Chapter X: Computational Complexity of Propositional Fuzzy Logics

ZUZANA HANIKOVÁ

1 Introduction

This chapter is about computational complexity of decision problems in propositional fuzzy logics and also in algebras which constitute their algebraic semantics. We investigate sets of formulas and relations thereon, with an aim to determine their complexity by ranking them alongside well-known decision problems, such as SAT and TAUT in classical propositional logic. A key problem is, for a given logic, to determine the complexity of the set of its theorems and of the relation of provability of a formula from a finite theory. We rely on completeness theorems and work in a suitably chosen class of algebras, so we are also interested in complexity of appropriate fragments of the algebraic theory. Owing to the multitude of fuzzy logics under investigation, the general framework yields many particular problems, some of which are open.

Naturally, many patterns of thinking familiar from classical logic are not applicable in the many-valued case. For example, in classical propositional logic, one can reduce the problem of provability from finite theories to the problem of theoremhood, using the deduction theorem. The classical deduction theorem is however not generally available in fuzzy logic, and that is why the provability relation is, in general, an interesting complexity problem. To give another example, the duality of satisfiability and tautologousness, known from classical logic and occasioned by its dichotomy, is not valid for algebras corresponding to fuzzy logics. The fact that not only the classical dichotomy is absent, but there are typically infinitely many truth values, makes it actually nontrivial to find upper bounds on complexity of sets of formulas such as SAT. Indeed, a major part of our efforts in this chapter will be targeted to showing, for various existential problems, that if there is a solution, there is a succinct one.

On the other hand, all decision problems considered in this chapter share common lower bounds (not necessarily tight): for each consistent axiomatic extension of the logic $\mathrm{FL_{ew}}$, the SAT problem for the corresponding class of algebras is $\mathbf{NP}$-hard, whereas the TAUT problem is $\mathbf{coNP}$-hard. The word 'hard' is ominous here: while the problems are algorithmically solvable, this chapter does a poor job on attempting to solve them. Rather, it is intent on *classifying* the problems, using polynomial equivalence; we are not concerned about polynomial differences in performance. Throughout, we investigate the *worst-case complexity* of problems, an approach that is preferable for its elegance and robustness as long as one is aware of its limitations.

There is a pattern in results presented in this chapter: for those decision problems whose complexity has been settled (the problems have been proved complete in some complexity class), the situation is analogous to the classical case: satisfiability is $\mathbf{NP}$-complete, while tautologousness and consequence (hence, theoremhood and provability) are $\mathbf{coNP}$-complete. One might ask why consequence relation comes out no more difficult than tautologousness. This chapter tries to answer this question by showing $\mathbf{coNP}$-containment (hence, $\mathbf{coNP}$-completeness) for the universal fragment of the theory of these algebras. Thus we are able to avail ourselves of the classical dichotomy after all, albeit on a metamathematical level: the universal fragment of the theory is $\mathbf{coNP}$-complete if and only if the existential fragment is $\mathbf{NP}$-complete. SAT can be viewed as a fragment of the existential theory and TAUT and CONS as fragments of the universal theory, and that is why complexity results come out as rather flat. It is of course a major question whether this might be the case for those problems that are, so far, open.

Complexity-wise, as well as otherwise, a territory well conquered is propositional Hájek's BL and its extensions. It is not an oversimplification to say that complexity results for the BL family rest on the results for particular MV-algebras, mainly the standard one, and the latter in turn can be derived from well-known results in linear algebra. However, the complexity picture is much less complete for fragments and expansions of BL and of its extensions: here, results are fragmentary despite some considerable effort, while on the other hand, many problems have not been addressed. Shifting from BL to MTL, one moves into an area where open problems outnumber existing results. Decidability results are available for MTL and some of its extensions, and computational complexity has been settled for *particular examples* of left-continuous t-norms. However, a suitable general methodology for tackling complexity problems in semilinear logics weaker than BL is still to be found. Lack of results also prevents us from even mentioning some even weaker semilinear systems; we usually assume our logics are axiomatic extensions of MTL or expansions thereof.

This chapter cannot lay claim to a proper introduction of the investigated logics. For a comprehensive presentation, the reader may wish to consult earlier chapters of this book. Indeed, this chapter will be indigestible to a reader who has not, at the very least, come across the logic BL, its extensions Ł, G, Π, and standard BL-algebras for these logics. Likewise, our treatment of basic computational complexity notions is not intended as an introduction to the topic, but rather as a condensed reference guide. Some skill in algorithmization might also come in useful, as algorithms, where needed, are presented informally within this chapter, and the verification of polynomial nature of some transactions is left to the reader.

The text is organized as follows. Section 2 gives definitions, important notions and results, and notational conventions. Section 3 collects general results, applicable to many particular logics, and some technical statements. Section 4 is dedicated to results on Łukasiewicz logic and its extensions; it contains prototypical complexity results and explains in detail some techniques. Section 5 presents results on (the remaining) extensions of BL given by standard BL-algebras. Section 6 is an overview of available results for fragments and expansions of BL or its extensions. Section 7 gives a flavour of results available for (extensions of) MTL. Section 8 offers an overview of results and Section 9 is an account of achievements in the field, giving references and credits.

2 Notions and problems

This section is a brief exposition of elements of logic, algebra and computational complexity theory. This is accompanied by definitions and discussion of the decision problems that form the subject matter of our investigation. Some other decision problems are pointed out whose already established complexity bounds are relevant.

2.1 Logics and algebras

Logics investigated in this chapter are *algebraizable*; the notion of algebraizability was introduced in [4]. This property amounts to the fact that, under a natural translation between propositional formulas and algebraic identities, provability in a particular propositional logic corresponds to the consequence relation in the (unique) class of algebras that forms its *equivalent algebraic semantics*. In particular, one gets strong completeness w.r.t. this class, which extends also to axiomatic extensions and many language expansions. See Chapter IV for a comprehensive exposition.

