Automated theorem proving by resolution in non-classical logics

Viorica Sofronie-Stokkermans

Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, Saarbrücken, Germany

Abstract

This paper is an overview of a variety of results, all centered around a common theme, namely embedding of non-classical logics into first order logic and resolution theorem proving. We present several classes of non-classical logics, many of which are of great practical relevance in knowledge representation, which can be translated into tractable and relatively simple fragments of classical logic. In this context, we show that refinements of resolution can often be used successfully for automated theorem proving, and in many interesting cases yield optimal decision procedures.

1 Introduction

During the last years, a large number of non-classical logics were studied. In particular, various methods for automated theorem proving in such logics have been proposed: sequent calculi, various kinds of analytic tableaux, and various types of resolution-like calculi. Most of these methods are strongly related to the particular characteristics of the logics. It usually is non-trivial to give efficient implementations for these specialized calculi, which makes it difficult to maintain, scale, and modify such provers in order to ensure that they develop at the same speed as performant theorem provers for classical logic. Therefore, it is very desirable to find uniform principles, applicable to large classes of logics, which lead to simple and reusable implementations.

Identifying such a unifying principle is the main goal of the present paper. We present several situations in which non-classical logics can be translated into tractable and simple fragments of classical logic, and resolution can be used successfully for automated theorem proving. The main advantage of such an approach is that it allows us to use existing automated theorem provers

Email address: `sofronie@mpi-sb.mpg.de` (Viorica Sofronie-Stokkermans).

for free, without the need of any sophisticated encodings. In this paper we will show that in many interesting and quite general situations translations to classical logic allow us to obtain decision procedures of optimal complexity.

The paper starts with a presentation of various non-classical logics, such as many-valued logics (with finite or infinite set of truth values), modal logics, intuitionistic logics, relevant logics, and description logics. The presentation focuses on the algebraic semantics: either in terms of one fixed finite algebra, in the case of finitely-valued logics, or in terms of an arbitrary algebra, in the case of more general many-valued logics; or in terms of classes of algebras. It is known that checking validity of formulae in non-classical logics having an algebraic semantics often can be reduced to checking whether corresponding word problems hold in the class of algebraic models. We show that similar phenomena also occur in some specific description logics: checking subsumption with respect to TBoxes can often be reduced to checking whether suitably defined uniform word problems hold in classes of Boolean algebras, distributive lattices or semilattices with operators. We therefore consider more general problems in universal algebra, such as word problems and uniform word problems with respect to specific classes of algebras, generally classes of (distributive) lattices with operators.

In the second part of the paper we describe methods for automated theorem proving for the logics in the classes considered above. We focus on methods based on translations to classical logic. We first show that various versions of many-valued resolution for finitely-valued logics can be reconstructed by using general saturation-based techniques for first order theories of transitive relations [GS00]. We then consider other non-classical logics which are not finitely valued, but for which nevertheless such embeddings into classical logics are possible. We present, for instance, a translation to clause form for prenex first order Gödel logics [BFC01] which allows to use saturation-based techniques for dense total orderings, and then focus on propositional logics based on distributive lattices with operators (possibly many-sorted). We show that resolution-based decision procedures with optimal complexity can be obtained in many cases by using refinements of resolution such as ordered resolution with selection, or ordered chaining with selection.

The paper is an overview of a variety of results, all centered around a common theme, namely embedding of non-classical logics into first order logic and resolution theorem proving. Some of the results in Section 3.3.1 – in particular the proof given there for the PTIME complexity of the description logic $\mathcal{EL}$ – are, to the best of our knowledge, new. We do not present or discuss implementation issues, nor technicalities about the automated reasoning part. Also experimental results and comparisons are out of the scope of the present paper due to the wide range of logics which we present here.

2 Preliminaries

For basic notions of universal algebra we refer e.g. to [BS81]. We also assume known standard notions, such as partially-ordered set, (bounded) lattice, and distributive lattice, as well as (prime) filters in lattices. For definitions and more details we refer to [DP90].

Let Σ be a signature and $a : \Sigma \to \mathbb{N}$ an arity function. A Σ-algebra is a structure $\mathbf{A} = (A, \{\sigma_A\}_{\sigma\in\Sigma})$, where A is a non-empty set and for every $\sigma \in \Sigma$, $\sigma_A : A^{a(\sigma)} \to A$. Given a set X, the term algebra over Σ in the variables X will be denoted $T_\Sigma(X)$. An *equation* is an expression of the form $t_1 = t_2$ where $t_1, t_2 \in T_\Sigma(X)$; an *implication* is an expression of the form $\beta_1 \wedge \cdots \wedge \beta_m \to \alpha$, where $\beta_1, \ldots, \beta_m, \alpha$ are equations. A *conditional equation* (or *quasi-equation*) is an expression which is either an equation or an implication. A Σ-algebra $\mathbf{A} = (A, \{\sigma_A\}_{\sigma\in\Sigma})$ *satisfies* an equation $t_1 = t_2$ (notation: $\mathbf{A} \models t_1 = t_2$) if t_1 and t_2 become equal for every substitution of elements in A for the variables. $\mathbf{A}$ satisfies an implication $\gamma := (t_1 = t'_1 \wedge \cdots \wedge t_m = t'_m) \to t = t'$ (notation: $\mathbf{A} \models \gamma$) if for every substitution v of elements in A for the variables in γ such that $v(t_i) = v(t'_i)$ for all $i = 1, \ldots, m$, $v(t) = v(t')$.

A class $\mathcal{K}$ of algebras satisfies an equation or implication γ (notation: $\mathcal{K} \models \gamma$) if every algebra $\mathbf{A}$ of $\mathcal{K}$ satisfies γ. An equational class (or variety) is the class of all algebras that satisfy a set of equations. A quasi-variety is the class of all algebras that satisfy a class of quasi-equations.

The *word problem* for a class $\mathcal{K}$ of Σ-algebras is the problem of deciding, for any two Σ-terms t_1, t_2 whether $\mathcal{K} \models t_1 = t_2$. The *uniform word problem* for a class $\mathcal{K}$ of Σ-algebras is the problem of deciding, for any implication of the form $(t_1 = t'_1 \wedge \cdots \wedge t_m = t'_m) \to t = t'$, whether $\mathcal{K} \models (t_1 = t'_1 \wedge \cdots \wedge t_m = t'_m) \to t = t'$.

3 Non-classical logics

This section contains general information about non-classical logics. We start with a brief presentation of first order many-valued logics (with finite or infinite set of truth values). We give several examples, and mention various decidability and complexity results. We then consider classes of (propositional) logics best described in terms of classes of algebras, such as modal logics, intuitionistic logics, relevant logics, and description logics.

The presentation focuses on the algebraic semantics: either in terms of one fixed finite algebra, in the case of finitely-valued logics, or in terms of an arbitrary algebra, in the case of more general many-valued logics; or in terms of classes of algebras. We emphasize the fact that checking validity of formulae (or various subsumption problems) can be reduced to checking (uniform) word problems for classes of Boolean algebras, distributive lattices or semilattices with operators.

3.1 Many-valued logics

Let $\mathbf{L} = (X, O, P, \Sigma, \{Q_1, \ldots, Q_k\})$ 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 $Q_1, \ldots, Q_k$. Terms, ground terms, atomic formulae and formulae are defined in the usual way. Let A be a 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 $\overline{Q} : \mathcal{P}(A) \backslash \{\emptyset\} \to A$.

A many-valued logic with language $\mathbf{L}$ and set of truth values A is a pair $\mathcal{L} = (\mathbf{L}, \mathcal{A})$ consisting of a first order language $\mathbf{L} = (X,\ O,\ P,\ \Sigma,\ \{Q_1,\ \ldots,\ Q_k\})$ and a set of truth values endowed with truth functions for all logical operators and quantifiers in $\mathbf{L}$, $\mathcal{A} = (A, \{\sigma_A\}_{\sigma \in \Sigma}, \{\overline{Q_i}\}_{i=1,\ldots,k})$. Many-valued logics with a finite set of truth values are called finitely valued logics, those with an infinite set of truth values are called infinitely valued logics.

Definition 3.1 *A* frame *for $\mathcal{L} = (\mathbf{L}, \mathcal{A})$ is a pair (D, I) where D is a non-empty set, the* domain*, and 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, $v_{\mathcal{I}} : \mathsf{Fma}(\mathcal{L}) \to A$, as follows:

- $v_{\mathcal{I}}(R(t_1, \ldots, t_n)) = I(R)(v_{\mathcal{I}}(t_1), \ldots, v_{\mathcal{I}}(t_n))$ for all n-ary $R \in P$, $n \geq 0$,
- $v_{\mathcal{I}}(\sigma(\phi_1, \ldots, \phi_n)) = \sigma_A(v_{\mathcal{I}}(\phi_1), \ldots, v_{\mathcal{I}}(\phi_n))$ for all n-ary $\sigma \in \Sigma$,
- $v_{\mathcal{I}}((Qx)\phi) = \overline{Q}(\{w \mid \exists d \in D \text{ s.t. } v_{\mathcal{I}_{x,d}}(\phi) = w\})$ 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 $v_{\mathcal{I}}(\phi) \in A_d$. For detailed introductions to many-valued logics we refer to [Urq86, BF95, BFS99, Häh01].

Examples of finitely-valued logics: The simplest example is classical logic (the set of truth values is $\{0, 1\}$, where 0 stands for "false" and 1 for "true"). Several three-valued logics have been introduced in the attempt of modeling possibility, meaningless statements, or undefinedness. In all cases, the set of truth values is the three element set $\{0, \frac{1}{2}, 1\}$. The logical connectives are defined according to the meaning of the intermediate value $\frac{1}{2}$ which needs to be expressed such as, for instance: "possible" in Łukasiewicz's 3-valued logic $Ł_3$; "meaningless" in Bochvar's 3-valued logic, or "undefined" in Kleene's logic. Several generalizations to finitely many degrees of truth exist. For instance, the Łukasiewicz logic of order n, $Ł_n$, has as set of truth values the n-element chain $\{0, \frac{1}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$, a totally ordered set, and connectives $\circ_{Ł}$ and $\rightarrow_{Ł}$, where $x \circ_{Ł} y = \max(0, x+y-1)$ and $x \rightarrow_{Ł} y = \min(1, 1-x+y)$. Logics with a set of truth values which is not linearly ordered (but usually lattice ordered) have been also defined. One example is the 4-valued Belnap logic (which has been used for reasoning about inconsistent databases), where the algebra of truth values is the product of the 2-element chain with itself. Another example is the class of SH_n-logics introduced by Iturrioz [Itu83, IO96], which have as algebra of operators the product of the n-element chain with itself, and with additional operators.

Examples of infinitely-valued logics: Typical examples of infinitely-valued logics are the so-called fuzzy logics. Fuzzy logics are many-valued logics having the interval $[0, 1]$ as set of truth values; premise combination $\circ$ is modeled by t-norms [1]. A binary operation $\rightarrow$ is a (right) residuation of $\circ$ if, for every $x, y, z \in [0, 1]$, $y \circ x \leq z$ if and only if $x \leq y \rightarrow z$. Every continuous t-norm $\circ$ on $[0, 1]$ has a unique right residuation $\rightarrow$. By choosing *the Gödel t-norm*, $x \circ y = \min(x, y)$; *the Łukasiewicz t-norm*, $x \circ y = \max(0, x + y - 1)$; or *the product t-norm*, $x \circ y = x \cdot y$ (product of reals), we can define the Gödel logic G_∞, the Łukasiewicz logic Ł, or the product logic L_Π, respectively.

For every $n \in \mathbb{N}$, n-valued variants $Ł_n$ and G_n of the propositional Łukasiewicz and Gödel logics, with set of truth values $\{0, \frac{1}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$, can be defined: premise combination $\circ$ is modeled by the Łukasiewicz t-norm and the Gödel t-norm respectively, and $\rightarrow$ is again the unique right residuation of $\circ$. (Product logic is only defined for the set of truth values $[0, 1]$, since $\{0, \frac{1}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$ is not closed under product.)

First order versions of the above-mentioned fuzzy logics can be obtained by defining the truth functions for quantifiers $Q_\forall = \mathsf{inf}$ and $Q_\exists = \mathsf{sup}$.

[1] A t-norm is a binary map $\circ : [0, 1] \rightarrow [0, 1]$ such that $([0, 1], \circ, 1)$ is a commutative semigroup with neutral element 1 and $\circ$ is monotone in both arguments.

Validity and satisfiability in propositional finitely-valued logics is obviously decidable. It is easy to see that satisfiability of formulae is in NP and validity is in co-NP. Obviously, first order many-valued logics are in general undecidable (since first order classical logic is undecidable). While the complexity of satisfiability and validity in propositional Gödel, Łukasiewicz, and product logic is the same as for two-valued logic, the situation is different in the first order case. The following results are well-known (for proofs we refer e.g. to [Mun87], [Háj98] and [Häh03]):

Theorem 3.2 [Mun87, Háj98, Häh03]

(1) Satisfiability is NP-complete and validity is co-NP-complete for the propositional Łukasiewicz logics $Ł_n$ and the propositional Gödel logics G_n; for the propositional Łukasiewicz logic Ł; for the propositional Gödel logic G_∞; and for the propositional product logic L_Π.
(2) Validity in the first order Gödel logic is Σ_1-complete, validity in the first order Łukasiewicz logic is Π_2-complete, and validity in the first order product logic is Π_2-hard.

