}\, L_2^f \right)$

(the cases $S_2 = \emptyset$ and $S_1 = \emptyset$ were considered separately; for $S_1 \neq \emptyset$ resp. $S_2 \neq \emptyset$ Lemma 21 (1) was used.)

iff $\boxed{m}\, L_1^t \Cap \boxed{\alpha}\, L_2^f \Cap \bigwedge_{\beta \geq \alpha} \bigwedge_{S_1, \mathsf{max}(S_1) = \beta} \left(\boxed{\beta}\, L_1^t \Cup \boxed{\mathsf{min}(\uparrow\alpha\backslash S_1)}\, L_2^f \right)$

iff $\boxed{m}\, L_1^t \Cap \boxed{\alpha}\, L_2^f \Cap \bigwedge_{S_1, \mathsf{max}(S_1) = m} \left(\boxed{m}\, L_1^t \Cup \boxed{\mathsf{min}(\uparrow\alpha\backslash S_1)}\, L_2^f \right) \Cap$
$\bigwedge_{\alpha \leq \beta < m} \bigwedge_{S_1, \mathsf{max}(S_1) = \beta} \left(\boxed{\beta}\, L_1^t \Cup \boxed{\mathsf{min}(\uparrow\alpha\backslash S_1)}\, L_2^f \right)$
(the case when $\beta = m$ was considered separately)

iff $\boxed{m}\, L_1^t \Cap \boxed{\alpha}\, L_2^f \Cap \bigwedge_{\alpha \leq \beta < m} \left(\boxed{\beta}\, L_1^t \Cup \bigwedge_{S_1, \mathsf{max}(S_1) = \beta} \boxed{\mathsf{min}(\uparrow\alpha\backslash S_1)}\, L_2^f \right)$
(by absorption and distributivity for $\Cup$, $\Cap$)

iff $\boxed{m}\, L_1^t \Cap \boxed{\alpha}\, L_2^f \Cap \bigwedge_{\alpha \leq \beta < m} \left(\boxed{\beta}\, L_1^t \Cup \boxed{s(\beta)}\, L_2^f \right)$.
(by Lemma 21 (3))

$\square$

The number of clauses generated by the above mentioned procedure is given by the following proposition.

Proposition 23 *Let $n \geq 2$ be a natural number and let ϕ be a formula with l non-atomic subformulae. The number of clauses generated from ϕ is in $O(n^2 \cdot l)$. If the formula ϕ does not contain the connective $\Rightarrow$, then the number of clauses generated from ϕ is in $O(n \cdot l)$.*

Proof: By Proposition 9, the number of clauses generated from a given formula ϕ is $1 + N_\sigma + N_Q$, where N_σ (resp. N_Q) is the number of clauses generated by those subformulae of ϕ which start with a logical connective (resp. a quantifier). Since all the rules for quantifiers have branching factor 1, N_Q is in $O(n \cdot l)$.

Concerning subformulae starting with a logical connective, note that the maximal number of clauses is generated by the subformulae of the form $\psi = \psi_1 \Rightarrow \psi_2$. In this case, by Lemma 22, for every $\alpha \in D(S_{n^2})$ the number of clauses generated by ($\boxed{\alpha}\ P^f_\psi \Cup \boxed{\alpha}\ (P_{\psi_1} \Rightarrow P_{\psi_2})^t$) and ($\boxed{\alpha}\ P^t_\psi \Cup \boxed{\alpha}\ (P_{\psi_1} \Rightarrow P_{\psi_2})^f$) is bounded by $2|\{\beta \mid \beta \geq \alpha\}| + 1$. Thus, the number of clauses generated from the formula ϕ has as upper bound $1 + 2 \cdot l \cdot \Sigma_{i=1}^{n-1}(2i+1) = 1 + 2 \cdot l \cdot (n^2 - 1)$, hence is in $O(n^2 l)$.

If ϕ does not contain the connective $\Rightarrow$ then, for every $\alpha \in D(S_{n^2})$ and every subformula ψ of ϕ the number of clauses generated from ψ at α is at most 3. Thus, the number of clauses generated from ϕ has as upper bound $1 + 6l(n-1)$, so it is in $O(n \cdot l)$. $\square$

Since in this case the set of truth values has n^2 elements, the method for translation to clause form described in this paper is at most linear in the number of truth values; for formulae not containing the Heyting implication the complexity is linear with respect to the square root of the number of truth values.

Further improvements. The structure of $D(S_{n^2})$ can be used in order to further reduce the number of clauses[7]. For this, we use the fact that $D(S_{n^2})$ consists of two branches and that the transformation rules for the operations in $\{\vee, \wedge, \neg, \Rightarrow\}$ preserve the branch of $D(S_{n^2})$. For formulae that do not contain the De Morgan negation $\sim$, all the clauses generated by the renaming of subformulae only contain signs in one of the branches of $D(S_{n^2})$. It is sufficient to give a refutation for the clauses corresponding to one of the branches of $D(S_{n^2})$. For the other branch a similar refutation can be constructed by simply renaming the nodes, and they can be then combined to a refutation by resolution for the clause form of ϕ.