Languages and expressions. A *language* $\mathcal{L}$ is a countable set of connectives, each with a given arity in N. The connectives with arity 0 are called *constants*. This chapter only considers languages with finitely many connectives of a positive arity (while there can be infinitely many constants). Given a countably infinite set of variables Var, using the connectives of $\mathcal{L}$ and parentheses one can build in the usual way the set $Fm_{\mathcal{L}}$ of $\mathcal{L}$*-expressions*. These can be viewed as propositional formulas (usually denoted with lowercase Greek characters φ, ψ, etc.) or as algebraic terms (usually denoted with lowercase Latin characters t, s, etc.); first-order $\mathcal{L}$-formulas feature $\mathcal{L}$ as the set of function symbols and $=$ as the predicate symbol.

For $\mathcal{L}$-expressions φ, ψ, we write $\psi \preceq \varphi$ to denote the fact that ψ is a subexpression[1] of φ. For $\mathcal{L}$-expressions φ, ψ, χ, we write $\varphi(\psi/\chi)$ for a substitutional instance of φ where all occurrences of ψ have been replaced with χ. If $X \subseteq Var$, we denote $Fm^X_{\mathcal{L}}$ the $\mathcal{L}$-expressions in variables from X.

If $\mathcal{L}_1 \subseteq \mathcal{L}_2$ are languages and T is a set of $\mathcal{L}_2$-expressions, then the $\mathcal{L}_1$*-fragment* of T is the set $T' \subseteq T$ containing all $\mathcal{L}_1$-expressions in T.

Some languages will be particularly important in this chapter. The logic $\mathrm{FL_{ew}}$ (full Lambek calculus with exchange and weakening) has binary connectives $\&$ (conjunction), $\to$ (implication), $\wedge$, $\vee$ (lattice conjunction/disjunction), and the constant $\overline{0}$. One further defines $\overline{1}$ as $\overline{0} \to \overline{0}$, $\neg\varphi$ as $\varphi \to \overline{0}$, and $\varphi \leftrightarrow \psi$ as $(\varphi \to \psi) \,\&\, (\psi \to \varphi)$. In logics stronger than $\mathrm{FL_{ew}}$, some of the connectives are definable;[2] in particular, $\vee$ is definable in MTL ($\varphi \vee \psi$ is defined as $((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi)$), and both $\vee$ and $\wedge$ are definable in BL ($\varphi \wedge \psi$ is defined as $\varphi \,\&\, (\varphi \to \psi)$). In Łukasiewicz logic, one can define all the above connectives using $\to$ and $\overline{0}$ (but one can also equivalently start with different sets of connectives). In superintuitionistic logics, $\wedge$ and $\&$ coincide. In classical logic, connectives become interdefinable in the familiar manner.

---

[1] A connected substring belonging to $Fm_{\mathcal{L}}$.

[2] If L is a logic (extending or expanding $\mathrm{FL_{ew}}$) in a language $\mathcal{L}$, we say that an n-ary connective $c \in \mathcal{L}$ is definable in L iff there is an $\mathcal{L}\setminus\{c\}$-formula $\varphi(x_1, \ldots, x_n)$ s.t. $\vdash_{\mathrm{L}} c(x_1, \ldots, x_n) \leftrightarrow \varphi(x_1, \ldots, x_n)$. Analogously, we say c is (term-)definable in a class $\mathbb{K}$ of $\mathcal{L}$-algebras iff $\mathbb{K} \models c(x_1, \ldots, x_n) = \varphi(x_1, \ldots, x_n)$ for φ as above.

The logic BL (focal to this chapter), together with many of its extensions, is usually considered in the language $\{\&, \rightarrow, \overline{0}\}$. This is also the case here; this language will be referred to as the *language of* BL. The definable connectives mentioned above are regarded as abbreviations, and one uses the defining formulas to translate any formula containing the definable connectives to the language of BL. Analogously for MTL with respect to the language $\{\&, \rightarrow, \wedge, \overline{0}\}$ (the *language of* MTL). Definable connectives (in particular, $\wedge$ and $\vee$) are used quite freely in many places, and some general results are given for the $\mathrm{FL_{ew}}$-language (in particular, Theorem 3.4.1). Yet a straightforward application of the translations given above, to eliminate definable connectives and thus pass from one language to another, may lead to an exponential blowup in formula size. We argue in Theorem 3.3.3 that there is a translation that preserves satisfiability and tautologousness and that can be performed polynomially.

Propositional logic. A logic L in a language $\mathcal{L}$ is a structural consequence relation $\vdash_{\mathrm{L}} \subseteq \mathcal{P}(Fm_{\mathcal{L}}) \times Fm_{\mathcal{L}}$. Often $\vdash_{\mathrm{L}}$ is given by a deductive system, i.e., axioms and deduction rules; cf. Chapter II for a detailed exposition. In a logical setting, we often speak of (propositional) $\mathcal{L}$-formulas rather than $\mathcal{L}$-expressions. An $\mathcal{L}$-theory is a set of $\mathcal{L}$-formulas. If L is a logic in $\mathcal{L}$, $T \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$, $T \vdash_{\mathrm{L}} \varphi$ reads 'φ is provable from T in L', and $\vdash_{\mathrm{L}} \varphi$ is a case of the former with $T = \emptyset$, meaning 'φ is a theorem of L'.

DEFINITION 2.1.1. *Let $\mathcal{L}$ be a language and* L *a logic in the language $\mathcal{L}$. We denote*

$$\mathrm{THM(L)} = \{\varphi \in Fm_{\mathcal{L}} \mid \ \vdash_{\mathrm{L}} \varphi\} \qquad \text{(theorems of L)}$$

$$\mathrm{CONS(L)} = \{\langle T, \varphi\rangle \in \mathcal{P}_{\mathrm{fin}}(Fm_{\mathcal{L}}) \times Fm_{\mathcal{L}} \mid T \vdash_{\mathrm{L}} \varphi\} \qquad \text{(provability from finite theories in L)}$$

The two above notions—theoremhood (for a formula) and provability (for a formula from a finite theory)—will be in the focus of our attention throughout this chapter. For various logics, it will be our objective to classify the set of theorems and the relation of provability from finite theories as to their computational complexity. The restriction to finite theories is necessitated by the need to work with finite objects.