3.2 Propositional non-classical logics

Many-valued logics are special logics, characterized by one given algebra of truth values, with a relatively simple structure. In general however, many non-classical logics are defined by describing the properties of premise combination and entailment by means of logical calculi (e.g. Gentzen-style calculi, Hilbert-style calculi, natural deduction systems). Logics defined this way usually have a natural algebraic model, namely their Lindenbaum algebra, which can be constructed by identifying provably equivalent formulae. The equivalence classes of the theorems can be regarded as *designated elements*. Thus, most of the known propositional non-classical logics can be regarded as many-valued logics with an infinite algebra of truth values, and a suitably defined set of designated elements. It is usually more convenient to identify *classes* of algebraic models for these logics, often classes of bounded lattices or semilattices with additional operators which are usually interpretations of logical connectives such as:

- the modal connectives for necessity ($\Box$) or possibility ($\Diamond$), or
- various types of negation ($\sim$) or
- various types of implication ($\rightarrow$).

The operators that correspond to these connectives often commute with part of the lattice structure, i.e. satisfy equations such as, for instance:

$$\Box(1) = 1,\ \Box(x \wedge y) = \Box(x) \wedge \Box(y), \tag{1}$$
$$\Diamond(0) = 0,\ \Diamond(x \vee y) = \Diamond(x) \vee \Diamond(y), \tag{2}$$
$$\sim 0 = 1,\ \sim(x \vee y) = \sim x \wedge \sim y, \tag{3}$$
$$\sim 1 = 0,\ \sim(x \wedge y) = \sim x \vee \sim y, \tag{4}$$
$$(0 \Rightarrow z) = 1,\ ((x \vee y) \Rightarrow z) = (x \Rightarrow z) \wedge (y \Rightarrow z), \tag{5}$$
$$(x \Rightarrow 1) = 1,\ (x \Rightarrow (y \wedge z)) = (x \Rightarrow y) \wedge (x \Rightarrow z). \tag{6}$$

Operators which, like $\Box$, preserve 1 and $\wedge$ are called *meet hemimorphisms*; operators like $\Diamond$, which preserve 0 and $\vee$ are called *join hemimorphisms*; operators like $\sim$ which "reverse" all the lattice structure are called *antimorphisms*. The operator $\Rightarrow$ above is a meet hemimorphism in the second argument, and a join antihemimorphism in the first argument.

Some non-classical logics which have as algebraic models lattices or semilattices with additional operators (in particular: Boolean algebras with operators, Heyting algebras with operators, or (distributive) lattices or semilattices with operators) are presented below. We first mention well-studied logics, such as modal logics, intuitionistic logic, fuzzy logics, relevant logics and other substructural logics. We then present in some detail some newer results related to TBox reasoning in description logics. The presentation focuses on the algebraic semantics. In particular we point out that:

- checking validity in non-classical logics having an algebraic semantics can often be reduced to checking whether corresponding word problems hold in the class of algebraic models [Ras74, ANS01, BRV01];
- checking subsumption with respect to TBoxes can often be reduced to checking whether suitably defined uniform word problems hold in classes of Boolean algebras, distributive lattices or semilattices with operators.

3.2.1 Logics based on classes of distributive lattices with operators

Most of the well-studied non-classical logics fall into this class. We mention some well known examples:

- *Modal logics* are in general sound and complete with respect to classes of Boolean algebras with operators $\mathbf{B} = (B, \vee, \wedge, \neg, 0, 1, \Box, \Diamond)$, where $\Diamond$ is a join hemimorphism, $\Box$ is a meet hemimorphism, and for every $x \in B$, $\Box x = \neg\Diamond\neg x$.
- *Intuitionistic logic* has as class of algebraic models the class of Heyting algebras. Various types of intuitionistic modal logics are sound and complete with respect to classes of Heyting algebras with operators.
- *Gödel's logic (or LC or Dummet's logic)* [Dum59] has as class of algebraic models the class of linear Heyting algebras (Heyting algebras satisfying $a \Rightarrow b \vee b \Rightarrow a = 1$).

Checking whether a formula ϕ is a theorem in such a logic can usually be reduced to checking whether $\mathcal{A} \models \phi = 1$, where $\mathcal{A}$ is a class of algebraic models of the logic.

- *Positive logics* (cf. also the so-called *binary logics* [Gol93] Ch.2, or the similar concept in [Dun95]) do not have the implication symbol as a logical connective. Their algebraic models are usually lattices with operators.

In positive logics, logical consequence can only be expressed by using the provability relation $\vdash$. Checking whether $\phi_1 \vdash \phi_2$ can usually be reduced to checking whether $\mathcal{A} \models \phi_1 \leq \phi_2$, where $\mathcal{A}$ is a class of algebraic models of the logic.

3.2.2 Logics based on residuated (semi)lattices

Residuated distributive lattices occur in a natural way as algebraic models for fuzzy, relevant and substructural logics.

Many *fuzzy logics* are sound and complete with respect to classes of residuated distributive lattices.

- The basic fuzzy logic (BL), for instance, has as algebraic models the class of all linearly ordered BL-algebras. BL-algebras [Háj98] are linearly ordered bounded lattices with two binary operators $\circ$ and $\rightarrow$, $(L, \vee, \wedge, 0, 1, \circ, \rightarrow)$, where $(L, \circ, 1)$ is a commutative semigroup with 1, $\circ$ is monotone in both arguments, and where for all $x, y, z \in L$,

$$x \circ z \leq y \text{ iff } z \leq (x \rightarrow y) \quad \text{ and } \quad x \wedge y = x \circ (x \rightarrow y).$$

- The *Gödel logic* has as algebraic models the class of all linearly ordered Heyting algebras. The *Łukasiewicz logics* [Łuk30] have as algebraic models the class of linearly ordered MV-algebras. MV-algebras are BL-algebras in which the identity $x = ((x \rightarrow 0) \rightarrow 0)$ holds.
- The *product logic* has as algebraic models the class of all (linearly ordered) product algebras, i.e. BL-algebras that satisfy

$$(z \rightarrow 0) \rightarrow 0 \leq ((x \circ z \rightarrow y \circ z) \rightarrow (x \rightarrow y)) \quad \text{ and } \quad x \cap (x \rightarrow 0) = 0.$$

The relevant logic RL introduced by Urquhart in [Urq96] has as class of algebraic models the class of relevant algebras (bounded distributive lattices $(L, \vee, \wedge, 0, 1)$ with a lattice antimorphism $\neg$ and a binary join hemimorphism $\circ$, with neutral element e, and residuation $\rightarrow$).

Other examples are *BCC and related logics* [OK85], sound and complete with respect to classes of lattice-ordered residuated monoids.

In many of these logics, checking whether a formula ϕ is a theorem can be reduced to checking whether $\mathcal{A} \models \phi \geq e$, where e is a designated element in their algebraic models $\mathcal{A}$, usually the neutral element with respect to a monoid operation (see Anderson and Belnap [AB75] p.364, [Ono93], p.272).

3.3 Logics based on DLOs in applications

Many non-classical logics which occur in a natural way in practical applications have as algebraic models lattices, distributive lattices, or Heyting algebras with operators.

- *Multimodal logics* are often used to model knowledge and belief in multi-agent systems.
- *Logics for reasoning about resources* have often as algebraic models residuated distributive lattices. As the notion of resource is quite general, logics based on residuated lattices occur in a natural way in many applications:
  - *Subtype entailment.* In [DCF02] a relevant logic is proposed for modelling "subtype" relationships between types. Under a certain interpretation of types and type constructors (cf. [DCF02]), subtype checking can be expressed, in algebraic terms, as a uniform word problem with respect to a class of distributive lattices with an additional binary operator, $\rightarrow: L \times L \rightarrow L$, which is a join-hemimorphism in the second argument and maps joins to meets in the first argument.
  - *Shape analysis.* Bunched implication (BI) logics are used e.g. in shape analysis, for modeling allocation and deallocation of resources. Their algebraic models are BI-algebras, i.e. Heyting algebras equipped with an additional residuated commutative monoid structure [IO01, Pym02].
- *Description logics* provide a logical basis for modeling, and reasoning about concepts (classes of objects) and rôles (relationships between objects). One of the important algorithmic problems in description logics, testing subsumption between concepts, can often be expressed as a uniform word problem for classes of lattices and semilattices with operators.

In what follows we present some details on description logics.

3.3.1 Description logics

Description logics are a family of logics for knowledge representation that have been studied extensively in Artificial Intelligence. They provide a logical basis for modeling, and reasoning about objects, classes (or concepts), and relationships (or links, or rôles) between them.

Concepts are one of the central notions in description logics. They are defined

Table 1
Constructors for $\mathcal{ALC}$

Constructor name	Syntax	Semantics
negation	$\neg C$	$D^{\mathcal{I}} \backslash C^{\mathcal{I}}$
conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{x \mid \exists y((x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}})\}$
universal restriction	$\forall R.C$	$\{x \mid \forall y((x, y) \in R^{\mathcal{I}} \implies y \in C^{\mathcal{I}})\}$

with the help of a set of *concept constructors*, starting with a set N_C of *concept names* and a set N_R of *rôles*. The available constructors determine the expressive power of a description logic. For instance, in the description logic $\mathcal{ALC}$, the constructors used are negation ($\neg$), conjunction ($\sqcap$), disjunction ($\sqcup$), existential restriction ($\exists R$) and universal restriction ($\forall R$). A *terminology* (or TBox, for short) is a finite set of concept definitions of the form $A \equiv C$, where A is a concept name and C a concept description. (In description logics it is usually required that TBoxes do not contain multiple definitions.)

The semantics of description logics is defined in terms of interpretations $\mathcal{I} = (D^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $D^{\mathcal{I}}$ is a non-empty set, and the function $\cdot^{\mathcal{I}}$ maps each concept name $C \in N_C$ to a set $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and each rôle name $R \in N_R$ to a binary relation $R^{\mathcal{I}} \subseteq D^{\mathcal{I}} \times D^{\mathcal{I}}$. Table 3.3.1 shows the constructor names used in $\mathcal{ALC}$ and their semantics. The extension of $\cdot^{\mathcal{I}}$ to concept descriptions is inductively defined using the semantics of the constructors described in Table 3.3.1. An interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ if it satisfies all the concept definitions in $\mathcal{T}$, i.e. $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all definitions $A \equiv C$ in $\mathcal{T}$.

Definition 3.3 *Let $\mathcal{T}$ be a TBox, and C_1, C_2 two concept descriptions. C_1 is subsumed by C_2 with respect to $\mathcal{T}$ (for short, $C_1 \sqsubseteq_{\mathcal{T}} C_2$) if and only if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$.*

In practical applications also description logics which are not closed under all Boolean connectives occur in a natural way. If we allow, for instance, only intersection and existential restriction as concept constructors, we obtain the description logic $\mathcal{EL}$, a logic used in terminological reasoning in medicine [Baa03b]. If we allow only intersection and universal restriction as concept constructors, we obtain the description logic $\mathcal{FL}_0$.

We show that deciding the subsumption problem in the description logics $\mathcal{ALC}, \mathcal{EL}$ and $\mathcal{FL}_0$ can be reduced to deciding a uniform word problem with respect to the class of all Boolean algebras (resp. distributive lattices, or semilattices) with operators. To prove this, we first give a translation of concept descriptions into terms in a signature naturally associated with the set of constructors. For every rôle name R, we introduce two unary function symbols, $f_{\exists R}$ and $f_{\forall R}$. The renaming function is inductively defined by:

- $\overline{C} = C$ for every concept name C,
- $\overline{\neg C} = \neg \overline{C}$,
- $\overline{C_1 \sqcap C_2} = \overline{C}_1 \wedge \overline{C}_2$, $\quad \overline{C_1 \sqcup C_2} = \overline{C}_1 \vee \overline{C}_2$,
- $\overline{\exists R.C} = f_{\exists R}(\overline{C})$, $\quad \overline{\forall R.C} = f_{\forall R}(\overline{C})$.

It is easy to see that there exists a one-to-one correspondence between interpretations of description logics, $\mathcal{I} = (D, \cdot^{\mathcal{I}})$ and Boolean algebras of sets $(\mathcal{P}(D), \cup, \cap, \neg, \emptyset, D, \{f_{\exists R}, f_{\forall R}\}_{R \in N_R})$, together with valuations for the $v : N_C \to \mathcal{P}(D)$, where the additional operations are defined, for every $U \subseteq D$, by:

$$
\begin{aligned}
f_{\exists R}(U) &= \{x \mid \exists y((x, y) \in R^{\mathcal{I}} \text{ and } y \in U)\} \\
f_{\forall R}(U) &= \{x \mid \forall y((x, y) \in R^{\mathcal{I}} \implies y \in U)\}.
\end{aligned}
$$

We define the following classes of algebras:

- BAO_{N_R}, the class of all Boolean algebras with operators $\mathbf{B} = (B, \vee, \wedge, \neg, 0, 1, \{f_{\exists R}, f_{\forall R}\}_{R \in N_R})$ where $f_{\exists R}$ is a join hemimorphism, $f_{\forall R}$ is a meet hemimorphism, and $f_{\forall R}(x) = \neg f_{\exists R}(\neg x)$ for every $x \in B$;
- $\mathsf{DLO}^{\forall}_{N_R}$, the class of all bounded distributive lattices with operators $\mathbf{L} = (L, \vee, \wedge, 0, 1, \{f_{\forall R}\}_{R \in N_R})$ such that $f_{\forall R}$ is a meet hemimorphism;
- $\mathsf{DLO}^{\exists}_{N_R}$, the class of all bounded distributive lattices with operators $\mathbf{L} = (L, \vee, \wedge, 0, 1, \{f_{\exists R}\}_{R \in N_R})$ such that $f_{\exists R}$ is a join hemimorphism;
- $\mathsf{SLO}^{\forall}_{N_R}$, the class of all bounded meet-semilattices with operators $\mathbf{S} = (S, \wedge, 1, \{f_{\forall R}\}_{R \in N_R})$ such that $f_{\forall R}$ is a meet hemimorphism;
- $\mathsf{SLO}^{\exists}_{N_R}$, the class of all bounded meet-semilattices with operators $\mathbf{S} = (S, \wedge, 0, 1, \{f_{\exists R}\}_{R \in N_R})$ such that $f_{\exists R}$ is monotone and $f_{\exists R}(0) = 0$.