6.2 SHK_n-logics

Let SHK_n be the logic obtained from the SH_n logic by adjoining the Kleene axiom:

$$(a \wedge {\sim} a) \Rightarrow (b \vee {\sim} b).$$

SHK_n-logics are sound and complete with respect to the subvariety $\mathsf{SHK_n}$ of $\mathsf{SH_n}$, consisting of those SH_n-algebras that satisfy the Kleene law (cf. [Itu83]):

$$(a \wedge {\sim} a) \leq (b \vee {\sim} b).$$

It is known [Itu83] that the variety $\mathsf{SHK_n}$ coincides with the variety of Łukasiewicz-Moisil algebras, and is, hence, generated by the finite algebra

$$L_n = (\{0, \frac{1}{n-1}, \ldots, \frac{n-2}{n-1}, 1\}, \vee, \wedge, \sim, S_1, \ldots, S_{n-1}, \neg, \Rightarrow, 0, 1),$$

[7]We thank Luisa Iturrioz who suggested that it may be possible to further improve the efficiency of automated theorem proving for SH_n-logics by exploiting the structure of $D(S_{n^2})$; the remarks below show that this is possible; a more thorough investigation, also for more general logics, is planned for future work.

where $x \Rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{if } x > y \end{cases}$, $\neg x := x \Rightarrow 0$, $x \vee y = \mathsf{max}(x, y), x \wedge y = \mathsf{min}(x, y), {\sim} x = 1 - x$, and for every $i \in \{1, \ldots, n-1\}$, $S_i(\frac{j}{n-1}) = \begin{cases} 1 & \text{if } i + j \geq n \\ 0 & \text{if } i + j < n \end{cases}$.
Kripke models for SHK_n-logics have been investigated in [IO96]: SHK_n-frames are defined as SH_n-frames that satisfy the following additional condition:

$$(K) \qquad R(x, g(x)) \text{ or } R(g(x), x).$$

A representation theorem for SHK_n-algebras is given for instance in [SS97]. Let

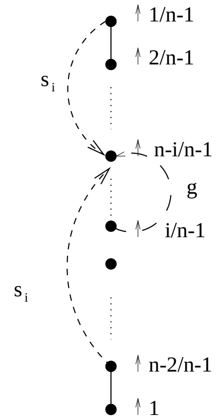

Figure 4: $D(L_n)$: The Priestley Dual of L_n

$D(L_n)$ be the Priestley dual of the n-element Łukasiewicz-Moisil algebra (represented in Figure 4). $D(L_n)$ is isomorphic to the ordered set of the join-irreducible elements of L_n, namely $\{\frac{1}{n-1}, \frac{2}{n-1}, \ldots, \frac{n-2}{n-1}, 1\}$ with the reversed order. These elements correspond, respectively, to the prime filters $\uparrow\frac{i}{n-1}$, $i = 1, \ldots, n-1$. The additional operations $g, s_1, \ldots, s_{n-1}$ are defined by $g(\uparrow\frac{i}{n-1}) = \uparrow\frac{n-i}{n-1}$ and $s_j(\uparrow\frac{i}{n-1}) = \uparrow\frac{n-j}{n-1}$, for every $i \in \{1, \ldots, n-1\}$.

The transformation rules for the connectives $\vee, \wedge, \sim$, and $S_1, \ldots, S_{n-1}$ are those in Lemma 11. The ones for the Heyting implication and negation are given below. They are essentially the same as those given in Lemma 22 for SH_n-logics. The slight simplification is due to the fact that $D(L_n)$ consists of only one branch, which is totally ordered, with maximal element $\uparrow\frac{1}{n-1}$.

Lemma 24 *Let $\alpha \in D(L_n)$, and L, L_1, L_2 be arbitrary atomic formulae. Then the following holds in every interpretation $\mathcal{M}$ in $\mathcal{O}(D(L_n))$:*

$(\neg, t)$ $\quad \boxed{\alpha}\, (\neg L)^t \quad$ iff $\quad \boxed{\uparrow\frac{1}{n-1}}\, L^f$

$(\neg, f)$ $\quad \boxed{\alpha}\, (\neg L)^f \quad$ iff $\quad \boxed{\uparrow\frac{1}{n-1}}\, L^t$

$(\Rightarrow, t)$ $\quad \boxed{\alpha}\, (L_1 \Rightarrow L_2)^t \quad$ iff $\quad$ for every $\beta \geq \alpha$, $\boxed{\beta}\, L_1^f \Cup \boxed{\beta}\, L_2^t$,

$(\Rightarrow, f)$ $\quad \boxed{\alpha}\, (L_1 \Rightarrow L_2)^f \quad$ iff $\quad \boxed{\uparrow\frac{1}{n-1}}\, L_1^t$, $\boxed{\alpha}\, L_2^f$, and, for every α_1 with $\alpha \leq \alpha_1 < \uparrow\frac{1}{n-1}$, $\boxed{\alpha_1}\, L_1^t \Cup \boxed{s(\alpha_1)}\, L_2^f$, where $s(\alpha_1)$ is the immediate successor of α_1 in $D(S_{n^2})$.