If L_1 is a logic in a language $\mathcal{L}_1$ and L_2 is a logic in a language $\mathcal{L}_2$, we say that L_2 is an *expansion* of L_1 iff $\mathrm{L}_1 \subseteq \mathrm{L}_2$ (entailing $\mathcal{L}_1 \subseteq \mathcal{L}_2$); if $\mathcal{L}_1 = \mathcal{L}_2$ we say 'extension' rather than 'expansion'. If a logic L_2 in a language $\mathcal{L}_2$ expands a logic L_1 in a language $\mathcal{L}_1$, we say the expansion is *conservative* iff, for each $\mathcal{L}_1$-theory $T \cup \{\varphi\}$, $T \vdash_{\mathrm{L}_2} \varphi$ implies $T \vdash_{\mathrm{L}_1} \varphi$; in such a case, we say that L_1 is the $\mathcal{L}_1$*-fragment* of L_2.

DEFINITION 2.1.2. *Basic logic* BL *in the language* $\{\&, \rightarrow, \overline{0}\}$ *has axioms:*

(A1) $(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$

(A2) $\varphi \,\&\, \psi \rightarrow \varphi$

(A3) $\varphi \,\&\, \psi \rightarrow \psi \,\&\, \varphi$

(A4) $\varphi \,\&\, (\varphi \rightarrow \psi) \rightarrow \psi \,\&\, (\psi \rightarrow \varphi)$

(A5a) $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \,\&\, \psi \rightarrow \chi)$

(A5b) $(\varphi \,\&\, \psi \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$

(A6) $((\varphi \rightarrow \psi) \rightarrow \chi) \rightarrow (((\psi \rightarrow \varphi) \rightarrow \chi) \rightarrow \chi)$

(A7) $\overline{0} \rightarrow \varphi$

The deduction rule of BL *is modus ponens. Moreover, monoidal t-norm logic* MTL *in the language* $\{\&, \to, \wedge, \overline{0}\}$ *has axioms* (A1)–(A3),

(A4a) $\varphi \wedge \psi \to \varphi$

(A4b) $\varphi \wedge \psi \to \psi \wedge \varphi$

(A4c) $\varphi \mathbin{\&} (\varphi \to \psi) \to \varphi \wedge \psi$

(A5)–(A7)*, and deduction rule modus ponens.*

Uppercase Latin characters are used for logics: MTL, BL, SBL, Ł, G, Π stand for monoidal t-norm logic, basic logic, strict basic logic, Łukasiewicz logic, Gödel logic, product logic respectively. These and other logics are discussed in previous chapters.

Algebraic semantics. Let $\mathcal{L}$ be a language. In an algebraic setting, the elements of $\mathcal{L}$ are thought of as function symbols; $=$ is the predicate symbol. Variables in *Var* are usually denoted with $x, y, z, \ldots$. $\mathcal{L}$*-terms* are $\mathcal{L}$-expressions, denoted with $s, t, \ldots$. For a given language $\mathcal{L}$, an *identity* is a formula $t = s$, where t, s are terms. A *quasiidentity* is a formula $\bigwedge_{i \leq n}(t_i = s_i) \to t = s$ for $n \in \mathsf{N}$, where t, t_i, s, s_i, $i \leq n$ are terms. An *open formula* is a formula without quantifiers. A closed formula (or *sentence*) is a formula without free variables. We use uppercase Greek characters (Φ, Ψ, $\ldots$) for first-order formulas.

An $\mathcal{L}$-algebra is a structure $\boldsymbol{A} = \langle A, \langle c^{\boldsymbol{A}} \mid c \in \mathcal{L}\rangle\rangle$; the functions in $\boldsymbol{A}$ are indexed with the function symbols of $\mathcal{L}$ of matching arities. If $\boldsymbol{A}$ is an $\mathcal{L}$-algebra and t is an $\mathcal{L}$-term, $t^{\boldsymbol{A}}$ denotes the function given by t in $\boldsymbol{A}$. If $\boldsymbol{A}$ is an algebra, A stands for its domain. If $\mathcal{L}' \subseteq \mathcal{L}$ are languages and $\boldsymbol{A} = \langle A, \langle c^{\boldsymbol{A}} \mid c \in \mathcal{L}\rangle\rangle$ is an $\mathcal{L}$-algebra, the $\mathcal{L}'$*-reduct of* $\boldsymbol{A}$ is the algebra $\langle A, \langle c^{\boldsymbol{A}} \mid c \in \mathcal{L}'\rangle\rangle$. If $\mathbb{K}$ is a class of $\mathcal{L}$-algebras, the theory of $\mathbb{K}$ is the set of first-order $\mathcal{L}$-formulas valid in each member of $\mathbb{K}$. We are particularly interested in the equational and quasiequational fragments of first-order algebraic theories, as these (for suitably chosen algebras) correspond to theoremhood and provability in our propositional logics via completeness theorems.

The following notation is used for function symbols of the language of $\mathrm{FL_{ew}}$-algebras: $\{*, \to, \wedge, \vee, 0\}$.

While $=$ is the only predicate symbol, our algebras are lattice-ordered, hence we take the liberty of using predicate symbols $\leq$ and $<$, where for any terms t_1 and t_2, $t_1 \leq t_2$ stands for $t_1 \wedge t_2 = t_1$, while $t_1 < t_2$ stands for $(t_1 \leq t_2) \wedge \neg(t_1 = t_2)$; so, by slight abuse, atomic formulas are of the form $t_1 = t_2$, $t_1 \leq t_2$, $t_1 < t_2$ for some terms t_1, t_2 (naturally, under this convention we may no longer claim that atomic formulas are just identities). Moreover, 0 is the least and 1 the greatest element of the lattice order.