Theorem 3.4 *For all concept descriptions C_1, C_2 and every TBox $\mathcal{T}$, the following hold:*

(1) If all the constructors of $\mathcal{ALC}$ are allowed then $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if $\mathsf{BAO}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \to \overline{C_1} \leq \overline{C_2}$.

(2) If the only constructors are intersection, union, and existential restriction then $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if $\mathsf{DLO}^{\exists}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \to \overline{C_1} \leq \overline{C_2}$.

(3) If the only constructors are intersection and existential restriction then $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if $\mathsf{SLO}^{\exists}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \to \overline{C_1} \leq \overline{C_2}$.

(4) If the only constructors are intersection, union, and universal restriction then $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if $\mathsf{DLO}^{\forall}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \to \overline{C_1} \leq \overline{C_2}$.

(5) If the only constructors are intersection and universal restriction then

$$C_1 \sqsubseteq_{\mathcal{T}} C_2 \textit{ if and only if } \mathsf{SLO}^{\forall}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \rightarrow \overline{C_1} \leq \overline{C_2}.$$

Proof: The proof is given in Appendix A. □

Theorem 3.5 *The uniform word problem for BAO_{N_R} is* EXPTIME*-complete. The uniform word problem for $\mathsf{SLO}^{\exists}_{N_R}$ is decidable in polynomial time.*

Proof: The proof is given in Appendix B. □

Corollary 3.6 *Concept subsumption with respect to TBoxes in $\mathcal{ALC}$ and $\mathcal{FL}_0$ can be tested in exponential time. Concept subsumption with respect to TBoxes in $\mathcal{EL}$ can be tested in polynomial time.*

The EXPTIME-completeness of concept subsumption in $\mathcal{ALC}$ is well-known. Recently, Kazakov and de Nivelle proved that concept subsumption with respect to TBoxes in $\mathcal{FL}_0$ is PSPACE-complete [KN03]. The polynomial time complexity of concept subsumption with respect to TBoxes in $\mathcal{EL}$ was first proved by Baader [Baa03a]. Theorem 3.5 provides a much simpler proof of this fact, and shows, in addition, that the restriction imposed in [Baa03a] that TBoxes do not contain multiple definitions is not really necessary for polynomial time decidability of concept subsumption in $\mathcal{EL}$.

4 Automated theorem proving

We present several approaches to automated theorem proving in non-classical logic based on translations to classical logic, which allow the use of (various refinements of) resolution. Because of space limitations, neither tableau nor proof-theoretic methods are discussed, although they often provide optimal time and space complexity bounds.

We first show that various versions of many-valued resolution for finitely-valued logics can be reconstructed by using general saturation-based techniques for first order theories of transitive relations. The inference systems which we obtain this way are much more restricted, in particular by ordering constraints and selection functions. We then consider other non-classical logics which are not finitely valued, but for which nevertheless such embeddings into classical logics are possible. We present, for instance, a translation to clause form for prenex first order Gödel logics [BFC01] which allowed the use of saturation-based techniques for dense total orderings, and then focus on propositional logics based on distributive lattices with operators (possibly many-sorted). We show that resolution-based decision procedures can be obtained in many interesting cases.

Propositional finitely-valued logics are very similar to propositional classical logics, and checking the validity of fomulae in finitely-valued logics can be reduced in a natural way to satisfiability checking in propositional classical logic. Reductions of validity testing in many-valued logics to SAT checking in propositional logics - including a study of various possible optimizations, and various complexity studied - were studied in various papers, among which we mention e.g. [BHM99, Man00, BHM00, BHM01].
In what follows we will refer to automated theorem proving in *first order* finitely-valued logics. Several papers on many-valued logics present methods for automated theorem proving which are similar to classical resolution. As classical resolution, they are based on two steps: (i) translation to clause form (usually called, in this context, signed clause form), and (ii) resolution. Here we only mention a few such results. (The list below is far from being exhaustive: it does not mention all results of historical interest; for an overview of automated deduction in many-valued logic we refer to [Häh93, Häh97a, BFS99, Häh01].)

- In [BF95], Baaz and Fermüller extended the resolution procedure to arbitrary finitely-valued logics. They describe methods for translation to a *many-valued clause form* (many-valued literals, L^v, are atomic formulae superscripted by truth values; many-valued clauses are disjunctions of many-valued literals), formulate a sound and complete many-valued resolution calculus, and show that the completeness of the calculus is preserved when applying simplification rules such as subsumption and deletion of certain types of tautologies.
- Many-valued resolution has also been extended to signed literals $S : L$, where S is a *set* of truth values, in [Häh94b], also see [BFS99].
- A special kind of signs (when the set A of truth values is ordered by a total order $\leq$) are *regular signs* [Häh94b, Häh96], i.e. signs of the form $\uparrow v_j := \{v \mid v \geq v_j\}$ or $\downarrow v_j := \{v \mid v \leq v_j\}$.
- A notion of regular signs has also been introduced in the context of annotated logics [KL92, LMR98] when the set A of truth values is a complete lattice with respect to an order $\leq$, with greatest element $\top$ and least element $\bot$. In this context, a regular literal is a literal with a sign of the form $\uparrow v$ or $A \backslash \uparrow v$ (notation: $\sim\uparrow v$), where $v \in A$; a regular clause is a disjunction of regular literals. An inference system consisting of annotated resolution, annotated reduction and elimination was shown to be sound and refutationally complete [KL92, LMR98].

The completeness proofs of these resolution-like calculi for many-valued logics are, in all cases, very similar to the completeness proof of classical resolution. A simple explanation for this fact is given in [GS00]: we show that unsatisfiability of a set of many-valued (or regular) clauses can be checked by using a simple translation to classical logic: many-valued literals are translated by replacing L^v with $L \approx v$; regular (signed) literals are translated by replacing $\uparrow v : L$ with $L \geq v$ and $\sim\uparrow v : L$ with $L \not\geq v$.

In addition, clauses which describe the properties of the set of truth values have to be added. We also need to explicitly express the fact that $\approx$ is a congruence (or, respectively, that $\leq$ is reflexive and transitive). However, congruence or transitivity axioms are extremely prolific in the context of resolution-based theorem proving. Refinements of resolution such as *superposition* and *ordered chaining* (calculi which encode inferences with the congruence, resp. transitivity axioms), have been devised by Bachmair and Ganzinger [BG94, BG98]. The main idea in [GS00] is to specialize superposition, resp. ordered chaining to the type of literals generated using the encoding above. This allows to reconstruct known completeness results (e.g. the calculi of Baaz and Fermüller, Kifer and Lozinskii, and Hähnle) mentioned at the beginning of this section. We describe these ideas in what follows.

As a convention, we assume that the set of truth values is $A = \{v_1, \dots, v_n\}$. We call the constants in A *truth values* (and denote them by u, v, w, s, t), and the terms of the form $R(t_1, \dots, t_n)$, with R a predicate symbol in the language of the many-valued logic under consideration, *predicate terms* (and denote them by L). As usual, the symbols $\vee$ and $\neg$ denote disjunction and negation, respectively. Formal equality is denoted by $\approx$, and atoms of the form $s \approx t$ are called equations. The symmetry of equality is built into the notation: we do not distinguish between $s \approx t$ and $t \approx s$. Negative (in)equations are also written as $s \not\approx t$, resp. $s \not\leq t$. Orderings on syntactic expressions play an important rôle in theorem proving. Any ordering on ground terms can be extended to ground literals, and then to ground clauses (by taking the multiset extension). We say that a literal L is maximal with respect to a clause C (denoted $L \succeq C$) if $L' \succ L$ for no literal L' in C; and that L is strictly maximal with respect to C (denoted $L \succ C$) if $L' \succeq L$ for no L' in C. In what follows let $\succ$ be a noetherian ordering on ground literals. In order to avoid unnecessary complication in the presentation we will only deal with the propositional variants of the various inference systems. That is, unless explicitly stated otherwise, all expressions (terms, literals, formulas) are assumed to be *ground*, that is, to not contain any variables. As the various completeness results also hold for infinite sets of clauses, lifting can be done in the standard manner by viewing non-ground expressions to represent the set of their ground instances and by employing unification to avoid their explicit enumeration.

4.1.1 Many-valued clauses

With every set of many-valued clauses Φ, consisting of literals L^v signed by truth values, we associate a set Φ_1 of first order clauses by replacing every signed literal L^v in Φ by the equation $L \approx v$. In what follows, literals of the form $L \approx v$ are called *MV*-literals, and clauses consisting of *MV*-literals are called *MV*-clauses.

Semantically, equality is a congruence. A formula is called equationally satisfied in an interpretation I whenever the formula is satisfied in I, and the interpretation of $\approx$ in I is a congruence over the given signature, satisfying the corresponding set of *congruence axioms Eq*.

Proposition 4.1 [**GS00**] *A set Φ of many-valued clauses is satisfiable if and only if $\Phi_1 \cup \Phi_A \cup \mathsf{Fin}$ is equationally satisfiable, where*

$$\Phi_A = \{u \not\approx v \mid u, v \in A, u \neq v\},$$
$$\mathsf{Fin} = \{s \approx v_1 \vee \ldots \vee s \approx v_n \mid s \text{ a term of sort } \mathsf{for}\}$$

are sets of clauses which express that there are exactly n pairwise different (congruence classes of) truth values $v_1, \ldots, v_n$ in any equality Herbrand interpretation satisfying $\Phi_A \cup \mathsf{Fin}$.

Satisfiability of $\Phi_1 \cup \Phi_A \cup \mathsf{Fin}$ can for instance be checked by using superposition [BG94]. When applied to sets of *MV*-clauses, the superposition calculus specializes to the following calculus, *SMV*:

Positive *MV*-superposition.

$$\frac{L \approx t \vee C \qquad L \approx v \vee D}{C \vee D}$$

provided that $t \neq v$ and (i) $L \approx t \succ C$; (ii) $L \approx v \succ D$; (iii) $L \approx v \succ L \approx t$.

Ordered factoring.

$$\frac{L \approx t \vee L \approx t \vee C}{L \approx t \vee C}$$

provided that $L \approx t$ is maximal with respect to C.

In fact, if a suitable notion of $Eq \cup \Phi_A$-redundancy is exploited, inferences with clauses in Φ_A can be avoided; also inferences with clauses in Fin can safely be ignored. This means that in order to check whether the set Φ of many-valued clauses is satisfiable, it is sufficient to check if the empty clause can be derived by *MV*-resolution from Φ_1 only (not from $\Phi_1 \cup \Phi_A \cup \mathsf{Fin}$).

Theorem 4.2 [**GS00**] *Let Φ be a set of many-valued clauses and let $\Phi_1 \cup \Phi_A \cup \mathsf{Fin}$ be the encoding of Φ in first order logic. Then Φ is unsatisfiable if* *and only if the empty clause can be derived from Φ_1 by a finite number of applications of inference rules in MV.*

As superposition into subterms is not possible for MV-clauses, $\succ$ needs not be a reduction ordering on terms. In conclusion, the calculus SMV is an order-refinement of the many-valued resolution method of Baaz and Fermüller. Its compatibility with simplification techniques which redundancy justifies follows from results on superposition and Theorem 4.2 (for further details we refer to [GS00]).

4.1.2 Annotated and regular clauses

Let $(A, \leq_A)$ be a finite partially ordered set, and $\mathsf{Min}(A)$ the set of minimal elements in A. Let Φ be a set of regular clauses, i.e. clauses containing only literals of the form $\uparrow v{:}L$ or $\sim\uparrow v{:}L$, where $v \in A$. The *encoding of* Φ *in first order logic*, Φ_1, is the set of clauses obtained from Φ by replacing $\uparrow v{:}L$ by $v \leq L$ and $\sim\uparrow v{:}L$ by $v \not\leq L$, where $v \not\leq L$ is an abbreviation for $\neg(v{\leq}L)$. Consider the following additional sets of clauses:

$$\begin{aligned}
\Phi_A &= \{u \trianglelefteq v \mid u, v \in A, u \trianglelefteq_A v, \trianglelefteq \in \{\leq, \not\leq, \not\approx\}\} \\
\mathsf{Sup} &= \{(u \not\leq s) \vee (v \not\leq s) \vee (\mathsf{sup}(u,v) \leq s) \mid \mathsf{sup}(u,v) \text{ exists in } A, \\
&\qquad s \text{ a term of sort } \mathsf{for}\} \\
\mathsf{Min} &= \{ \bigvee_{m \in \mathsf{Min}(A)} (m{\leq}s) \mid s \text{ a term of sort } \mathsf{for}\}.
\end{aligned}$$

In the following we will only consider clauses with inequalities $s{\leq}t$ as atoms. Equalities $s \approx t$ will be used on the meta-level as an abbreviation for conjunctions $(s{\leq}t) \wedge (t{\leq}s)$. Fin will again denote the set of clauses (represented by) $\{s \approx v_1 \vee \ldots \vee s \approx v_n \mid s \text{ a term of sort } \mathsf{for}\}$.

By Tr we denote the transitivity axiom for $\leq$: $(x \leq y) \wedge (y \leq z) \rightarrow (x \leq z)$. By a *transitivity interpretation* we mean a model of Tr. We say that a set of clauses N is *Tr-satisfiable* if there exists a transitivity interpretation I that satisfies N. Otherwise N is *Tr-unsatisfiable.*

Proposition 4.3 [GS00] *If Φ be a set of regular clauses then:*

(1) If $(A, \leq_A)$ is a partially ordered set, then Φ is satisfiable if and only if $\Phi_1 \cup \Phi_A \cup \mathsf{Fin}$ is (classically) Tr-satisfiable.
(2) If $(A, \leq_A)$ is a sup-semilattice, then Φ is satisfiable if and only if $\Phi_1 \cup \Phi_A \cup \mathsf{Sup} \cup \mathsf{Min}$ is (classically) Tr-satisfiable.
(3) If $(A, \leq_A)$ is a totally-ordered set with minimal element $\perp$ then Φ is satisfiable if and only if $\Phi_1 \cup \Phi_A \cup \{\perp{\leq}s \mid s \text{ a term of sort } \mathsf{for}\}$ is (classically) Tr-satisfiable.