The number of clauses generated by the above mentioned procedure is given by the following proposition.

Proposition 25 *Let $n \geq 2$ be a natural number and let ϕ be a formula with l non-atomic subformulae. The number of clauses generated from ϕ is in $O(n^2 \cdot l)$. If the formula ϕ does not contain the connective $\Rightarrow$, then the number of clauses generated from ϕ is in $O(n \cdot l)$.*

Proof: The proof closely follows that of Proposition 23, additionally taking into account the fact that the space $D(L_n)$ consists of only one chain of length $n-1$. □

In this case, the translation to clause form is at most quadratic in the number of truth values (it is linear in the number of truth values for those formulae that do not contain the connective $\Rightarrow$). The rules for translating formulae containing $\Rightarrow$ are similar to the rules that can be obtained with the optimization method of Salzer [Sal96, Sal99]. The method for automated theorem proving presented in Section 5 reduces in this case to the automated theorem proving method for regular clauses presented in [Häh96b].

6.3 P_{mn}-logics

P_{mn}-logics (where $m > n \geq 0$) are propositional logics which are sound and complete with respect to the variety $\mathsf{P_{mn}}$ of P_{mn}-algebras. P_{mn}-algebras are distributive lattices with 0 and 1, and with a unary operation f satisfying $f(0) = 1$, $f(1) = 0$, the De Morgan laws with respect to $\vee$ and $\wedge$, and the additional conditions:

$$f^m(x) = f^n(x) \qquad \text{for every } x \in L \qquad \text{if } m-n \text{ is even;}$$

$$\left\{\begin{array}{ll} f^m(x) \vee f^n(x) = 1 & \text{for every } x \in L \\ f^m(x) \wedge f^n(x) = 0 & \text{for every } x \in L \end{array}\right. \qquad \text{if } m-n \text{ is odd.}$$

For the complete definitions we refer to [Urq79, Gol81].

Urquhart [Urq79] showed that the Priestley duality induces a dual equivalence between the category of P_{mn}-algebras and a corresponding category of Priestley spaces endowed with an operation g that satisfies the identity $g^m = g^n$. It is also shown that the variety $\mathsf{P_{mn}}$ is generated by one algebra, L_{mn}; its dual S_{mn} will be described in what follows. For $m > n \geq 0$, let S_{mn} be the structure $(X, \leq, \tau, g)$ defined by setting $X = \{0, 1, \ldots, m-1\}$, with the discrete topology and order, and with $g(k) = k+1$ for $k < m-1$ and $g(m-1) = n$, as shown in Figure 5. Let $L_{mn} = (\mathcal{P}(X), \cup, \cap, \emptyset, X, f)$ (the lattice of all subsets of X, with $f : \mathcal{P}(X) \to \mathcal{P}(X)$ defined by $f(Y) = X \backslash g^{-1}(Y)$). The variety $\mathsf{P_{mn}}$ is generated by L_{mn}, hence P_{mn}-logics can be seen as many-valued logics having L_{mn} as algebra of truth values (first-order P_{mn}-logics can be defined as explained in Section 4.2). Note that L_{mn} has 2^m elements, whereas its dual, S_{mn}, has only m elements.

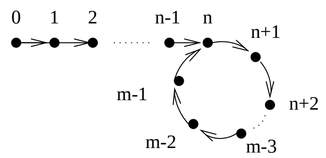

Figure 5: S_{mn}

Example. The P_{20}-logic is the fragment of relevance logic $\mathsf{E_{fed}}$ considered by Anderson and Belnap [AB75]. This logic is sound and complete with respect to the variety P_{20} of *De Morgan algebras*. The variety of De Morgan algebras is generated by $L_{20} = (\{0, 1, a, b\}, 0, 1, \vee, \wedge, f)$, where 0 is the first element, 1 the last element, a, b are incomparable and $f(0) = 1, f(1) = 0, f(a) = a, f(b) = b$. The dual S_{20} of L_{20} has only two incomparable elements, $\uparrow a, \uparrow b$, and $g(\uparrow a) = \uparrow b, g(\uparrow b) = \uparrow a$.