If $\mathcal{L}$ is a language and $\boldsymbol{A}$ is an $\mathcal{L}$-algebra, an $\boldsymbol{A}$*-evaluation* on $\mathit{Fm}_{\mathcal{L}}$ is any homomorphism from $\boldsymbol{Fm}_{\mathcal{L}}$ (i.e., the free algebra on *Var*) to $\boldsymbol{A}$. Each mapping $e\colon \mathit{Var} \to A$ can then be uniquely extended to an $\boldsymbol{A}$-evaluation on $\mathit{Fm}_{\mathcal{L}}$. We denote $\mathit{Val}(\boldsymbol{A})$ the set of all $\boldsymbol{A}$-evaluations on $\mathit{Fm}_{\mathcal{L}}$. Further, if $X \subseteq \mathit{Var}$, we denote $\mathit{Val}^X(\boldsymbol{A}) = \{e \restriction \mathit{Fm}^X_{\mathcal{L}} \mid e \in \mathit{Val}(\boldsymbol{A})\}$ (the set of all evaluations on $\mathit{Fm}^X_{\mathcal{L}}$).

For $T \cup \{\varphi\}$ a set of $\mathcal{L}$-expressions, $T \models_{\boldsymbol{A}} \varphi$ iff, for all $\boldsymbol{A}$-evaluations e, we have $e(\varphi) = 1^{\boldsymbol{A}}$ whenever for all $\psi \in T$ we have $e(\psi) = 1^{\boldsymbol{A}}$. For $\mathbb{K}$ a class of $\mathcal{L}$-algebras, $T \models_{\mathbb{K}} \varphi$ iff $T \models_{\boldsymbol{A}} \varphi$ for all $\boldsymbol{A} \in \mathbb{K}$. The relation $\models_{\mathbb{K}}$ is referred to as the *consequence relation in* $\mathbb{K}$. The *finite consequence relation* in $\mathbb{K}$ is the restriction of $\models_{\mathbb{K}}$ to finite theories. We often use the notion of *logic given by* $\mathbb{K}$ (or simply 'logic of $\mathbb{K}$'): within this chapter, the logic given by $\mathbb{K}$ is identified with the finite consequence relation of $\mathbb{K}$.[3] We write $\models_{\mathbb{K}} \varphi$ for $\emptyset \models_{\mathbb{K}} \varphi$ and we speak of *tautologies* of $\mathbb{K}$.

We now define some of these, and other, familiar notions as operators on (classes of) algebras, generalizing the cases from classical logic.

DEFINITION 2.1.3. *Let $\mathcal{L}$ be a language subsuming the language of $\mathrm{FL_{ew}}$. Let $\mathbb{K} \cup \{\boldsymbol{A}\}$ be a class of $\mathcal{L}$-algebras whose reducts to the $\mathrm{FL_{ew}}$-language are $\mathrm{FL_{ew}}$-algebras, and let $\mathbf{1}$ denote the trivial $\mathcal{L}$-algebra. We denote*

$$\mathrm{TAUT}(\boldsymbol{A}) = \{\varphi \in \mathit{Fm}_{\mathcal{L}} \mid \forall e \in \mathit{Val}(\boldsymbol{A})(e(\varphi) = 1^{\boldsymbol{A}})\} \qquad \text{(tautologies of } \boldsymbol{A})$$

$$\mathrm{TAUT_{pos}}(\boldsymbol{A}) = \{\varphi \in \mathit{Fm}_{\mathcal{L}} \mid \forall e \in \mathit{Val}(\boldsymbol{A})(e(\varphi) > 0^{\boldsymbol{A}})\} \qquad \text{(positive tautologies of } \boldsymbol{A})$$

$$\mathrm{SAT}(\boldsymbol{A}) = \{\varphi \in \mathit{Fm}_{\mathcal{L}} \mid \exists e \in \mathit{Val}(\boldsymbol{A})(e(\varphi) = 1^{\boldsymbol{A}})\} \qquad \text{(satisfiable formulas of } \boldsymbol{A})$$

$$\mathrm{SAT_{pos}}(\boldsymbol{A}) = \{\varphi \in \mathit{Fm}_{\mathcal{L}} \mid \exists e \in \mathit{Val}(\boldsymbol{A})(e(\varphi) > 0^{\boldsymbol{A}})\} \qquad \text{(positively satisfiable formulas of } \boldsymbol{A})$$

$$\mathrm{CONS}(\boldsymbol{A}) = \{\langle T, \varphi\rangle \in \mathcal{P}_{\mathrm{fin}}(\mathit{Fm}_{\mathcal{L}}) \times \mathit{Fm}_{\mathcal{L}} \mid T \models_{\boldsymbol{A}} \varphi\} \qquad \text{(finite consequence in } \boldsymbol{A})$$

$$\mathrm{TAUT}(\mathbb{K}) = \bigcap_{\boldsymbol{A} \in \mathbb{K}} \mathrm{TAUT}(\boldsymbol{A}) \qquad \text{(tautologies of } \mathbb{K})$$

$$\mathrm{TAUT_{pos}}(\mathbb{K}) = \bigcap_{\boldsymbol{A} \in (\mathbb{K} \setminus \{\mathbf{1}\})} \mathrm{TAUT_{pos}}(\boldsymbol{A}) \qquad \text{(positive tautologies of } \mathbb{K})$$

$$\mathrm{SAT}(\mathbb{K}) = \bigcup_{\boldsymbol{A} \in (\mathbb{K} \setminus \{\mathbf{1}\})} \mathrm{SAT}(\boldsymbol{A}) \qquad \text{(satisfiable formulas of } \mathbb{K})$$

$$\mathrm{SAT_{pos}}(\mathbb{K}) = \bigcup_{\boldsymbol{A} \in \mathbb{K}} \mathrm{SAT_{pos}}(\boldsymbol{A}) \qquad \text{(positively satisfiable formulas of } \mathbb{K})$$

$$\mathrm{CONS}(\mathbb{K}) = \bigcap_{\boldsymbol{A} \in \mathbb{K}} \mathrm{CONS}(\boldsymbol{A}) \qquad \text{(finite consequence in } \mathbb{K})$$

In the above definition, the trivial algebra $\mathbf{1}$ is omitted from consideration for the $\mathrm{TAUT_{pos}}$ and SAT operators, because $\mathrm{TAUT_{pos}}(\mathbf{1}) = \emptyset$ and $\mathrm{SAT}(\mathbf{1}) = \mathit{Fm}_{\mathcal{L}}$. This seems more convenient than handling the omission separately for each case.