In what follows we refer to literals of the form $v \leq L$ or $v \not\leq L$, where L is a predicate term and v is a truth value, as $\leq$-*literals*. A $\leq$-clause is a disjunction of $\leq$-literals. When applied to $\leq$-clauses, the chaining calculus of Bachmair and Ganzinger [BG98] specializes to the following calculus, *CS*:

Negative chaining for $\leq$-clauses.

$$\frac{(u \leq L) \vee C \qquad (v \not\leq L) \vee D}{C \vee D}$$

provided that $v \leq_A u$ and (i) holds.

Sup-reduction.

$$\frac{(u \leq L) \vee C \qquad (v \leq L) \vee D}{(\mathsf{sup}(u, v) \leq L) \vee C \vee D}$$

provided that u and v are incomparable, and (ii) holds.

Ordered (positive) factoring.

$$\frac{B \vee B \vee C}{B \vee C}$$

provided that B is maximal with respect to C.

The restrictions are: (i) $(u \leq L) \succ C$ and $(v \not\leq L) \succeq D$; (ii) $(u \leq L) \succ C$ and $(v \leq L) \succ D$.

Theorem 4.4 [GS00] *Let Φ be a set of regular clauses over a finite set A truth values, and let Φ_1 be the encoding of Φ in first order logic.*

(1) If A is a sup-semilattice with minimal elements $\mathsf{Min}(A)$ then Φ is unsatisfiable if and only if the empty clause can be derived from $\Phi_1 \cup \mathsf{Min}$ by a finite number of applications of inference rules in CS.
(2) Assume that A is a complete lattice with minimal element $\perp$. Let Φ_2 be obtained from Φ_1 by removing all literals of the form $\perp \not\leq L$ and all clauses containing a literal of the form $\perp \leq L$. Then Φ is unsatisfiable if and only if there exists a derivation in CS of the empty clause from Φ_2.

As chaining into subterms is not possible for $\leq$-clauses, $\succ$ needs not be a reduction ordering on terms. Thus, the calculus *CS* is an order-refinement of the annotated resolution calculus in [LMR98]. If $(A, \leq_A)$ is a totally ordered set then sup-reduction never applies. Let *CT* be the inference system consisting of all inference rules in *CS* except sup-reduction. The refutational completeness of the *CT* calculus in the case when $(A, \leq_A)$ is a totally ordered set is a direct consequence of Theorem 4.4.

Since first order many-valued logics are undecidable, in general we cannot hope to obtain decision procedures based on the calculi above. It can however be seen that, in the propositional case, they yield exponential time decision procedures in the length of the input.

4.2 Resolution for infinitely-valued logics

The method for translation to clause form for finitely valued logics of Baaz and Fermüller cannot be applied in general when the set of truth values is infinite, nor for logics whose semantics is given in terms of a *class* of algebras. There have been several attempts for giving methods for automated theorem proving in infinitely-valued logics. A resolution-like calculus for the infinitely-valued sentential calculus of Łukasiewicz based on a different representation of clauses was given, for instance, by Mundici and Olivetti in [MO98]. We do not discuss this approach here in detail. Instead we present methods which in our opinion are natural extensions of the methods used in the finitely-valued case, namely approaches which rely on reductions to mixed integer programming, on reductions of infinitely-valued to finitely-valued logics, or on embeddings into theories of dense total orderings.

4.2.1 Propositional Łukasiewicz and Gödel logics

One of the possibilities of checking validity in infinitely-valued logic is to reduce the problem to checking validity in a suitable finitely-valued logic [2]. Aguzzoli and Ciabattoni [AC00] did this for the infinitely valued propositional Łukasiewicz logic Ł. They proved that a formula ϕ is valid in Ł if and only if it is valid in a suitable m-valued Łukasiewicz logic $Ł_m$, where m only depends on the length of the formula to be proved (in fact, $m = 2^{\mathsf{lenght}(\phi)} + 1$). Thus, in this case, the methods discussed in Section 4.1 may still be used, but could be highly inefficient, since the size of the algebra $Ł_m$ is exponential in the length of the formula ϕ.

An alternative approach, proposed by Hähnle in [Häh94a, Häh97b] is based on a reduction to mixed integer programming. This method uses ideas similar to those used for the translation to regular clauses for regular (finitely-valued) logics: CNF translations are obtained which allow reductions to mixed integer programming (MIP) in the case of infinitely-valued propositional Łukasiewicz

[2] Similar ideas were used also in other contexts. For instance, Caferra and Zabel [CZ90] give a translation from the propositional modal logic $S5$, viewed as an infinite-valued logic, into finite-valued logics.

logic and Gödel logics. (The connectives of the product logic, however, lead outside MIP, and into non-linear programming.) The method is applicable for a whole class of many-valued logics, namely for "MIP"-representable logics, i.e. propositional many-valued logics whose connectives have the property that their graphs can be represented as solutions of mixed integer programs, i.e. are of the form

> $\Gamma \subseteq [0,1]^k$, where there exists a system J of linear inequations over variables $\{x_1, \ldots, x_k, x_{k+1}, \ldots x_n\}$, where the first k variables have domain $[0,1]$ and the remaining ones have domain $\{0,1\}$, and
>
> $\Gamma = \{(\rho(x_1), \ldots, \rho(x_k)) \mid \rho$ solution of $J\sigma$ for some $\sigma : \{x_{k+1}, \ldots, x_n\} \to \{0,1\}\}$.

The infinitely-valued Łukasiewicz logic is "MIP"-representable. For instance, the graph $z = \circ_{Ł}(x,y) := \max\{0, x+y-1\}$ is the solution of the following system of linear inequations:

$$\begin{cases} x+y+t-z \geq 1 \\ -x-y+t+z \geq -1 \\ -x-y-t \geq -2 \\ x+y+t \geq 1 \\ -t-z \geq -1 \end{cases}$$

Theorem 4.5 [Häh94a, Häh97b]

(1) If $\phi(\overline{p})$ is a formula of an MIP-representable logic then there is an mixed integer program J_ϕ (in fact, a system of linear inequations of size linear in the size of ϕ) with argument variables $\overline{p}$ and output variable y whose solutions restricted to $(\overline{p}, y)$ are completely described by the [0,1]-function associated[3] with ϕ, $f_\varphi(\overline{p})$.

(2) Let J be a system of linear inequations over variables P. Then there exists a formula in propositional Łukasiewicz logic with variables P which is satisfiable if and only if J has a solution.

Proof: (1) follows from the fact that the composition of MIP representable functions is again MIP representable; (2) uses a theorem of McNaughton. □

As in the case of infinitely-valued Łukasiewicz logics the translation above was proved to be polynomial [Häh94a], the reduction to mixed integer programming in Theorem 4.5 justifies the NP-easiness (and therefore, the NP-completeness) of satisfiability checking in infinitely-valued Łukasiewicz logic.

[3] Every formula ϕ of $\mathcal{L}$ over r variables $\{p_1, \ldots, p_r\}$ defines an r-ary function $f_\phi : [0,1]^r \to [0,1]$ defined for every $(x_1, \ldots, x_r) \in [0,1]^r$ by $\overline{v}(\phi)$, where $v : \{p_1, \ldots, p_r\} \to [0,1]$ is defined by $v(p_i) = x_i$.

4.2.2 First order Łukasiewicz and Gödel logics

First order fuzzy logics are more complicated than propositional fuzzy logics; they are also more complicated than first order classical logic. Usually it is difficult, or even impossible, to construct a prenex normal form. Therefore the theory of automated theorem proving in these logics is not so well developed.

In what follows we present a method for automated theorem proving for a fragment of first order Gödel logic (with projection modalities) proposed by Baaz, Fermüller and Ciabattoni [BFC01] . The method uses a translation into the first order theory of dense total orderings with endpoints, and can be seen as a natural extension – to the first order case – of the translation to mixed integer programming for propositional Łukasiewicz and Gödel logics described in the previous section.

Semantically, first order Gödel logic G_∞ is viewed as an infinitely-valued logic, with the real interval $[0, 1]$ as set of truth values, equipped with the Gödel t-norm $\circ : [0, 1]^2 \to [0, 1]$, $x \circ y := \mathsf{min}(x, y)$, and its residuation $\to$; the semantics of the quantifiers is given by supremum (for $\exists$) and infimum (for $\forall$). In [BFC01] the logic G^Δ_∞ is studied. G^Δ_∞ is obtained by extending G_∞ with projection modalities ∇, Δ, interpreted by the maps $\nabla, \Delta : [0, 1] \to \{0, 1\}$, where $\nabla(x) = 1$ if and only if $x = 0$, and $\Delta(x) = 1$ if and only if $x = 1$.

Theorem 4.6 [BFC01] *For each prenex formula of G^Δ_∞, of the form $\varphi = Q_1y_1 \dots Q_ny_n\phi(y_1, \dots, y_n)$, there exists a set $CF^d(\exists\overline{x}\phi(\overline{x}))$ of order clauses*[4] *(which can be computed in linear time), such that $Q_1y_1 \dots Q_ny_n\phi(y_1, \dots, y_n)$ is valid in G^Δ_∞ if and only if $CF^d(\exists\overline{x}\phi(\overline{x}))$ is unsatisfiable with respect to the theory of dense total orderings with endpoints.*

The embedding is, up to a certain extent, similar to that described in Section 4.1 in the case of finitely-valued logics based on partially ordered, or totally ordered sets. A chaining calculus for dense total orderings with endpoints [BG98] is then used for efficient deduction with such sets of clauses. However, since G^Δ_∞ is undecidable, one cannot hope to use chaining for dense total orderings with endpoints as a decision procedure in this case.

In order to be able to use resolution as a decision procedure, in what follows we focus on propositional non-classical logics.

[4] Order clauses are classical clauses with predicate symbols $<$ and $\leq$ interpreted as total dense orderings (strict and reflexive, respectively).

4.3 Resolution-based decision procedures for modal logics

In the attempt of understanding why so many modal logics are decidable many authors noticed that the definition of the Kripke-style semantics justifies an embedding into (decidable fragments of) classical logic. For instance, in [ABN98] Andréka, Van Benthem and Németi introduced the so-called *guarded fragment* (GF) of classical logic, which abstracts many of the properties of formulae obtained from the structure-preserving translation to clause form for many modal logics. The main advantage of the embedding into first order logic is that it is very suitable to use for automated theorem proving, since proof techniques developed for classical logic can be used for free. Refinements of resolution such as ordered resolution, the use of selection functions, and specially devised calculi to deal with equivalence (or congruence) relations, or with transitive relations proved to be extremely useful in this context. For instance, ordered resolution was used as a decision procedure for modal logics such as K in [Ohl93, Sch99], ordered chaining with selection was used to obtain (doubly exponential [5]) decision procedures for the relational translation of propositional modal logics with modal operators satisfying the axioms D, T or $\mathsf{4}$ in [GHM01]. A doubly-exponential decision procedure for the guarded fragment with equality, that uses superposition, was given in [GN99].

The embedding into classical logic for modal logics mentioned above is a special instance of a more general result, which we present in the next section.

4.4 Resolution and uniform word problems in DLO

Uniform word problems are relevant and important in mathematics and computer science. On the one hand, as pointed out already in Section 3.2, most of the validity problems in logics whose algebraic semantics can be given in terms of distributive lattices with operators, can be formulated as word problems with respect to classes of algebras. More complex problems, such as the problem of checking concept subsumption in terminological databases can be formulated as uniform word problems. On the other hand, the study of uniform word problems has a great theoretical importance, because, for every quasi-variety $\mathcal{V}$ of algebras, the decidability of the uniform word problem of $\mathcal{V}$ implies the decidability of the universal theory of $\mathcal{V}$.

In the previous sections on automated theorem proving in many-valued logics, the information about the algebra of truth values was directly used, both in

[5] In [GHM01] it is actually showed that a single-exponential space representation can be obtained by splitting the clauses into their variable-disjoint regions and connecting them with the help of auxiliary monadic predicates.

the translation to clause form, and - sometimes - for the resolution process. A similar approach would theoretically be possible also in this case: Given a (quasi-)variety $\mathcal{V}$ presented by a finite set E of equations (resp. implications), one possibility of proving

$$\mathcal{V} \models \bigwedge_{i=1}^{n} t_i = t_i' \rightarrow t = t' \tag{7}$$

would be to show that $\bigwedge_{i=1}^{n} t_i = t_i' \rightarrow t = t'$ is a consequence, in equational logic, of the axioms E of $\mathcal{V}$. However, if $\mathcal{V}$ is a class of (distributive) lattices with operators then equational reasoning modulo E may be quite inefficient, due to the necessity of handling, for instance, axioms such as associativity, commutativity and idempotence of the lattice axioms.

In many cases, it is possible to avoid equational reasoning in lattice theory. In what follows we show that checking implications of the form (7) can often be reduced to checking whether they hold in certain specific algebras of sets. All terms occurring in a problem of type (7) can then be encoded as *sets*; the lattice operations are be encoded by the usual set operations: suprema are unions and infima are intersections. Further, sets are encoded as *unary predicates*, and unions resp. intersections are expressed using logical disjunction resp. conjunction. This remark can be used for obtaining embeddings into decidable fragments of first order logic (without equality). These results justify, in particular, existing embeddings into classical logic for many-valued logics over finite distributive lattices with operators [Sof01], but also for modal logics, and, in many cases yield optimal resolution-based decision procedures.