Theorem Proving. Both the translation to clause form and the general automated proving procedure presented in Section 5 can also be applied in this case. The complexity (expressed by the number of clauses generated starting from a formula ϕ) is, by Proposition 15 (1), $O(l \cdot m)$, where l is the number of all non-atomic subformulae of ϕ. The advantage of using the dual of the algebra of truth values instead of the algebra itself are quite clear in this case.

7 Related approaches

The work towards finding a procedure for automated theorem proving for this class of logics was inspired by the papers of Baaz and Fermüller [Baa92, BF92, BF95] and Hähnle [Häh94] on resolution-based theorem proving in finite-valued quantificational first-order logics on the one hand, and by the existing representation theorems for distributive lattices with operators [Pri70, Gol89, SS00a] on the other hand.

The goal of our approach was to exploit the finer structure of the algebra of truth values of a given logic, and, when this structure allows it, to improve the efficiency of the automated theorem proving procedure both by reducing the number of clauses in the clause form of a formula, and by refining the resolution procedure. In what follows we compare the method described in this paper with other approaches to clause generation and automated theorem proving in many-valued logics.

Many-valued resolution. The theoretical result of Baaz and Fermüller establishes a general method for resolution-based theorem proving in finite-valued quantificational first-order logics, and at the same time it presents the resolution procedure in a new light. In [BF95] it is shown that, applying structure-preserving rules, the clause form of a formula with k occurrences of at most r-ary operators and m occurrences of quantifiers based on a finite set A of truth values contains no more than $k|A|^r + m2^{|A|} + 1$ clauses, if optimal translation rules are used (where A is the set of truth values). Thus, for a formula with k occurrences of at most r-ary operators and no quantifiers, the clause form contains no more than $k|A|^r$ clauses. Improvements may be achieved by using sets as signs, as in [Häh90, Häh91, Häh93, BFS99] and, additionally, by special minimization techniques, as done for instance by Salzer in [Sal96, Sal99]. The method of Salzer starts with an arbitrary conjunctive normal form (e.g. constructed from the truth table, as in [BF95]), saturates it under sets-as-signs resolution, and selects a minimal subset of the saturated set of clauses by any covering-set-algorithm. The results in [Sal96, Sal99] show that the conjunctive normal forms obtained this way are optimal. However, the method does not bring much information about the general form of the conjunctive normal forms, or on the kind of signs actually needed in such conjunctive normal forms for whole classes of logics.

An example of logics in which the knowledge about the kind of signs actually needed in a signed clause form leads to an increase in both clarity and efficiency are the so-called *regular logics*. In regular logics all sentences can be naturally expressed in a signed clause form, where all clauses are *regular*, i.e. have all signs of the form $\boxed{\geq j} := \{j, \dots, \top\}$ or $\boxed{\leq j} := \{\bot, \dots, j\}$. Automated theorem proving in such logics has been studied by Hähnle [Häh94].

Comparison. One important advantage of the method presented in this paper is that only elements of the Priestley dual of the set of truth values are used as signs, thus fewer signs have to be taken into account in the process of translation to clause form. The results presented here have similarities with the work of Hähnle on regular logics. Expressions of the type $\boxed{\alpha}\ \phi^t$ respectively $\boxed{\alpha}\ \phi^f$ used by us correspond to the "positive and negative regular formulae" of the form $\boxed{\geq i}\ \phi$ respectively $\boxed{\leq i}\ \phi$ used for regular logics. In the particular case of totally ordered lattices of truth values, Hähnle's notions of positive and negative literal are recovered: $\boxed{\geq i}\ \phi$ corresponds to $\boxed{\alpha}\ \phi^t$ and $\boxed{\leq i-1}\ \phi$ to $\boxed{\alpha}\ \phi^f$, where $\alpha = \ \uparrow i$. The signed hyperresolution method given by us is thus a generalization of this signed hyperresolution method for regular logics due to Hähnle to the case of sets of truth values which are not linearly ordered.

Our method is efficient especially in the cases when the set of truth values is not linearly ordered, i.e. when the algebra of truth values has considerably more elements than its dual space, which is the case with SH_n-logics and P_{mn}-logics discussed in Sections 6.1 and 6.3. In the case of SHK_n-logics, considered in Section 6.2, the method described here is essentially the same as the method using "regular signs" developed in [Häh94]. Moreover, for linearly-ordered sets of truth values (such as in the case of SHK_n-logics) the set of clauses introduced by the Heyting implication (determined by a technique similar to that used in Proposition 22) is the same as that obtained with Salzer's algorithm [Sal96, Sal99], i.e. linear in the number of truth values.