NOTATION 2.1.4. For $\mathbb{K}$ a class of algebras, the term $\mathrm{SAT_{(pos)}}(\mathbb{K})$ stands for either of the problems $\mathrm{SAT}(\mathbb{K})$ and $\mathrm{SAT_{pos}}(\mathbb{K})$. Similarly for $\mathrm{TAUT_{(pos)}}(\mathbb{K})$.

We now come to the notion of the (first-order) *theory of* $\mathbb{K}$, where $\mathbb{K}$ is a class of algebras in a given language $\mathcal{L}$. The theory of $\mathbb{K}$ is the set of first-order $\mathcal{L}$-sentences Φ valid in $\mathbb{K}$ (i.e., valid in each $\boldsymbol{A} \in \mathbb{K}$; write $\mathbb{K} \models \Phi$). We use 'equational theory of $\mathbb{K}$' for 'equational fragment of the theory of $\mathbb{K}$'; analogously for the other fragments.

[3]This is not quite consistent with the conception of logics as consequence relations, because the latter involve infinite sets of formulas. However, the finite consequence relation captures full information about the consequence relation in case the latter is finitary, or about its finitary companion in case it is not.

DEFINITION 2.1.5. *Let $\mathcal{L}$ be a language, $\mathbb{K}$ a class of $\mathcal{L}$-algebras. We write*

(i) $\mathrm{Th}_{\mathrm{Eq}}(\mathbb{K})$ *for the equational theory of* $\mathbb{K}$*, i.e., the set of universally quantified* $\mathcal{L}$*-identities valid in* $\mathbb{K}$;

(ii) $\mathrm{Th}_{\mathrm{QEq}}(\mathbb{K})$ *for the quasiequational theory of* $\mathbb{K}$*, i.e., the set of universally quantified* $\mathcal{L}$*-quasiidentities valid in* $\mathbb{K}$;

(iii) $\mathrm{Th}_{\forall}(\mathbb{K})$ *for the universal theory of* $\mathbb{K}$*, i.e., the set of universally quantified open* $\mathcal{L}$*-formulas valid in* $\mathbb{K}$;

(iv) $\mathrm{Th}_{\exists}(\mathbb{K})$ *for the existential theory of* $\mathbb{K}$*, i.e., the set of existentially quantified open* $\mathcal{L}$*-formulas valid in some* $\boldsymbol{A} \in \mathbb{K}$;

(v) $\mathrm{Th}(\mathbb{K})$ *for the (full, first-order) theory of* $\mathbb{K}$.

It is obvious from the definition that $\mathrm{Th}_{\mathrm{Eq}}(\mathbb{K}) \subseteq \mathrm{Th}_{\mathrm{QEq}}(\mathbb{K}) \subseteq \mathrm{Th}_{\forall}(\mathbb{K}) \subseteq \mathrm{Th}(\mathbb{K})$ and $\mathrm{Th}_{\exists}(\mathbb{K}) \subseteq \mathrm{Th}(\mathbb{K})$. It is important to observe that all the inclusions in fact stand for fragments given by conditions that are easy to verify.

The link between the concepts introduced in the last two definitions is clear: for any language $\mathcal{L}$, any $\mathcal{L}$-algebra $\boldsymbol{A}$, any $\boldsymbol{A}$-evaluation e and any $\mathcal{L}$-expression φ, we have $e(\varphi) = 1^{\boldsymbol{A}}$ iff $\boldsymbol{A} \models (\varphi = 1)[e]$. Hence, e.g., $\varphi \in \mathrm{TAUT}(\boldsymbol{A})$ iff $(\varphi = 1) \in \mathrm{Th}_{\mathrm{Eq}}(\boldsymbol{A})$. This yields some straightforward reducibilities, which are collected in Lemma 3.1.1.

If L is a logic in a language $\mathcal{L}$, we usually write $\mathbb{L}$ for the class of $\mathcal{L}$-algebras that forms its equivalent algebraic semantics. The elements of $\mathbb{L}$ are referred to as L-algebras; the linearly ordered elements are L-*chains*.

If $\mathcal{L}$ is a language and $\mathbb{K}$ a class of $\mathcal{L}$-algebras, we write $\mathbf{V}(\mathbb{K})$ for the variety and $\mathbf{Q}(\mathbb{K})$ for the quasivariety generated by $\mathbb{K}$; further, we write $\mathbf{I}(\mathbb{K})$, $\mathbf{H}(\mathbb{K})$, $\mathbf{S}(\mathbb{K})$, $\mathbf{P}(\mathbb{K})$, $\mathbf{P}_{\mathrm{U}}(\mathbb{K})$ for the classes of isomorphic images of $\mathbb{K}$, homomorphic images of $\mathbb{K}$, subalgebras of $\mathbb{K}$, direct products of $\mathbb{K}$, ultraproducts of $\mathbb{K}$, respectively. For $\boldsymbol{A}, \boldsymbol{B}$ two $\mathcal{L}$-algebras, $\boldsymbol{A}$ is *partially embeddable* into $\boldsymbol{B}$ iff every finite partial subalgebra of $\boldsymbol{A}$ is embeddable into $\boldsymbol{B}$, that is, for each finite set $A_0 \subseteq A$ there is a one-one mapping $f\colon A_0 \to B$ such that for each n-ary function symbol g in $\mathcal{L}$, if for $a_1, \ldots, a_n \in A_0$ we have $g^{\boldsymbol{A}}(a_1, \ldots, a_n) \in A_0$, then $f(g^{\boldsymbol{A}}(a_1, \ldots, a_n)) = g^{\boldsymbol{B}}(f(a_1), \ldots, f(a_n))$. For $\mathbb{K}$, $\mathbb{L}$ two classes of $\mathcal{L}$-algebras, $\mathbb{K}$ is partially embeddable into $\mathbb{L}$ iff each finite partial subalgebra of a member of $\mathbb{K}$ is embeddable into a member of $\mathbb{L}$.