4.4.1 Distributive lattices with operators

We want to make the class of algebras we consider broad enough to encompass operations which satisfy equations such as (1)–(6) presented in Section 3.2, but also operations between different lattices, such as Galois connections, i.e. pairs (f, g) of maps $f : L_1 \rightarrow L_2$, $g : L_2 \rightarrow L_1$, with the property that

$$\begin{aligned} f(x \vee y) &= f(x) \vee f(y), \quad f(0) = 0, & (8)\\ g(x \wedge y) &= g(x) \wedge g(y), \quad g(1) = 1, & (9)\\ f(x) \leq y \text{ iff } x &\leq g(y) \quad \text{for all } x \in L_1, y \in L_2. & (10) \end{aligned}$$

or operators which take numeric values, for instance (assuming that c is some cost function) $\mathsf{maxcost}, \mathsf{mincost} : \mathcal{P}(X) \rightarrow \mathbb{N}$, where $\mathsf{maxcost}(U) = \mathsf{max}\{c(u) \mid u \in U\}$, and $\mathsf{mincost}(U) = \mathsf{min}\{c(u) \mid u \in U\}$. It is easy to see that:

$$\begin{aligned} \mathsf{maxcost}(U_1 \cup U_2) &= \mathsf{max}(\mathsf{maxcost}(U_1), \mathsf{maxcost}(U_2)),\\ \mathsf{mincost}(U_1 \cup U_2) &= \mathsf{min}(\mathsf{mincost}(U_1), \mathsf{mincost}(U_2)). \end{aligned}$$

Therefore, we consider classes of many-sorted algebraic structures, with many-sorted operations. We now formally define operators that have properties such as (1)–(9) above.

Definition 4.7 *Let S be a set of sorts, $\{\mathbf{L}_s\}_{s\in S}$ be an S-sorted family of bounded lattices $\mathbf{L}_s = (L_s, \vee, \wedge, 0, 1)$ and let $s_1, \dots, s_n, s \in S$. A join hemimorphism of type $s_1 \dots s_n \to s$ is a function $f : L_{s_1} \times \cdots \times L_{s_n} \to L_s$ such that for every $i, 1 \leq i \leq n$,*

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

We say that a map $f : L_{s_1} \times \cdots \times L_{s_n} \to L_s$ is a join hemimorphism of type $s_1^{\varepsilon_1} \dots s_n^{\varepsilon_n} \to s^{\varepsilon}$, where $\varepsilon_1, \dots, \varepsilon_n, \varepsilon \in \{-1, +1\}$, if $f : L_{s_1}^{\varepsilon_1} \times \cdots \times L_{s_n}^{\varepsilon_n} \to L_s^{\varepsilon}$ is a join hemimorphism, where $\mathbf{L}^{+1} := \mathbf{L}$ and $\mathbf{L}^{-1} := \mathbf{L}^d$, the order dual of $\mathbf{L} = (L, \vee, \wedge, 0, 1)$, i.e. the lattice $(L, \vee^d, \wedge^d, 0^d, 1^d)$, where for every $x, y \in L$, $x \vee^d y = x \wedge y$, $x \wedge^d y = x \vee y$; $0^d = 1$; and $1^d = 0$.

Definition 4.8 *Let $\{\mathbf{L}_s\}_{s\in S}$ be an S-sorted family of bounded lattices and let f, g be two n-ary operators such that $f : \mathbf{L}_{s_1}^{\varepsilon_1} \times \cdots \times \mathbf{L}_{s_i} \times \cdots \times \mathbf{L}_{s_n}^{\varepsilon_n} \to L_s$ and $g : \mathbf{L}_{s_1}^{\varepsilon_1} \times \cdots \times \mathbf{L}_s^d \times \cdots \times \mathbf{L}_{s_n}^{\varepsilon_n} \to L_{s_i}^d$ are join hemimorphisms. We say that g is an i-residuation*[6] *associated with f if for all $a_1 \in L_{s_1}, \dots, a_n \in L_{s_n}, a \in L_s$:*

$$f(a_1, \dots, a_n) \leq a \text{ if and only if } a_i \leq g(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n).$$

Examples:

(1) The operator $\Diamond$ on a modal algebra $\mathbf{B}$ is a join hemimorphism. The operator $\Box$ on $\mathbf{B}$ is a join hemimorphism on the dual $\mathbf{B}^d$ of $\mathbf{B}$.
(2) A binary lattice operation $\Rightarrow$ satisfying conditions (5) and (6) is a join hemimorphism of type $\mathsf{lat}, \mathsf{lat}^d \to \mathsf{lat}^d$.
(3) Let $\mathbf{L_1}, \mathbf{L_2}$ be two lattices and let $f : \mathbf{L_1} \to \mathbf{L_2}$ and $g : \mathbf{L_2} \to \mathbf{L_1}$ be a Galois connection. Let $(\mathbf{L}_1, \mathbf{L}_2)$ be the 2-sorted algebra with sorts $S = \{\mathsf{l}_1, \mathsf{l}_2\}$. Then f is a join hemimorphism of type $l_1 \to l_2$, g is a join hemimorphism of type $l_2^d \to l_1^d$; g is the 1-residuation associated with f.
(4) Let $(\mathbf{L}, \mathbf{C}_{n+1})$ be the 2-sorted algebra with sorts $S = \{\mathsf{lat}, \mathsf{num}\}$, where L is a bounded lattice, and $C_{n+1} = (\{0, 1, \dots, n\}, \vee, \wedge, 0, n)$ is the $n+1$-element chain. A function $f : \mathbf{L} \to \mathbf{C}_{n+1}$ that associates with every element of L an element of $\{0, 1, \dots, n\}$ such that $f(x \vee y) = f(x) \vee f(y)$ and $f(0) = 0$ is a join hemimorphism of type $\mathsf{lat} \to \mathsf{num}$.

[6] It can be seen that two i-residuations associated with the same operator coincide.

4.4.2 Algebraic and relational models

We establish a link between truth of universal sentences in classes of S-sorted distributive lattices with operators and truth in S-sorted relational structures.

Definition 4.9 *An* S-sorted RT Σ-relational structure $(\{(X_s, \leq)\}_{s\in S}, \{R_X\}_{R\in\Sigma})$ *is an S-sorted family of sets, each endowed with a reflexive and transitive relation $\leq$ and with additional maps and relations indexed by Σ, where, if $\varepsilon_1, \dots, \varepsilon_n, \varepsilon \in \{-1, +1\}$, $s_1, \dots, s_n, s \in S$ then: if $R \in \Sigma$ is of type $s_1^{\varepsilon_1} \dots s_n^{\varepsilon_n} \to s^{\varepsilon}$, $R_X \subseteq \prod_{i=1}^n X_{s_i} \times X_s$ is increasing if $\varepsilon = +1$ and decreasing if $\varepsilon = -1$.*

We denote by DLO_Σ^S, BAO_Σ^S, and HAO_Σ^S the class of all S-sorted bounded distributive lattices, Boolean algebras, and resp. Heyting algebras, with operators in Σ, and by RT_Σ^S the class of all S-sorted RT Σ-relational structures.

If $\mathbf{L} \in \mathsf{DLO}_\Sigma^S$, let $D(\mathbf{L}) = (\{(\mathcal{F}_p(\mathbf{L}_s), \subseteq)\}_{s\in S}, \{R_f\}_{f\in\Sigma})$, where if $f : \prod_{i=1}^n \mathbf{L}_{s_i}^{\varepsilon_i} \to \mathbf{L}_s^{\varepsilon}$ is a join hemimorphism, where $\varepsilon_1, \dots, \varepsilon_n, \varepsilon \in \{-1, +1\}$, then we define

$$R_f(F_1, \dots, F_n, F) \text{ if and only if } f(F_1^{\varepsilon_1}, \dots, F_n^{\varepsilon_n}) \subseteq F^{\varepsilon},$$

where $F^{+1} := F$ and F^{-1} is the complement of F.

Conversely, for every $\mathbf{X} \in RT_\Sigma^S$, let $\mathcal{O}(\mathbf{X}) = (\{\mathcal{O}(\mathbf{X_s})\}_{s\in S}, \{f_R\}_{R\in\Sigma})$, where, for every $s \in S$, $\mathcal{O}(\mathbf{X_s}) = (\mathcal{O}(X_s), \cup, \cap, \emptyset, X_s)$ is the bounded distributive lattice of all upwards-closed subsets of X_s, and if $R \subseteq \prod_{i=1}^n X_{s_i} \times X_s$ is of type $s_1^{\varepsilon_1} \dots s_n^{\varepsilon_n} \to s^{\varepsilon}$ then $f_R : \prod_{i=1}^n \mathcal{O}(X_{s_i}) \to \mathcal{O}(X_s)$ is defined, for every $(U_1, \dots U_n) \in \prod_{i=1}^n \mathcal{O}(X_{s_i})$ by

$$f_R(U_1, \dots, U_n) = \left(R^{-1}(U_1^{\varepsilon_1}, \dots, U_n^{\varepsilon_1})\right)^{\varepsilon}, \tag{11}$$

where $R^{-1}(U_1, \dots, U_n) = \{x \mid \exists x_1 \dots x_n(x_1 \in U_1, \dots, x_n \in U_n, R(x_1, \dots, x_n, x))\}$, and $U^{+1} := U$ and U^{-1} is the complement of U.

Theorem 4.10 **[Sof02]** *For every $\mathbf{L} = (\{\mathbf{L}_s\}_{s\in S}, \{f_L\}_{f\in\Sigma}) \in \mathsf{DLO}_\Sigma^S$, $D(\mathbf{L}) \in RT_\Sigma^S$, and $\eta_L : \mathbf{L} \to \mathcal{O}(D(\mathbf{L}))$ defined for every $s \in S$ and every $x \in L_s$ by $\eta_L^s(x) = \{F \in \mathcal{F}_p(\mathbf{L}_s) \mid x \in F\}$ is an injective homomorphism between algebras in DLO_Σ^S.*

Similar correspondences can be established for (possibly many-sorted) Boolean algebras or Heyting algebras with operators. Note that in the case of Boolean algebras, the dual spaces are discretely ordered (i.e. $x \leq y$ if and only if $x = y$).

Also, note that for every preordered set $\mathbf{X} = (X, \leq)$, a Heyting implication ($\Rightarrow$) and a Heyting negation ($\neg$) can be defined on $\mathcal{O}(\mathbf{X})$ by:

$$U \Rightarrow V := \{x \mid \forall y((x \leq y, y \in U) \to y \in V\}; \qquad \neg U := U \Rightarrow \emptyset.$$

We consider subclasses $\mathcal{V}$ of DLO^S_Σ, BAO^S_Σ or HAO^S_Σ that satisfy the condition:

(K) There exists a $\mathcal{K} \subseteq RT^S_\Sigma$ such that (i) for every $\mathbf{A} \in \mathcal{V}$, $D(\mathbf{A}) \in \mathcal{K}$;
(ii) for every $\mathbf{X} \in \mathcal{K}$, $\mathcal{O}(\mathbf{X}) \in \mathcal{V}$.

Example 4.11 *Condition (K) holds in the following cases:*

(1) $\mathcal{V} = \mathsf{DLO}^S_\Sigma$; $\mathcal{K} = RT^S_\Sigma$.

(2) $\mathcal{V} = \mathsf{RDLO}^S_{\Sigma,\mathsf{Res}}$ *the class of all algebras in* DLO^S_Σ *satisfying the residuation conditions in* Res;
$\mathcal{K} = RT^S_{\Sigma,\mathsf{Res}}$ *the class of those spaces in* RT^S_Σ *which satisfy in addition:*
$\{R_f(x_1, \dots, x_n, x) \leftrightarrow R_g(x_1, \dots, x, \dots, x_n, x_i) \mid$ *"g i-residuation of f"* $\in \mathsf{Res}\}$.

(3) $\mathcal{V} = \mathsf{BAO}^S_\Sigma$; $\mathcal{K} = R^S_\Sigma$ *the subclass of* RT^S_Σ *consisting only of discretely-ordered spaces.*

If $\mathbf{A} \in \mathsf{D}_{01}$ *is a fixed* finite *lattice and* $S = \{\mathsf{lat}, \mathsf{a}\}$, *then condition (K) holds in the following cases:*

(4) $\mathcal{V} = \mathsf{DLO}^A_\Sigma = \{(\mathbf{L}, \mathbf{A}, \{f\}_{f \in \Sigma_L}, \{g\}_{g \in \Sigma_A}) \mid \mathbf{L} \in \mathsf{D}_{01}; f : \mathbf{L}^k \to \mathbf{L}$ *join hemimorphism if* $f \in \Sigma_L$; $g : \mathbf{L}^m \to \mathbf{A}$ *join hemimorphism if* $g \in \Sigma_A\}$;
$\mathcal{K} = \mathcal{K}_A = \{(X, D(\mathbf{A}), \{R_f\}_{f \in \Sigma_L}, \{R_g\}_{g \in \Sigma_A}) \mid (X, \{R_f\}_{f \in \Sigma_L}) \in RT_{\Sigma_L}$; $R_g \subseteq X^m \times D(\mathbf{A})$ *increasing for all* $g \in \Sigma_A$ *with arity* $m\}$.

(5) $\mathcal{V} = \mathsf{RDLO}^A_{\Sigma,\mathsf{Res}}$ *the subclass of all algebras in* DLO^A_Σ *in which the operators in* Σ_L *satisfy the residuation conditions in* Res;
$\mathcal{K}$ *the class of those spaces in* $\mathcal{K}_A$ *which satisfy:*
$\{R_f(x_1, \dots, x_n, x) \leftrightarrow R_g(x_1, \dots, x, \dots, x_n, x_i) \mid$ *"g i-residuation of f"* $\in \mathsf{Res}\}$.