Proposition 15 shows that, as expected, our method generates fewer clauses than the very general method described in [BF95] for those formulae that only contain the following logical connectives: conjunction, disjunction, and unary operators in $Lh \cup La \cup Jh \cup Mh \cup Ja \cup Ma$. Further properties of the logics under consideration and the structure of the dual space of the algebra of truth values lead to better performances also for formulae containing Heyting implication and negation in certain cases (such as SH_n-logics). A comparison with the optimal conjunctive normal forms obtained by using the minimization techniques in [Sal99] is more difficult, since there general classes of sets of signs can (and have to) be considered.

Another advantage of our method is that the description of the induced maps (resp. relations) on $D(A)$ is often much more compact than giving the truth tables of the logical operators.

Other related approaches. We would like to mention some other approaches that exploit the finer properties of the algebra of truth values:

Algorithmic and Post logics. Orłowska was the first (to our knowledge) to introduce the notion of "resolution-interpretability of a logic in another logic". She applied it for constructing theorem proving systems for algorithmic and m-valued Post logics [Orł78, Orł79, Orł80]. We now refer to the resolution methods for m-valued Post logics (cf. also the work of Morgan on this topic [Mor76]): the clauses have the form $\sigma_1(p_1) \vee \cdots \vee \sigma_k(p_k)$, where $\sigma_1, \ldots . \sigma_k$ are certain unary connectives. As pointed out in [Häh94], one can write the clauses, equivalently, $S_1 : p_1 \vee \cdots \vee S_n : p_n$, where $S_i = \sigma_i^{-1}(1)$. (Note that if σ is a lattice morphism into $\{0, 1\}$ then $\sigma^{-1}(1)$ is a prime filter.)

Operators and quantifiers based on semi-lattices. Salzer [Sal96] investigates the problem of finding optimal axiomatizations for operators and distribution quantifiers in finitely-valued first-order logics. He shows that in the case of operators and quantifiers based on semi-lattices, rules with a minimal branching degree can be obtained by instantiating a schema, which can also be used for optimal tableaux with sets-as-signs.

Distribution quantifiers based on finite distributive lattices. In [Häh98], Hähnle provides tools for distribution quantifiers based on finite distributive lattices (propositional connectives are not taken into account). Hähnle's work provides a link between skolemized signed formulas and filters/ideals in Boolean set lattices, and tableau-style axiomatizations of distribution quantifiers are derived using lattice-theoretic tools such as Birkhoff's representation theorem for finite distributive lattices. The signed normal forms obtained by applying this method contain as signs both order-ideals and order-filters.

In our approach we explicitly requested that the many-valued logics considered are based on distributive lattices, and we obtained a uniform way of expressing signed clauses, the signs being pairs consisting of an prime filter and a truth value (t (true) and f (false)). As a side-effect this also shows that, for the class of logics we consider, only certain sets-as-signs are really needed in a signed clause form.

Paraconsistent and annotated logics. Approaches to automated proving in paraconsistent [KL92] or to annotated logics [LHSdC91, Mes95] also have certain similarities with our own approach, especially concerning the form of literals. Moreover, although a version of signed resolution has been proved correct for paraconsistent logics [KL92, LHSdC91], we are not aware of any work using a version of signed hyperresolution, as presented in this paper.

$\mho$-resolution. We also mention the work of Leach, Lu, Murray and Rosenthal [LMR98], who extended the signed resolution procedure of [KL92] to a so-called $\mho$-resolution procedure for certain classes of dually relatively-pseudo-complemented lattices [LLMR98]. In their approach the signs are again of the form $\uparrow i$ and $\neg\uparrow i$, where i is an element in the set of truth values.

The results in our paper show that actually only those signs corresponding to the join-irreducible elements are really needed in a signed clause form for such logics. This follows from the fact that the existence of a dual relative pseudocomplement for every element in the lattice of truth values implies the distributivity of the lattice of truth values.

Modal logics. Finally, there seem to exist similarities with methods for automated theorem proving in modal logics (cf. e.g. [Ohl93]).

8 Conclusions

Our goal was to exploit the algebraic properties of the algebra of truth values of a given logic, and, when these allow it, to improve the efficiency of automated theorem proving in such logics. We showed that for some classes of many-valued logics based on distributive lattices with operators both a procedure for translation to a signed clause from using structure-preserving rules, and a method for proving non-satisfiability of such sets of signed clauses can be given.

In the particular cases in which it can be applied, our method leads to fewer clauses than the general method if the connectives corresponding to the operators (besides meet and join) are unary. Some problems have still to be solved in order to make it more efficient in dealing with the Heyting implication and with join and meet hemi(anti)morphisms with arity greater than one. It seems however that most of these problems can be solved by introducing additional quantification over the elements in the dual space of the lattice.