Structure of BL-chains. We review a few important facts about decomposition of BL-chains as ordinal sums. This decomposition is an essential part of standard completeness results for BL and also—as we shall see—of the results on its computational complexity. We remark that, while MTL (unlike BL) actually enjoys strong standard completeness, an analogously lucid result about the structure of MTL-chains is not available. Within this book, BL-algebras are studied in detail in Chapter V.

A t-norm $*$ on $[0, 1]$ is a binary operation that is associative, commutative, nondecreasing, satisfying boundary conditions $x * 0 = 0$ and $x * 1 = x$. If $*$ is left continuous, then its residuum $\to$ is uniquely given by $x \to y = \max\{z \mid x * z \leq y\}$ and $[0, 1]_* = \langle [0, 1], *, \to, \wedge, \vee, 0, 1\rangle$ is a standard MTL-algebra. $[0, 1]_*$ is a standard BL-algebra iff $*$ is continuous.

There are three outstanding examples of continuous t-norms; together with their residua, they are listed in the following table (note that $x \to y = 1$ whenever $x \leq y$):

	$x * y$	$x \to y$ for $x > y$
Łukasiewicz	$\max\{x + y - 1, 0\}$	$1 - x + y$
Gödel	$\min\{x, y\}$	y
product	xy	y/x

The next proposition justifies the importance of the three examples above. For a continuous t-norm $*$, the set of its idempotents is a closed subset of $[0, 1]$, its complement is a union of countably many pairwise disjoint open intervals; denote this set of intervals $\mathcal{I}_o$. Let $\mathcal{I}$ be the set of closures of the elements of $\mathcal{I}_o$.

PROPOSITION 2.1.6 (Mostert–Shields Theorem [36]).
Let $*$ be a continuous t-norm on $[0, 1]$.

(i) For each $[a, b] \in \mathcal{I}$, the restriction of $*$ to $[a, b]$ is isomorphic either to the product t-norm on $[0, 1]$ or to the Łukasiewicz t-norm on $[0, 1]$.

(ii) If there are no a, b such that $x, y \in [a, b] \in \mathcal{I}$, then $x * y = \min\{x, y\}$.

For each standard BL-algebra $[0, 1]_*$, the maximal, nontrivial, closed intervals on which $*$ is isomorphic to the Łukasiewicz, Gödel, or product t-norm are referred to as Ł-*components*, G-*components*, and Π-*components* of the t-norm, hence of the algebra $[0, 1]_*$. Not every element of $[0, 1]_*$ belongs to an Ł, G, or Π-component; one also considers trivial, one-element algebras as possible components. If $\boldsymbol{A}$ is a (standard) BL-algebra, one can write $\boldsymbol{A} = \bigoplus_{i \in I} \boldsymbol{A}_i$ for some linearly ordered index set I and for each $\boldsymbol{A}_i$ among copies of $[0, 1]_{\text{Ł}}$, $[0, 1]_{\text{G}}$, $[0, 1]_{\Pi}$, and the trivial algebra $\mathbf{1}$.

We remark that one can prove an analogous decomposition result for saturated BL-chains, and also that BL-chains can be decomposed as ordinal sums of Wajsberg hoops. We will not need these results in this chapter.

Completeness. If $\mathcal{L}$ is a language, L is a logic in the language $\mathcal{L}$ and $\mathbb{K}$ is a class of $\mathcal{L}$-algebras, we say that L is:

(i) *complete* w.r.t. $\mathbb{K}$ iff, for each $\mathcal{L}$-formula, we have $\vdash_{\text{L}} \varphi$ iff $\models_{\mathbb{K}} \varphi$ (i.e., THM(L) = TAUT($\mathbb{K}$));

(ii) *finitely strongly complete* w.r.t. $\mathbb{K}$ iff, for each finite set $T \cup \{\varphi\}$ of $\mathcal{L}$-formulas, we have $T \vdash_{\text{L}} \varphi$ iff $T \models_{\mathbb{K}} \varphi$ (i.e., CONS(L) = CONS($\mathbb{K}$));

(iii) *strongly complete* w.r.t. $\mathbb{K}$ iff, for each set $T \cup \{\varphi\}$ of $\mathcal{L}$-formulas, we have $T \vdash_{\text{L}} \varphi$ iff $T \models_{\mathbb{K}} \varphi$.

Obviously (iii) implies (ii) and (ii) implies (i) for any choice of $\mathcal{L}$, L and $\mathbb{K}$. Algebraizability of a logic L implies strong completeness w.r.t. the class of algebras $\mathbb{L}$ forming its equivalent algebraic semantics; logics investigated within this chapter are semilinear, and hence, strongly complete w.r.t. the chains in $\mathbb{L}$.

In this chapter we are mainly interested in *standard*[4] algebras for each logic L: completeness results, where available, are then formulated in terms of *standard completeness* (SC), *finite strong standard completeness* (FSSC), or *strong standard completeness*

[4]For many logics/classes of algebras (such as BL or MTL), the notion 'standard algebra' has a clear and established meaning. In other cases (and particularly for some expanded languages), it is better to state explicitly what is meant by 'standard'.

(SSC), where in all cases, the term 'standard' means that in the above definitions, the role of the class $\mathbb{K}$ is played by standard algebras in $\mathbb{L}$.