(6) $\mathcal{V} = \mathsf{BAO}^A_\Sigma = \{(\mathbf{B}, \mathbf{A}, \{f\}_{f \in \Sigma_B}, \{g\}_{g \in \Sigma_A}) \mid \mathbf{B} \in \mathsf{Bool}$; $f : \mathbf{B}^k \to \mathbf{B}$ *k-ary join hemimorphism for all* $f \in \Sigma_B$; *and* $g : \mathbf{B}^m \to \mathbf{A}$ *m-ary join hemimorphism for all* $g \in \Sigma_A\}$;
$\mathcal{K}$ *the class of those spaces in* $\mathcal{K}_A$ *with the support of sort* lat *discretely ordered.*

(7) $\mathcal{V} = \mathsf{H}$, *the class of all Heyting algebras;* $\mathcal{K}$ *the family of all preordered spaces.*

For automated theorem proving it is important to find subclasses of RT^S_Σ with good theoretical and logical properties, for instance subclasses which are *first order definable.*

Theorem 4.12 [**Sof03**] *If $\mathcal{V}$ satisfies condition (K) then, for every formula $\phi = \forall x_1, \ldots, x_k(\bigwedge_{i=1}^n s_{i1} = s_{i2} \rightarrow \bigvee_{j=1}^m t_{j1} = t_{j2})$, $\mathcal{V} \models \phi$ if and only if for every $\mathbf{X} \in \mathcal{K}, \mathcal{O}(\mathbf{X}) \models \phi$.*

Theorem 4.12 allows to encode all terms occurring in ϕ as sets. The lattice operations can be encoded simply as set operations: suprema are unions and infima are intersections, and the additional operators are defined, using an inverse-image construction for the relations (cf. the definition of f_R in (11)). Let $ST(\phi)$ be the set of all subterms of s_{il} and t_{jp}, $1 \leq i \leq n, 1 \leq j \leq m, l, p \in \{1, 2\}$ (including the variables and s_{il}, t_{jp} themselves).

4.4.3.1 Distributive lattices and Boolean algebras with operators.

We first consider the situation when the only operations besides conjunction and disjunction are many-sorted hemimorphisms.

Proposition 4.13 *Let $\mathcal{K} \subseteq RT^S_\Sigma$. The following are equivalent:*

(1) *For every $\mathbf{X} \in \mathcal{K}$, $\mathcal{O}(\mathbf{X}) \models \phi$.*
(2) *For every $\mathbf{X} = (\{(X_s, \leq)\}_{s \in S}, \{R\}_{R \in \Sigma}) \in RT^S_\Sigma$ and every family of subsets of X indexed by all subterms of ϕ, $\{I_e \subseteq X_s \mid e \in ST(\phi)$ of sort $s \in S\}$, if:*

$$\left\{\begin{array}{llll}
(\mathsf{Dom}_s) & \mathbf{X} \in \mathcal{K}, & & \\
(\mathsf{Her}_s) & I_e \in \mathcal{O}(X_s) & & \forall e \in ST(\phi) \text{ of sort } s, \\
(\mathsf{Ren}_s) & (1,0) & I_{1_s} = X_s, \quad I_{0_s} = \emptyset, & \\
 & (\wedge) & I_{e_1 \wedge e_2} = I_{e_1} \cap I_{e_2}, & \\
 & (\vee) & I_{e_1 \vee e_2} = I_{e_1} \cup I_{e_2}, & \\
 & (\Sigma_{s_1 \ldots s_n \rightarrow s}) & I_{f(e_1, \ldots, e_n)} = R_f^{-1}(I_{e_1}, \ldots, I_{e_n}), & \\
(\mathsf{P}_s) & & I_{s_{i1}} = I_{s_{i2}} & \text{for } i = 1, \ldots, n,
\end{array}\right.$$

then

(C_s) *for some $j \in \{1, \ldots, m\}$ $I_{t_{j1}} = I_{t_{j2}}$,*

where the rules in (Σ) *range over all terms in $ST(\phi)$ starting with an operator in $\Sigma_{s_1 \ldots s_n \rightarrow s}$.* (We used the abbreviation $R^{-1}(U_1, \ldots, U_n) := \{x \mid \exists x_1 \in U_1 \ldots \exists x_n \in U_n : R(x_1, \ldots, x_n, x)\}$.)

If the class $\mathcal{K}$ is first order definable, Proposition 4.13 justifies a structure-preserving translation of universal formulae to sets of clauses.

Theorem 4.14 [**Sof02**] *Assume that $\mathcal{V}$ and $\mathcal{K}$ satisfy condition (K), where $\mathcal{K}$ is a class of RT Σ-structures definable by a finite set C of first order sentences. The following are equivalent:*

(1) $\mathcal{V} \models \phi$.
(2) *The conjunction of* $(\mathsf{Dom}) \cup (\mathsf{Her}) \cup (\mathsf{Ren}) \cup (\mathsf{P}) \cup (\mathsf{N_1}) \cup \cdots \cup (\mathsf{N_m})$ *is unsatisfiable, where:*

$$
\begin{array}{llrcll}
(\mathsf{Dom}) & & C, & & & \\
 & & \leq \subseteq X_s \times X_s & \text{is} & \textit{reflexive and transitive} & \textit{for every sort } s \in S, \\
 & & R_f \subseteq \prod_{i=1}^{n+1} X_{s_i} & \text{is} & \textit{increasing} & \textit{for } f \in \Sigma_{s_1 \ldots s_n \to s_{n+1}}, \\
(\mathsf{Her}) & \forall x, y & (x \leq y \wedge P_e(x) & \to & P_e(y)) & \\
(\mathsf{Ren}) & (1)\ \forall x & P_{1_s}(x) & & & \textit{for every sort } s \in S, \\
 & (0)\ \forall x & \neg P_{0_s}(x) & & & \textit{for every sort } s \in S, \\
 & (\wedge)\ \forall x & (P_{e_1 \wedge e_2}(x) & \leftrightarrow & P_{e_1}(x) \wedge P_{e_2}(x)) & \\
 & (\vee)\ \forall x & (P_{e_1 \vee e_2}(x) & \leftrightarrow & P_{e_1}(x) \vee P_{e_2}(x)) & \\
 & (\Sigma)\ \forall x & (P_{f(e_1, \ldots, e_n)}(x)^{\varepsilon} & \leftrightarrow & \exists x_1 \ldots x_n (\bigwedge_{i=1}^{n} P_{e_i}(x_i)^{\varepsilon_i} \wedge R_f(x_1, \ldots, x_n, x))) & \\
(\mathsf{P}) & \forall x & (\bigwedge_{i=1}^{n} P_{s_{i1}}(x) & \leftrightarrow & P_{s_{i2}}(x)) & \\
(\mathsf{N}_1) & \exists x_1 & (P_{t_{11}}(x_1) & \not\leftrightarrow & P_{t_{12}}(x_1)) & \\
 & \ldots & & \ldots & & \\
(\mathsf{N}_m) & \exists x_m & (P_{t_{m1}}(x_m) & \not\leftrightarrow & P_{t_{m2}}(x_m)) &
\end{array}
$$

where the unary predicates P_e *are indexed by elements in* $ST(\phi)$*, and the formulae in* Σ *range over all operators* $f \in \Sigma$ *such that* f *is a join hemimorphism of type* $s_1^{\varepsilon_1} \ldots s_n^{\varepsilon_n} \to s^{\varepsilon}$*, where* $\varepsilon_i, \varepsilon \in \{-1, +1\}$*, and* $L^{+1} := L$ *and* $L^{-1} := \neg L$.

Similar ideas can be used for obtaining translations to clause form for formulae of the form $\bigwedge_{i=1}^{n} s_{i1} {\leq} s_{i2} \to \bigwedge_{j=1}^{m} t_{j1} {\leq} t_{j2}$. Then only the direct implications are necessary in (P) and (N).

If $\mathcal{V} = \mathsf{RDLO}^{S}_{\Sigma,\mathsf{Res}}$ or $\mathcal{V} = \mathsf{RDLO}^{A}_{\Sigma,\mathsf{Res}}$, where Res is a set of generalized residuation rules, then the set C of formulae contains the conditions:

$$R_f(x_1, \ldots, x_i, \ldots, x_n, x) \leftrightarrow R_g(x_1, \ldots, x, \ldots, x_n, x_i)$$

for all n-ary operators f, g such that "g is an i-residuation of f" $\in \mathsf{Res}$.

4.4.3.2 Heyting algebras. If $\mathcal{V} = \mathsf{H}$, the class of Heyting algebras, then the set C contains only the preorder axioms for the relation $\leq$. In this case Ren contains additional rules for the Heyting implication and negation, namely:

$$
\begin{array}{ll}
(\mathsf{Ren})\ (\Rightarrow)\ \forall x & (P_{e_1 \Rightarrow e_2}(x) \leftrightarrow \forall y (x \leq y \wedge P_{e_1}(y) \to P_{e_2}(y))), \\
\qquad\quad (\neg)\ \ \forall x & (P_{\neg e}(x) \leftrightarrow \forall y (x \leq y \to \neg P_e(y))).
\end{array}
$$

4.4.4 Some decidability results

We now present some examples in which decidability results can be obtained.

Theorem 4.15 [Sof02] *Ordered resolution with selection decides, in time* exponential in the size of the input *if the arity of operators in* Σ *has an upper bound, and* exponential in the square of the size of the input *in general, the universal clause theory of* DLO^S_Σ*, and* RDLO^S_Σ*.*

Proof: (Idea) The results of [Sof03], Section 5.1 can easily be adapted to prove this theorem. As pointed out in [Sof03], the selection strategy we adopt for this purpose shows that in this case inferences with the clauses containing the $\leq$ symbol are not needed for refutational completeness. $\square$

It can easily be seen that for uniform word problems which contain only conjunction and join hemimorphisms, resolution yields a polynomial time decision procedure [7] .

Theorem 4.16 [Sof02] *Ordered resolution with selection decides, in time* exponential in the size of the input *if the arity of operators in* Σ *has an upper bound, and* exponential in the square of the size of the input *in general, the universal clause theory of* DLO^A_Σ*, and* RDLO^A_Σ*, where* A *is a finite distributive lattice.*

Proof: (Idea) We can show that inferences with the clauses containing the $\leq$ symbol applied to arguments of sort lat are not needed in the case of DLO^A_Σ. Since $D(A)$ is finite, the monotonicity and heredity rules for sort a, can be replaced with their instances with elements in $D(A)$. For instance the monotonicity and heredity rules can alternatively be expressed by:

$$R_f(x_1, \dots, x_n, a) \rightarrow R_f(x_1, \dots, x_n, b) \qquad \text{for all } a, b \in D(A), a \leq b \quad (12)$$

$$P_e(a) \rightarrow P_e(b) \qquad \text{for all } a, b \in D(A), a \leq b \quad (13)$$

We can now introduce $D(A)$ copies for every predicate symbol with last argument of sort a, e.g. by replacing, for every $a \in D(A)$, $R_f(x_1, \dots, x_n, a)$ with $R^a_f(x_1, \dots, x_n)$ and $P_e(a)$ with P^a_e. Arguments in [Sof03], Section 5.1 can now be applied and also in this case yield the desired complexity results. $\square$

Similar arguments can be also used for (many sorted) Boolean algebras with operators, by considering, in addition, the renaming rules for Boolean negation. Also in this case we obtain exponential time resolution-based decision procedures, i.e. decision procedures with optimal time complexity.

[7] Y. Kazakov, personal communication

The renaming rules $\mathsf{Ren}(\Rightarrow, \neg)$ for Heyting algebras contain the predicate $\leq$, and in this situation inferences with the clauses containing the $\leq$ symbol are needed. $\leq$ is a preorder, i.e. reflexive and transitive, so we can use ordered chaining with selection for checking the satisfiability of the conjunction of formulae in $(\mathsf{Dom}) \cup (\mathsf{Her}) \cup (\mathsf{Ren}) \cup (\mathsf{P}) \cup (\mathsf{N})$.