Implementation. We implemented the automated theorem proving by signed negative hyperresolution resolution in non-classical logics described in Section 5 in SICStus Prolog. The implementation is described in detail in [SS97]. Since in [SS98a] we noticed that the translation to clause form for the class of many-valued logics described in this paper is actually a translation to classical logic, as mentioned in Section 5.2.3, we also used the set of signed clauses (modified accordingly) as input to theorem provers in classical logic such as Saturate [GNN94] and SPASS [WAB$^+$99]. This last solution, relying on the translation to classical logic, appears to be more efficient. This happens because there exist many relatively powerful theorem provers for classical logic, with which, for instance, our theorem prover for signed hyperresolution described in [SS97] cannot (for the moment) compete. Similar ideas can be also used for general many-valued logics as well as for regular logics: in ongoing work together with Harald Ganzinger we use techniques from classical logic such as superposition and ordered chaining for automated theorem proving in many-valued logics.

Future work. Analyzing the proofs of the results presented so far, it seems that similar ideas can be also used for more general logics, which are not necessarily finitely-valued. Some results in this direction are presented in [SS99b]. We would like to investigate the degree of generality of this approach. Priestley duality helped us to better understand the link between algebraic and Kripke models for certain classes of logics, and the natural way of defining truth and satisfiability in Kripke models, starting from the notions of truth and satisfiability in algebraic models (cf. also [SS00a, SS00b]). This link between algebraic and Kripke-style models induced by the Priestley duality (in the case of distributive lattices with operators) seems to be a possible starting point towards a systematic way of translating formulae of certain non-classical logics (not necessarily finitely-valued) into classical two-valued logics. We would like to see if similar results can be also obtained with more general representation theorems, e.g. for non-distributive lattices.

In future work we would also like to investigate (when possible) whether methods similar to the ones used in modal logics would yield decision procedures for some more general logics, e.g. based on distributive lattices with operators.

Acknowledgements. Some of the results presented in this paper are contained in my PhD thesis [SS97]. This work was partially supported by MEDLAR II (ESPRIT Basic Research Project 6471; financed for Austria by FWF) at RISC-Linz, and partially by a postdoctoral fellowship at MPII-Saarbrücken. I also gratefully acknowledge the financial support from COST Action 15 MVL. Many thanks to M. Baaz, H. Ganzinger, R. Hähnle, L. Iturrioz, H. de Nivelle, H.J. Ohlbach, E. Orłowska, J. Pfalzgraf, H. Priestley, G. Salzer and K. Stokkermans for their papers and for stimulating discussions. I also thank the referees for their helpful comments.

References

[AB70] R. Anderson and W. Bledsoe. A linear format for resolution with merging and a new technique for establishing completeness. *Journal of the ACM*, 17:525–534, 1970.

[AB75] A.R. Anderson and N.D. Belnap. *Entailment – The Logic of Relevance and Necessity.* Princeton University Press, 1975.

[Baa92] M. Baaz. Automatisches Beweisen für endlichwertige Logiken. *Mitt. Math. Ges. Hamburg*, 12:1141–1155, 1992.

[BF92] M. Baaz and C.G. Fermüller. Resolution for many-valued logics. In A. Voronkov, editor, *Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR'92)*, LNAI 624, pages 107–118, St. Petersburg, Russia, July 1992. Springer Verlag.

[BF95] M. Baaz and C.G. Fermüller. Resolution-based theorem proving for many-valued logics. *Journal of Symbolic Computation*, 19:353–391, 1995.

[BFS99] M. Baaz, C.G. Fermüller, and G. Salzer. Automated deduction for many-valued logics. In A. Robinson and A. Voronkow, editors, *Handbook of Automated Reasoning*. Elsevier Science, to appear 1999.

[BS81] S. Burris and H.P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics. Springer Verlag, 1981.

[Car87] W.A. Carnielli. Systematization of finite many-valued logics through the method of tableaux. *Journal of Symbolic Logic*, 52(2):473–493, 1987.

[Car91] W.A. Carnielli. On sequents and tableaux for many-valued logics. *Journal of Non-Classical Logics*, 8(1):59–76, 1991.

[CL73] C. Chang and R.C. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

[DP90] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[Ede92] E. Eder. *Relative Complexities of First Order Calculi*. Wiesbaden: Vieweg, Braunschweig, 1992.

[FLTZ93] C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution Methods for the Decision Problem*. LNCS 679. Springer-Verlag, 1993.

[GNN94] H. Ganzinger, R. Nieuwenhuis, and P. Nivela. The Saturate system. Available on the world-wide web under url: `http://www.mpi-sb.mpg.de/SATURATE/Saturate.html`, 1994.

[Gol81] M. Goldberg. Distributive Ockham algebras: Free algebras and injectivity. *Bull. Austral. Math. Soc.*, 24:161–203, 1981.

[Gol89] R. Goldblatt. Varieties of complex algebras. *Annals of Pure and Applied Logic*, 44(3):153–301, 1989.