PROPOSITION 2.1.7 (Standard completeness). *The logic* BL *enjoys finite strong standard completeness. The logic* MTL *enjoys strong standard completeness.*

2.2 A visit to complexity theory

Computational model. *Turing machines* capture essential notions of algorithmization, such as computations and their resources, notably *time* and *space*; algorithms are formally identified with Turing machines. A Turing machine has a finite sequence of *tapes* for data storage, each tape consisting of infinitely many fields, with a cursor indicating the current field. One of the tapes is the input tape, and there may also be an output tape; the input (output) tape is assumed to be read-only (write-only). Tape fields may be blank or may contain symbols out of a given finite alphabet. A particular Turing machine is fully determined by a finite alphabet Σ, a finite set of states Q (with the initial state $q_0 \in Q$), and a finite set of instructions Δ.

Each computation starts with all tapes blank except the input tape, which includes the input—a finite string of symbols from Σ, with the cursor on its leftmost symbol; the machine's state is the initial state $q_0 \in Q$. The computation runs in steps, each step processing one instruction from Δ. The next instruction is chosen on basis of the current state and the content of current fields on the sequence of tapes. Each instruction consists of a current state of the machine, a sequence of symbols on current fields of all tapes (some of which may be blank), the next state of the machine out of Q (which may be one of its halting states), a sequence of symbols to be written down to the current fields of all tapes (some of which may be blank), and a sequence out of $\{-1, 0, 1\}$ indicating the move of cursors on all tapes by at most one field. A computation may terminate or not depending on whether a halting state is reached. If it does, output may be written on an output tape. Among halting states, some states may be indicated as accepting or rejecting. A computation will not continue from a halting state, as there is no instruction available; for all other states and sequences of symbols read from tapes, there are one or more instructions available in Δ; if the former is the case for all possible combinations of states and read sequences—or in other words, if the transition relation induced by Δ is a *function*—then the Turing machine in question is *deterministic*; otherwise, it is *nondeterministic*.

Decision problems. Let Σ be a finite alphabet; Σ^* is the set of finite strings out of Σ; a *word* is a finite string $x \in \Sigma^*$ and $|x|$ denotes the *size* of x (the number of symbols on tape). A *decision problem* (or just 'problem') is a set of words $P \subseteq \Sigma^*$. Words in Σ^* are often called inputs or instances. The complement of a problem P is $\overline{P} = \Sigma^* \setminus P$. A Turing machine M with alphabet Σ *accepts* a problem P iff, for each word $x \in \Sigma^*$, we have $x \in P$ iff there is a computation of M with input x that terminates in an accepting state. A problem $P \subseteq \Sigma^*$ is *recursively enumerable* iff it is accepted by a Turing machine. A problem $P \subseteq \Sigma^*$ is *recursive* (or *decidable*) iff both P and $\overline{P}$ are accepted by a Turing machine. This entails there is a Turing machine with alphabet Σ which terminates on any word x in the given alphabet—in an accepting state if $x \in P$, and in a rejecting state if $x \in \overline{P}$; such a machine is said to *decide* P.

Complexity classes. Consider functions $f, g\colon \mathsf{N} \to \mathsf{N}$. Then $f \in O(g)$ ('f is of the order of g') iff there are $c, n_0 \in \mathsf{N}$ such that for each $n \geq n_0$ we have $f(n) \leq c\, g(n)$.

Let $f\colon \mathsf{N} \to \mathsf{N}$ be a function. A Turing machine M (deterministic or not) operates in time f iff, for any input x in the alphabet Σ of M, any computation with input x takes at most $f(|x|)$ steps. $\mathbf{TIME}(f)$ is the class of problems P such that there is a deterministic Turing machine M that accepts P and operates in time $O(f)$; analogously for $\mathbf{NTIME}(f)$ and nondeterministic Turing machines. A Turing machine M (deterministic or not) operates in space f iff, for any input x in the alphabet Σ of M, any computation with input x (terminates and) writes to at most $f(|x|)$ fields on all its tapes together except the input and the output tapes. $\mathbf{SPACE}(f)$ is the class of problems P such that there is a deterministic Turing machine M that accepts P and operates in space $O(f)$; analogously for $\mathbf{NSPACE}(f)$ and nondeterministic Turing machines.
Particular complexity classes important in this chapter are defined as follows:

$$\mathbf{P} = \bigcup_{k \in \mathsf{N}} \mathbf{TIME}(n^k)$$

$$\mathbf{NP} = \bigcup_{k \in \mathsf{N}} \mathbf{NTIME}(n^k)$$

$$\mathbf{PSPACE} = \bigcup_{k \in \mathsf{N}} \mathbf{SPACE}(n^k)$$

If $\mathbf{C}$ is a complexity class, we denote $\mathbf{coC} = \{P \mid \overline{P} \in \mathbf{C}\}$, the class of complements of problems in $\mathbf{C}$. Each deterministic complexity class $\mathbf{C}$ is closed under complementation: if $P \in \mathbf{C}$, then also $\overline{P} \in \mathbf{C}$. It is widely believed, but not known, not to be the case for the class $\mathbf{NP}$. By definition, $\mathbf{P} \subseteq \mathbf{NP}$ and hence $\mathbf{P} \subseteq \mathbf{coNP}$, and it is easy to see that $\mathbf{NP} \subseteq \mathbf{PSPACE}$. It is an important open problem whether any of the inclusions $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$ are proper. Each of the classes $\mathbf{P}$, $\mathbf{NP}$, $\mathbf{coNP}$, and $\mathbf{PSPACE}$ is closed under finite unions and intersections.