Theorem 4.17 [**Sof03**] *For every formula* $\phi = \forall x_1, \ldots, x_k(\bigwedge_{i=1}^n s_{i1} = s_{i2} \rightarrow \bigvee_{j=1}^m t_{j1} = t_{j2})$*, ordered chaining with eager condensation and selection decides (in at most doubly exponential time and exponential space with respect to the length of* ϕ*) whether* $\mathsf{H} \models \phi$*.*

4.4.5 A special case: Finitely-valued logics based on DLO.

The results above can be applied to automated theorem proving in propositional many-valued logics based on finite distributive lattices with operators. Let $\mathbf{A} = (A, \vee, \wedge, 0, 1, \{f_A\}_{f \in \Sigma})$ be a finite distributive lattice with operators, and let $D(\mathbf{A})$ be the Priestley dual of $\mathbf{A}$. Since $\mathbf{A}$ is finite, $D(\mathbf{A}) = (\{\uparrow j \mid j \in A, \text{ join irreducible}\}, \subseteq)$, and $\mathbf{A}$ is isomorphic to $\mathcal{O}(D(\mathbf{A}))$. In this case $\mathcal{V} = \{\mathbf{A}\}$ and $\mathcal{K} = \{D(\mathbf{A})\}$ satisfy condition (K). Let $\phi := \forall x_1, \ldots, x_k(\bigwedge_{i=1}^n s_{i1} = s_{i2} \rightarrow \bigvee_{j=1}^m t_{j1} = t_{j2})$ be a formula in the signature of $\mathbf{A}$:

Corollary 4.18 *Let* $\mathbf{A} = (A, \vee, \wedge, 0, 1, \{f_A\}_{f \in \Sigma})$ *be a finite distributive lattice with operators. The following are equivalent:*

(1) $\mathbf{A} \models \phi$.
(2) *The conjunction of* $(\mathsf{Dom}) \cup (\mathsf{Her}) \cup (\mathsf{Ren}) \cup (\mathsf{P}) \cup (\mathsf{N_1}) \cup \cdots \cup (\mathsf{N_m})$ *is unsatisfiable, where:*

(Dom) $\forall x(x = \uparrow j_1 \vee \ldots \vee x = \uparrow j_k)$ *where* $D(A) = (\{\uparrow j_1, \ldots, \uparrow j_k\}, \subseteq)$

$\uparrow j_i \leq \uparrow j_k$; $R_f(\uparrow j_{i_1}, \ldots, \uparrow j_{i_l}, \uparrow j_{i_{l+1}})$ *whenever it holds in* $D(A)$

(Her) $\forall x, y \quad (x \leq y \wedge P_e(x) \rightarrow P_e(y))$

(Ren) $(1, 0) \quad \forall x P_1(x) \quad \forall x \neg P_0(x)$

$(\wedge) \forall x \quad (P_{e_1 \wedge e_2}(x) \leftrightarrow P_{e_1}(x) \wedge P_{e_2}(x))$

$(\vee) \forall x \quad (P_{e_1 \vee e_2}(x) \leftrightarrow P_{e_1}(x) \vee P_{e_2}(x))$

$(\Sigma) \forall x (P_{f(e_1, \ldots, e_n)}(x)^{\varepsilon} \leftrightarrow \exists x_1 \ldots x_n (\bigwedge_{i=1}^n P_{e_i}(x_i)^{\varepsilon_i} \wedge R_f(x_1, \ldots, x_n, x)))$

(P) $\forall x \quad (\bigwedge_{i=1}^n P_{s_{i1}}(x) \leftrightarrow P_{s_{i2}}(x))$

(N_1) $\exists x_1 \quad (P_{t_{11}}(x_1) \not\leftrightarrow P_{t_{12}}(x_1))$

$\cdots \quad \cdots$

(N_m) $\exists x_m \quad (P_{t_{m1}}(x_m) \not\leftrightarrow P_{t_{m2}}(x_m))$

where the unary predicates P_e *are indexed by elements in* $ST(\phi)$*, and the for-*

mulae in Σ range over all operators $f \in \Sigma$ such that f is a join hemimorphism of type $\varepsilon_1 \dots \varepsilon_n \to \varepsilon$, where $\varepsilon_i, \varepsilon \in \{-1, +1\}$, and $L^{+1} := L$ and $L^{-1} := \neg L$.

It is easy to see that the conjunction above is unsatisfiable if and only if the set of all its ground instances, where the variables are instantiated with elements in $D(A)$, is satisfiable. We thus recover some of the results in [Sof01]. (The labeled literals of the form $\boxed{\uparrow j}\, P_e$ used in [Sof01] correspond to ground literals of the form $P_e(\uparrow j)$ in the present setting.)

For an extension to automated theorem proving in first order many-valued logics based on distributive lattices with operators, where the truth functions for quantifiers are defined by $Q_\forall = \mathsf{inf}$ and $Q_\exists = \mathsf{sup}$, we refer to [Sof01], where, in addition, we illustrate on several examples the effectivity of the method, in particular the fact that using the Priestley dual of the algebra of truth values instead of the algebra of truth values itself helps to drastically reduce the number of generated clauses.

5 Conclusions

The main goal of this paper was to show that, in many situations, efficient methods for automated theorem proving can be obtained if we can find suitable embeddings into first order classical logic. We illustrated the ideas by means of various examples, ranging from many-valued logics to description logics.

In the case of many-valued logics, such embeddings into classical logic allow to reconstruct known completeness results for existing methods for automated theorem proving. Apart from this, the inference systems we obtain are much more restricted, in particular with ordering constraints and selection functions. In addition, general results in first order logic for simplification and for eliminating redundancies can then be used for free in the derived calculi. Both in many-valued logics and in more general logics, such as modal logic, intuitionistic logic and generalizations thereof, such embeddings into classical logic allow to use existing efficient theorem provers for first order logic; there is no need to devise specialized theorem provers for particular non-classical logics. In many cases – for instance in the case of the uniform word problem for Boolean algebras with operators, or for Heyting algebras – the complexity of the decision procedures obtained this way is in EXPTIME, hence (time-)optimal.

References

[AB75] A.R. Anderson and N.D. Belnap. *Entailment: The Logic of Relevance and Necessity*, volume 1. Princeton University Press, 1975.

[ABN98] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[AC00] S. Aguzzoli and A. Ciabattoni. Finiteness of infinite-valued Łukasiewicz logic. *Journal of Logic, Language and Information*, 9:5–29, 2000.

[ANS01] H. Andréka, I. Németi, and I. Sain. Algebraic logic. In *Handbook of Philosophical Logic*, volume 2. Kluwer, Dordrecht, 2nd edition edition, 2001.

[Baa03a] F. Baader. Terminological cycles in a description logic with existential restrictions. In G. Gottlob and T. Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence, (IJCAR 2003)*, pages 325–330. Morgan Kaufmann, 2003.

[Baa03b] F. Baader. The instance problem and the most specific concept in the description logic $\mathcal{EL}$ w.r.t. terminological cycles with descriptive semantics. In A. Günter, R. Kruse and B. Neumann, editors, *Proceedings of the 26th Annual German Conference on Artificial Intelligence, (KI 2003), LNAI 2821*, pages 64–78. Springer, 2003.

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

[BFC01] M. Baaz, C.G. Fermüller, and A. Ciabattoni. Herbrand's theorem for prenex Gödel logic and its consequences for theorem proving. In R. Nieuwenhuis and A. Voronkov, editors, *Proceedings of LPAR'2001, LNCS 2250*, pages 201–215. Springer, 2001.

[BFS99] M. Baaz, C.G. Fermüller, and G. Salzer. Automated deduction for many-valued logics. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, pages 1355–1402. Elsevier Science 2001.

[BG94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BG98] L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6):1007–1049, 1998.

[BHM99] B. Beckert, R. Hähnle, and F. Manyà. Transformations between signed and classical clause logic. In *Proceedings of the 29th International Symposium on Multiple-valued Logic (ISMVL'99), Freiburg, Germany*, pages 248–255. IEEE Press, 1999.

[BHM00] B. Beckert, , R. Hähnle, and F. Manyà. The 2-SAT problem of regular signed CNF formulas. In *Proceedings of the 30th Interna-*

tional Symposium on Multiple-valued Logic (ISMVL'2000), Portland, USA, pages 331–336. IEEE Press, 2000.

[BHM01] R. Béjar, R. Hähnle, and F. Manyà. A modular reduction of regular logic to classical logic. In *Proceedings if the 31st International Symposium on Multiple-valued Logic (ISMVL'01), Warsaw, Poland*, pages 221–226. IEEE Press, 2001.

[BRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

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

[Bur95] S. Burris. Polynomial time uniform word problems. *Mathematical Logic Quaterly*, 41:173–182, 1995.

[CZ90] R. Caferra and N. Zabel. An application of many-valued logic to decide propositional $S5$ formulae: A strategy designed for a parametric tableaux-based theorem prover. In *Artificial Intelligence IV: Methodology, Systems, Applications*, pages 23–32, 1990. Elsevier.

[DCF02] M. Dezani-Ciangaglini, A. Frisch, E. Giovannetti, and Y. Motohama. The relevance of semantic subtyping. *Electronic Notes in Theoretical Computer Science*, 70(1), 2002.

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

[Dum59] M. Dummet. A propositional calculus with denumerable matrix. *Journal of Symbolic Logic*, 24(2):97–106, 1959.

[Dun95] J.M. Dunn. Positive modal logic. *Studia Logica*, 55:301–317, 1995.

[GHM01] H. Ganzinger, U. Hustadt, C. Meyer, and R.A. Schmidt. A resolution-based decision procedure for extensions of K4. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*, volume 119 of *CSLI Lecture Notes*, chapter 9, pages 225–246. CSLI, Stanford, USA, 2001.

[GN99] H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proceedings of the Fourteenth Annual IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 295–303. IEEE Computer Society Press, 1999.

[Gol93] R. Goldblatt. *Mathematics of modality*, volume 43 of *Center for the Study of Language and Information*. Univ. of Chicago Press, 1993.

[GS00] H. Ganzinger and V. Sofronie-Stokkermans. Chaining techniques for automated theorem proving in many-valued logics. In *Proceedings of the 30th International Symposium on Multiple-valued Logic (ISMVL'2000), Portland, USA*, pages 337–344. IEEE Computer Society Press, 2000.

[Häh93] R. Hähnle. *Automated Theorem Proving in Multiple-Valued Logics*. Oxford University Press, 1993.

[Häh94a] R. Hähnle. Many-valued logic and mixed integer programming. *Annals of Mathematics and Artificial Intelligence*, 12(3,4):231–264, December 1994.

[Häh94b] R. Hähnle. Short conjunctive normal forms in finitely valued logics. *Journal of Logic and Computation*, 4(6):905–927, 1994.

[Häh96] R. Hähnle. Exploiting data dependencies in many-valued logics. *Journal of Applied Non-Classical Logics*, 6(1):49–69, 1996.

[Häh97a] R. Hähnle. Proof theory of many-valued logic – linear optimization – logic design. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 1(3):107–119, 1997.

[Häh97b] R. Hähnle. Proof theory of many-valued logic—linear optimization—logic design: Connections and interactions. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 1(3):107–119, September 1997.

[Häh01] R. Hähnle. Advanced many-valued logics. In *Handbook of Philosophical Logic*, volume 2, pages 297–395. Kluwer, Dordrecht, 2nd edition edition, 2001.

[Häh03] R. Hähnle. Complexity of many-valued logics. In M. Fitting and E. Orłowska, editors, *Beyond Two: Theory and Applications of Multiple Valued Logic*, volume 114 of *Studies in Fuzziness and Soft Computing*, chapter 3, pages 211–233. Springer, Berlin, 2003.

[Háj98] P. Hájek. *Metamathematics of Fuzzy Logic.* Trends in Logic, Volume 4. Kluwer Academic Publishers, Dordrecht, 1998.

[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.

[IO01] S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *Proceedings of 28th Symposium on Principles of Programming Languages (POPL'01)*, pages 14–26. ACM, 2001.

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

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

[KN03] Y. Kazakov and H. de Nivelle. Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete. In D. Calvanese, G. De Giacomo, and E. Franconi, editors, *2003 International Workshop on Description Logics (DL-03)*, volume 81 of *CEUR Workshop Proceedings*, pages 56–64, Rome, Italy, September 2003. University of Rome "La Sapienza" and Free University of Bolzano/Bozen, CEUR.

[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.

[Łuk30] J. Łukasiewicz. Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. *Comptes rendus de la Société des Sciences et Lettres de Varsovie*, cl.iii, 23:51–77, 1930.

[Man00] F. Manyà. The 2-SAT problem in signed CNF formulas. *Multiple-*

Valued Logic. An International Journal, 5, 2000.

[MGK02] D. McAllester, R. Givan, D. Kozen, and C. Witty. Tarskian set constraints. *Information and Computation*, 174:105–131, 2002.

[MO98] D. Mundici and N. Olivetti. Resolution and model building in the infinite-valued calculus of Łukasiewicz. *Theoretical Computer Science*, 200:335–366, 1998.

[Mun87] D. Mundici. Satisfiability of many-valued sentential logics is NP-complete. *Theoretical Computer Science*, 52:145–153, 1987.

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

[OK85] H. Ono and Y. Komori. Logics without the contraction rule. *Journal of Symbolic Logic*, 50:169–201, 1985.

[Ono93] H. Ono. Semantics for substructural logics. In P. Schroeder-Heister and K. Došen, editors, *Substructural Logics*, pages 259–291. Oxford University Press, 1993.

[Pym02] D.J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*, volume 26 of *Applied Logics Series*. Kluwer, 2002.

[Ras74] H. Rasiowa. *An Algebraic Approach to Non-Classical Logics*, volume 78 of *Studies in Logic and the Foundations of Mathematics*. North Holland, Amsterdam, 1974.

[Sch99] R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.

[Sof01] V. Sofronie-Stokkermans. Automated theorem proving by resolution for finitely-valued logics based on distributive lattices with operators. *Multiple-Valued Logic - An International Journal*, 6(3/4):289–344, 2001.

[Sof02] V. Sofronie-Stokkermans. On uniform word problems involving bridging operators on distributive lattices. In U. Egly and Ch. Fermüller, editors, *Proceedings of TABLEAUX 2002. LNAI 2381*, pages 235–250, Copenhagen, Denmark, 2002. Springer.

[Sof03] V. Sofronie-Stokkermans. Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators. *Journal of Symbolic Computation*, 36(6): 891–924, 2003.

[Urq86] A. Urquhart. Many-valued logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic Vol III*, pages 71–116. D. Reidel Publishing Company, 1986.

[Urq96] A. Urquhart. Duality for algebras of relevant logics. *Studia Logica*, 56(1,2):263–276, 1996.

Submitted to the Special Issue for JIM'03: October 10, 2003; Revised: January 7, 2005.

To appear in Annals of Mathematics and Artificial Intelligence, 2007.

A Appendix: Proof of Theorem 3.4

Proposition A.1 *For all concept descriptions C_1, C_2, and every TBox $\mathcal{T}$, $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if* $\mathsf{BAO}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \rightarrow \overline{C_1} \leq \overline{C_2}$.

Proof: This follows from the fact that every algebra in BAO_{N_R} homomorphically embeds into a Boolean algebra of sets. $\Box$

Lemma A.2 *Every semilattice* $\mathbf{S} \in \mathsf{SLO}^{\exists}_{N_R}$ *embeds into a lattice in* $\mathsf{DLO}^{\exists}_{N_R}$.