[Häh90] R. Hähnle. Towards and efficient tableau proving procedure for multiple-valued logics. In *Proceedings of Workshop on Computer Science Logic, Heidelberg*, LNCS 533, pages 248–260. Springer Verlag, Berlin, 1990.

[Häh91] R. Hähnle. Uniform notation of tableaux rules for multiple-valued logics. In *Proceedings of International Symposium of Multiple-valued Logics, Victoria*, pages 238–245, Los Alamitos. CA., 1991. IEEE Press.

[Häh93] R. Hähnle. *Automated Theorem Proving in Multiple-Valued Logics*, volume 10 of *International Series of Monographs on Computer Science*. Oxford University Press, 1993.

[Häh94] R. Hähnle. Short conjunctive normal forms in finitely valued logics. *Journal of Logic and Computation*, 4(6):905–927, 1994.

[Häh96a] R. Hähnle. Commodious axiomatization of quantifiers in multiple-valued logic. In *Proc. 26th International Symposium on Multiple-Valued Logics, Santiago de Compostela, Spain*, pages 118–123. IEEE Press, Los Alamitos, May 1996.

[Häh96b] R. Hähnle. Exploiting data dependencies in many-valued logics. *Journal of Applied Non-Classical Logics*, 6(1):49–69, 1996.

[Häh98] R. Hähnle. Commodious axiomatization of quantifiers in multiple-valued logic. *Studia Logica*, 61(1):101–121, 1998. Special Issue on Many-Valued Logics, their Proof Theory and Algebras.

[IO96] L. Iturrioz and E. Orłowska. A Kripke-style and relational semantics for logics based on Łukasiewicz algebras. Conference in honour of J. Łukasiewicz, Dublin, 1996.

[Itu82] L. Iturrioz. Modal operators on symmetrical Heyting algebras. In T. Traczyk, editor, *Universal Algebra and Applications*, Banach Center Publications, Vol.9, pages 289–303. PWN-Polish Scientific Publishers, 1982.

[Itu83] L. Iturrioz. Symmetrical Heyting algebras with operators. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 29:33–70, 1983.

[Itu95] L. Iturrioz. Symmetric Heyting algebras with a finite order type of operators. *Studia Logica*, 55:89–98, 1995.

[KL92] M. Kifer and M. Lozinskii. A logic for reasoning with inconsistency. *Journal of Automated Reasoning*, 9:179–215, 1992.

[LHSdC91] J.J. Lu, L.J. Henschen, V.S. Subrahmanian, and N.C.A. da Costa. Reasoning in paraconsistent logics. In R. Boyer, editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe*, pages 181–210. Kluwer, 1991.

[LLMR98] S.M. Leach, J. Lu, N.V. Murray, and E Rosenthal. $\mho$-resolution: An inference rule for regular multiple-valued logics. In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Proceedings of the European Workshop JELIA'98, Dagstuhl, Germany*, LNAI 1489, pages 154–168, 1998.

[LMR93] J. Lu, N.V. Murray, and E. Rosenthal. Signed formulae and annotated logics. In *Proc. 23rd International Symposium on Multiple-Valued Logics*, pages 48–53. IEEE Computer Society Press, Los Alamitos, 1993.

[LMR98] J. Lu, N.V. Murray, and E. Rosenthal. A framework for reasoning in multiple-valued logics. *Journal of Automated Reasoning*, 21(1):39–67, 1998.

[Mes95] B. Messing. Knowledge representation in many-valued Horn clauses. In *Proceedings of the 6-th Conference of the Spanish Association for Artificial Intelligence, Alicante*, pages 83–92. Asociatión Española para la Inteligencia Artificial, AEPIA, 1995.

[Mor76] C.G. Morgan. A resolution principle for a class of many-valued logics. *Logique et Analyse*, 19(74-76):311–339, 1976.

[Mos48] A. Mostowski. Proofs of non-deducibility in intuitionistic functional calculus. *The Journal of Symbolic Logic*, 13:204–207, 1948.

[Nou97] A. Nour. *Etude de systèmes logiques extensions de la logique intuitionniste (in French).* PhD thesis, Université Joseph Fourier, Grenoble 1, 1997.

[Ohl93] H.J. Ohlbach. Translation methods for non-classical logics: An overview. *Bulletin of the IGPL*, 1(1):69–89, 1993.

[Orł78] E. Orłowska. The resolution principle for ω^+-valued logic. *Fundamenta Informaticae*, 2:1–15, 1978.

[Orł79] E. Orłowska. Resolution systems and their applications I. *Fundamenta Informaticae*, 3:235–268, 1979.

[Orł80] E. Orłowska. Resolution systems and their applications II. *Fundamenta Informaticae*, 3:333–362, 1980.

[Pri70] H.A. Priestley. Representation of distributive lattices by means of ordered Stone spaces. *Bull. London Math Soc.*, 2:186–190, 1970.

[Pri72] H.A. Priestley. Ordered topological spaces and the representation of distributive lattices. *Proc. London Math. Soc.*, 3:507–530, 1972.

[Pri84] H.A. Priestley. Ordered sets and duality for distributive lattices. *Annals of Discrete Mathematics*, 23:39–60, 1984.

[Ras88] H. Rasiowa. Logic of approximation reasoning. In *Proc. 1st Workshop on Computer Science Logic, Karlsruhe, 1987*, LNCS 329. Springer Verlag, Berlin, 1988.

[Sal96] G. Salzer. Optimal axiomatizations for multiple-valued operators and quantifiers based on semilattices. In M. McRobbie and J. Slaney, editors, *Proc. 13th Conference on Automated Deduction, New Brunswick/NJ, USA*, LNCS 1104, pages 688–702. Springer Verlag, 1996.

[Sal99] G. Salzer. Optimal axiomatizations of finitely-valued logics. Information and Computation, To appear, 1999.

[Sco73] D.S. Scott. Background to formalization. In H. Leblanc, editor, *Truth, Syntax and Modality. Proceedings of the Temple University Conference on Alternative Semantics*, Studies in Logic and the Foundations of Mathematics, Vol. 68, pages 244–273, 1973.

[SS97] V. Sofronie-Stokkermans. *Fibered Structures and Applications to Automated Theorem Proving in Certain Classes of Finitely-Valued Logics and to Modeling Interacting Systems.* PhD thesis, RISC-Linz, J.Kepler University Linz, Austria, 1997. Available on the world-wide web under url: `http://www.mpi-sb.mpg.de/~sofronie/thesis.ps.gz`.

[SS98a] V. Sofronie-Stokkermans. On translation of finitely-valued logics to classical first-order logic. In H. Prade, editor, *Proceedings of ECAI 98*, pages 410–411. John Wiley & Sons, 1998.

[SS98b] V. Sofronie-Stokkermans. Resolution-based theorem proving for SH_n-logics. Proceedings of FTP'98, Vienna, November 23-25, 1998; Technical Report E1852-GS-981, Technische Univ. Wien., November 1998. Available on the world-wide web under url: `http://www.logic.at/FTP98/`.

[SS99a] V. Sofronie-Stokkermans. Priestley duality for SHn-algebras and applications to the study of Kripke-style models for SHn-logics. Multiple-Valued Logic – An International Journal, 1999. To appear. Preliminary version available at: `http://www.mpi-sb.mpg.de/~sofronie/dual-shn.ps.gz`.

[SS99b] V. Sofronie-Stokkermans. Representation theorems and automated theorem proving in non-classical logics. In *Proceedings of the 29th IEEE International Symposium on Multiple-Valued Logic (ISMVL 99)*, pages 242–247, Freiburg im Breisgau, Germany, May 1999. IEEE Computer Society Press.

[SS00a] V. Sofronie-Stokkermans. Duality and canonical extensions of bounded distributive lattices with operators, and applications to the semantics of non-classical logics I. *Studia Logica*, 64, 2000. To appear. Preliminary version available under url: `http://www.mpi-sb.mpg.de/~sofronie/relational-models-general1.ps.gz`.

[SS00b] V. Sofronie-Stokkermans. Duality and canonical extensions of bounded distributive lattices with operators, and applications to the semantics of non-classical logics II. *Studia Logica*, 64, 2000. To appear. Preliminary version available under url: `http://www.mpi-sb.mpg.de/~sofronie/relational-models-general2.ps.gz`.

[Suc74] W. Suchoń. La méthode de Smullyan de construire le calcul n-valent de Łukasiewicz avec implication and négation. *Reports on Mathematical Logic, Universities Cracow and Katowicze*, pages 37–42, 1974.

[Sur84] S.J. Surma. An algorithm for axiomatizing every finite logic. In E. Lusk and R. Overbeek, editors, *Computer Science and Multiple-Valued Logics*, pages 143–149. North-Holland, Amsterdam, 1984.

[Tse70] G. S. Tseitin. On the complexity of derivation in propositional calculus. In *Seminars in Mathematics V.A. Steklov Math. Institute, Leningrad*, volume 8, pages 115–125. Consultants Bureau, New York-London, 1970. Reprinted in J. Siekmann and G. Wrightson (eds): Automation of Reasoning, Vol. 2, 1983, Springer Verlag, pp. 466–483.

[Urq79] A. Urquhart. Distributive lattices with a dual homomorphic operation. *Studia Logica*, 38(2):201–209, 1979.

[WAB+99] C. Weidenbach, B. Afshordel, U. Brahm, C. Cohrs, T. Engel, E. Keen, C. Theobalt, and D. Topic. System description: SPASS version 1.0.0. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 314–318, Trento, Italy, 1999. Springer.