Proof: Let $\mathbf{S} = (S, \wedge, 0, 1, \{f_S\}_{f \in \Sigma})$ be a semilattice with 0, 1, and with monotone operators in Σ. Let $\mathcal{OI}^*(\mathbf{S}) = (\mathcal{OI}^*(\mathbf{S}), \cap, \cup, \{0\}, S, \{\overline{f}_S\}_{f \in \Sigma})$ be the lattice of all non-empty order-ideals of S, where join is set union, meet is set intersection, and the additional operators in Σ are defined, for every non-empty order ideal of S, U, by $\overline{f}_S(U) = \downarrow f_S(U)$. It is easy to see that for every $f \in \Sigma$, $\overline{f}_S(\{0\}) = \downarrow f_S(\{0\}) = \{0\}$ and, for every $U_1, U_2 \in \mathcal{OI}^*(\mathbf{S})$, $\overline{f}_S(U_1 \cup U_2) = \overline{f}_S(U_1) \cup \overline{f}_S(U_2)$. Obviously, $(\mathcal{OI}^*(\mathbf{S}), \cap, \cup, \{0\}, S)$ is a bounded distributive lattice. Thus, $\mathcal{OI}(\mathbf{S}) \in \mathsf{DLO}^{\exists}_{N_R}$.

Let $\eta : \mathbf{S} \rightarrow \mathcal{OI}^*(\mathbf{S})$ defined by $\eta(x) := \downarrow x$. Obviously, η is an order embedding (and, hence, injective), $\eta(0) = \{0\}, \eta(1) = S$ and $\eta(x \wedge y) = \downarrow(x \wedge y) = \downarrow x \cap \downarrow y$. We show that $\eta(f_S(x)) = \downarrow f_S(x) = \overline{f}_S(\downarrow x)$. If $y \in \downarrow f_S(x)$ then $y \leq f_S(x)$, so $y \in \downarrow\{f_S(\downarrow x)$. Conversely, if $y \in \downarrow\{f_S(\downarrow x)$ then $y \leq f_S(z)$ for some $z \leq x$, hence, by the monotonicity of f_S, $y \leq f_S(x)$.) Thus, η is a homomorphism with respect to the whole signature of $\mathbf{S}$. $\Box$

Lemma A.3 *Every semilattice* $\mathbf{S} \in \mathsf{SLO}^{\forall}_{N_R}$ *embeds into a lattice in* $\mathsf{DLO}^{\forall}_{N_R}$.

Proof: Let $\mathbf{S} = (S, \wedge, 1, \{f_S\}_{f \in \Sigma})$ be a semilattice with 1, such that f_S is a unary meet homomorphism for every $f \in \Sigma$. Consider the lattice of all order-ideals of S, $\mathcal{OI}(\mathbf{S}) = (\mathcal{OI}(\mathbf{S}), \cap, \cup, \emptyset, S, \{\overline{f}_S\}_{f \in \Sigma})$, where join is set union, meet is set intersection, and the additional operators in Σ are defined, for every non-empty order ideal of S, U, by $\overline{f}_S(U) = \downarrow f_S(U)$. It is easy to see that for every $f \in \Sigma$, $\overline{f}_S(S) = \downarrow f_S(S) = S$ (since $f_S(1) = 1$). We show that if f is a meet hemimorphism then for every $U_1, U_2 \in \mathcal{OI}(\mathbf{S})$, $\overline{f}_S(U_1 \cap U_2) = \overline{f}_S(U_1) \cap \overline{f}_S(U_2)$. The direct inclusion is obvious. In order to prove the converse inclusion, let $x \in \overline{f}_S(U_1) \cap \overline{f}_S(U_2)$. Then there exist $y_1 \in U_1$ and $y_2 \in U_2$ such that $x \leq f_S(y_1)$ and $x \leq f_S(y_2)$ Then $x \leq f_S(y_1) \wedge f_S(y_2) = f_S(y_1 \wedge y_2)$ (since f_S is a meet hemimorphism). Let $y = y_1 \wedge y_2$. Then $y \leq y_i$ for $i = 1, 2$, so $y \in U_1 \cap U_2$. This shows that $x \leq f_S(y)$, with $y \in U_1 \cap U_2$, so $x \in \overline{f}_S(U_1 \cap U_2)$.
The fact that $\eta : \mathbf{S} \rightarrow \mathcal{OI}(\mathbf{S})$ defined by $\eta(x) := \downarrow x$ is an order-embedding and a homomorphism with respect to the whole signature of $\mathbf{S}$ can be proved as before. $\Box$

Lemma A.4 *Every bounded distributive lattice with operators in Σ homomorphically embeds into a Boolean lattice with (the same type of) operators in Σ.*

Proof: Consequence of results of Priestley duality for distributive lattices and Stone duality for Boolean algebras. $\square$

Proposition A.5 *Assume that the only concept constructors are intersection and existential restriction. For all concept descriptions C_1, C_2, and every TBox $\mathcal{T}$, $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if $\mathsf{SLO}^{\exists}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \rightarrow \overline{C_1} \leq \overline{C_2}$.*

Proof: ($\Rightarrow$) Assume that $C_1 \sqsubseteq_{\mathcal{T}} C_2$. Then we know, by Proposition A.1, that $\mathsf{BAO}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \rightarrow \overline{C_1} \leq \overline{C_2}$. Let $\mathbf{S} = (S, \wedge, 0, 1, \{f_{\exists R}\}_{R \in N_R}) \in SLO^{\exists}_{N_R}$. Then there exists an injective homomorphism of bounded semilattices, $\eta : \mathbf{S} \rightarrow \mathcal{O}I^*(\mathbf{S})$, and a homomorphic embedding h of the bounded lattice with operators $\mathcal{O}I^*(\mathbf{S})$ into a Boolean lattice with operators $\mathbf{B}$. Let $v : N_C \rightarrow \mathbf{S}$ be an arbitrary valuation in $\mathbf{S}$ such that $v(A) = \overline{v}(C)$ for every $A \equiv C \in \mathcal{T}$. Then $h \circ \eta \circ v : N_C \rightarrow \mathbf{B}$ is an assignment into an algebra in BAO_{N_R} with $h(\eta(v(A))) = \overline{h \circ \eta \circ v}(C)$ for every $A \equiv C \in \mathcal{T}$. So, $h(\eta(\overline{v}(C_1 \wedge C_2))) = \overline{h \circ \eta \circ v}(C_1 \wedge C_2) = \overline{h \circ \eta \circ v}(C_1) = h(\eta(\overline{v}(C_1)))$, so, by the injectivity of $h \circ \eta$, $\overline{v}(C_1 \wedge C_2) = \overline{v}(C_1)$.
($\Leftarrow$) The converse follows immediately from the fact that the reduct of every algebra in BAO_{N_R} to the signature $\{\wedge, 0, 1, \{f_{\exists R}\}_{R \in N_R}\}$ is in $SLO^{\exists}_{N_R}$. $\square$

Proposition A.6 *Assume that the only concept constructors are intersection and universal restriction. For all concept descriptions C_1, C_2, and every TBox $\mathcal{T}$, $C_1 \sqsubseteq_{\mathcal{T}} C_2$ if and only if $\mathsf{SLO}^{\forall}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \rightarrow \overline{C_1} \leq \overline{C_2}$.*

Proof: ($\Rightarrow$) Assume that $C_1 \sqsubseteq_{\mathcal{T}} C_2$. Then we know, by Proposition A.1, that $\mathsf{BAO}_{N_R} \models \left(\bigwedge_{A \equiv C \in \mathcal{T}} A = \overline{C}\right) \rightarrow \overline{C_1} \leq \overline{C_2}$. Let $\mathbf{S} = (S, \wedge, 1, \{f_{\forall R}\}_{R \in N_R}) \in SLO^{\forall}_{N_R}$. Then there exists an injective homomorphism of bounded semilattices, $\eta : \mathbf{S} \rightarrow \mathcal{O}I(\mathbf{S})$, and a homomorphic embedding h of the bounded lattice with operators $\mathcal{O}I(\mathbf{S})$ into a Boolean lattice with operators $\mathbf{B}$. Let $v : N_C \rightarrow \mathbf{S}$ be an arbitrary valuation in $\mathbf{S}$ such that $v(A) = \overline{v}(C)$ for every $A \equiv C \in \mathcal{T}$. Then $h \circ \eta \circ v : N_C \rightarrow \mathbf{B}$ is an assignment into an algebra in BAO_{N_R} with $h(\eta(v(A))) = \overline{h \circ \eta \circ v}(C)$ for every $A \equiv C \in \mathcal{T}$. So, $h(\eta(\overline{v}(C_1 \wedge C_2))) = \overline{h \circ \eta \circ v}(C_1 \wedge C_2) = \overline{h \circ \eta \circ v}(C_1) = h(\eta(\overline{v}(C_1)))$, so, by the injectivity of $h \circ \eta$, $\overline{v}(C_1 \wedge C_2) = \overline{v}(C_1)$.
($\Leftarrow$) The converse follows immediately from the fact that the reduct of every algebra in BAO_{N_R} to the signature $\{\wedge, 1, \{f_{\forall R}\}_{R \in N_R}\}$ is in $SLO^{\forall}_{N_R}$. $\square$

B Appendix: Proof of Theorem 3.5

Theorem B.1 *The uniform word problem for BAO_{N_R} is* EXPTIME*-complete.*

Proof: (Sketch) A resolution-based exponential time algorithm for the uniform word problem for BAO_{N_R} is obtained e.g. in [Sof03]. EXPTIME-hardness (even for word problems which only contain conjunction and universal and existential restriction) can be proved using arguments similar to those used in

[MGK02], Theorem 1. □

Lemma B.2 *Every partial* $\mathsf{SLO}^{\exists}_{N_R}$*-algebra weakly embeds into a total* $\mathsf{SLO}^{\exists}_{N_R}$*-algebra.*

Proof: Let $\mathbf{P} = (P, \wedge, 0, 1, \{f_{\exists R}\}_{R \in N_R})$ be a partial $\mathsf{SLO}^{\exists}_{N_R}$-algebra. Then:

- $\wedge$ is a partially defined binary operation and for every $R \in N_R$, $f_{\exists R}$ is a partially defined unary operation;
- $x \wedge x$ is defined in P for every $x \in P$;
  $x \wedge y$ is defined in P if and only if $y \wedge x$ is defined in P and then they are equal;
  if $x \wedge y$ is defined in P and $x \wedge (y \wedge z)$ is defined in P then also $(x \wedge y) \wedge z$ is defined in P and $x \wedge (y \wedge z) = (x \wedge y) \wedge z$;
- $f_{\exists R}(0)$ is defined in P and equal to 0 for every $R \in N_R$;
  if $x \wedge y$ is defined in P and $x \wedge y = x$, and $f_{\exists R}(x), f_{\exists R}(y)$ are defined in P then $f_{\exists R}(x) \wedge f_{\exists R}(y)$ is defined in P and equals $f_{\exists R}(x)$, for every $R \in N_R$.

A partial order $\leq$ can be defined by $x \leq y$ if and only if $x \wedge y$ is defined in P and equals x. Let $\mathcal{OI}(\mathbf{P}) := (\mathcal{OI}(P, \leq), \cap, \{0\}, S, \{\overline{f}_{\exists R}\}_{R \in N_R})$, where union is join, intersection is meet, and the additional operators are defined, for every order ideal of S, U, by $\overline{f}_{\exists R}(U) = \downarrow\{f_{\exists R}(x) \mid f_{\exists R}(x) \text{ defined in } P, x \in U\}$. It is easy to see that $\overline{f}_{\exists R}(\{0\}) = \downarrow\{f_{\exists R}(0)\} = \{0\}$; and $\overline{f}_{\exists R}$ is monotone for every $R \in N_R$.

Let $\eta : \mathbf{P} \to \mathcal{OI}(\mathbf{P})$ be defined by $\eta(x) = \downarrow x$. η is obviously injective. We show that η is a weak embedding, i.e., in addition, whenever $f_P(p_1, \dots, p_n)$ is defined in $\mathbf{P}$, $\eta(f_P(p_1, \dots, p_n)) = \overline{f}(\eta(p_1), \dots, \eta(p_n))$.

Obviously, $\eta(1) = S$, $\eta(0) = \{0\}$, and whenever $x \wedge y$ is defined in P, $\eta(x \wedge y) = \eta(x) \cap \eta(y)$. Assume that $f_{\exists R}(x)$ is defined in P. We prove that $\eta(f_{\exists R}(x)) = \downarrow f_{\exists R}(x) = \downarrow\{f_{\exists R}(y) \mid f_{\exists R}(y) \text{ defined in } P, y \leq x\} = \overline{f}_{\exists R}(\downarrow x) = \overline{f}_{\exists R}(\eta(x))$.

If $y \in \eta(f_{\exists R}(x))$ then $y \leq f_{\exists R}(x)$, so $y \in \overline{f}_{\exists R}(\downarrow x)$. If $y \in \overline{f}_{\exists R}(\downarrow x)$ then $y \leq f_{\exists R}(z)$ for some z such that $f_{\exists R}(z)$ is defined and $z \leq x$ (i.e. such that $z \wedge x$ is defined in P and equals z). But then $f_{\exists R}(z) \wedge f_{\exists R}(x)$ is defined in P and equal to $f_{\exists R}(z)$, so $y \leq f_{\exists R}(z) \leq f_{\exists R}(x)$. Hence $y \in \eta(f_{\exists R}(x))$. □

Proposition B.3 *The uniform word problem for* $\mathsf{SLO}^{\exists}_{N_R}$ *is decidable in polynomial time.*

Proof: By a result of Burris [Bur95], a quasivariety $\mathcal{K}$ has a polynomial time decidable uniform word problem if every finite partial algebra which weakly satisfies the (quasi-)identities of $\mathcal{K}$ weakly embeds into a total algebra in $\mathcal{K}$. Lemma B.2 shows that this is the case for $\mathcal{K} = \mathsf{SLO}^{\exists}_{N_R}$. □