The following is an equivalent definition of the class $\mathbf{NP}$: a problem $P \subseteq \Sigma^*$ is in $\mathbf{NP}$ iff there is a polynomially balanced[5] binary relation $R \subseteq \Sigma^* \times \Sigma^*$ in $\mathbf{P}$, such that $P = \{x \in \Sigma^* \mid \exists y \in \Sigma^*(\langle x, y\rangle \in R)\}$. Any such word y is called a *witness* for $x \in P$. It is easy to see that any problem P that satisfies this definition is in $\mathbf{NP}$: given x, first "guess" y and then continue (deterministically) to check $\langle x, y\rangle \in R$. This definition of the class $\mathbf{NP}$ constitutes the basis for many proofs of containment in $\mathbf{NP}$ within this chapter. It also hints at how to transform each nondeterministic polynomial-time algorithm into a deterministic one; given x, one searches through all possible witnesses y (up to a polynomial bounded size) and for each such y, checks whether $\langle x, y\rangle \in R$.

Many nondeterministic algorithms in this chapter follow the guess-and-check pattern described above. In the guessing stage, an information of size polynomial in the input size is guessed. The checking stage may come in several steps. In each step, it is understood (though not stated explicitly each time the construction is used), that if a check is unsuccessful, then the given computation terminates in a rejecting state. Likewise, the formulation 'guess an X such that $C(X)$' (for some condition C) is to be understood as 'guess X and check that $C(X)$ holds'.

[5] A relation R is polynomially balanced iff there is a polynomial p s.t. $\langle x, y\rangle \in R$ implies $|y| \leq p(|x|)$.

Reductions and completeness. Throughout this chapter we use the polynomial-time many-one reducibility, also known as Karp reducibility. To define a reduction between two problems P_1 (in an alphabet Σ_1) and P_2 (in an alphabet Σ_2), it is convenient to consider a Turing machine with input tape alphabet Σ_1 and output tape alphabet Σ_2. A problem P_1 is (many-one, polynomial-time) *reducible* to a problem P_2 (write $P_1 \preceq_{\mathbf{P}} P_2$)[6] iff there is a deterministic Turing machine with input tape alphabet Σ_1 and output tape alphabet Σ_2, operating in time n^k for some $k \in \mathsf{N}$ and all $n \geq n_0 \in \mathsf{N}$, and such that, for any pair of input $x \in \Sigma_1^*$ and its output $y \in \Sigma_2^*$ on M, we have $x \in P_1$ iff $y \in P_2$. In other words, there is a polynomial-time-computable function $f\colon \Sigma_1^* \to \Sigma_2^*$ such that $P_1 = \{x \in \Sigma_1^* \mid f(x) \in P_2\}$; if that is the case, then also $\overline{P_1} = \{x \in \Sigma_1^* \mid f(x) \in \overline{P_2}\}$. Reducibility is a preorder, inducing its corresponding equivalence: two decision problems P_1 and P_2 are *polynomially equivalent* (write $P_1 \approx_{\mathbf{P}} P_2$) iff $P_1 \preceq_{\mathbf{P}} P_2$ and $P_2 \preceq_{\mathbf{P}} P_1$. The equivalence $\approx_{\mathbf{P}}$ provides a classification of decision problems of roughly the same complexity. Even though we are currently unable to tell how equivalence classes of $\approx_{\mathbf{P}}$ span over complexity classes defined above, we can prove positive results on $\approx_{\mathbf{P}}$ for particular decision problems.

A decision problem P is said to be *hard* for a complexity class $\mathbf{C}$ (shortly, $\mathbf{C}$-hard) iff any decision problem P' in $\mathbf{C}$ is reducible to P. A decision problem P is *complete* in $\mathbf{C}$ (shortly, $\mathbf{C}$-complete) iff P is $\mathbf{C}$-hard and $P \in \mathbf{C}$. Thanks to transitivity of $\preceq_{\mathbf{P}}$, hardness of a problem P for $\mathbf{C}$ is typically demonstrated by reducing to P *one* problem already known to be $\mathbf{C}$-hard. Showing that a problem is $\mathbf{C}$-hard can be viewed as setting a *lower bound* on its complexity: it is no easier to solve than the problems in $\mathbf{C}$.

Complexity classes in the focus of our attention—$\mathbf{P}$, $\mathbf{NP}$, and $\mathbf{PSPACE}$—are closed under $\preceq_{\mathbf{P}}$: if $\mathbf{C}$ is one of the above classes, $P_1 \preceq_{\mathbf{P}} P_2$ and $P_2 \in \mathbf{C}$, then also $P_1 \in \mathbf{C}$. This provides a way to demonstrate *containment* of a problem in a class $\mathbf{C}$. Containment in $\mathbf{C}$ can of course be proved in a direct way, by designing an algorithm that works within resource bounds given by $\mathbf{C}$. The algorithm may use subroutines that also satisfy the bounds for $\mathbf{C}$. Showing containment of a problem in a complexity class sets an *upper bound* on its complexity: given the computation mode and the bounds, the problem is algorithmically solvable.

Following the nature of decision problems investigated in this chapter, we shall be mainly interested in the classes $\mathbf{NP}$, $\mathbf{coNP}$, and $\mathbf{PSPACE}$, namely in the respective subclasses of problems that are complete for each of them. With each problem, we seek to find a match between its upper and its lower bound; then the problem is ranked alongside other problems already known to be in the particular $\approx_{\mathbf{P}}$-class.

As a matter of fact, a classification in the above sense for many decision problems in fuzzy logic is missing. There are lower bounds that predetermine the problems investigated in this chapter to be computationally hard (cf. Theorem 3.4.1). In particular, theorems of a consistent fuzzy logic extending $\mathrm{FL}_{\mathrm{ew}}$ are always $\mathbf{coNP}$-hard. However, some problems may be much harder than that. As for upper bounds, some decision problems are known to be recursive, but no more than that. Important examples include theoremhood and provability from finite theories in MTL and some of its axiomatic

[6] It would be more docile to write $\preceq_{\mathrm{m}}^{\mathbf{P}}$, since Karp reducibility is exactly the polynomial-time analogue of many-one reducibility $\preceq_{\mathrm{m}}$ provided by recursive functions, to be introduced later.

extensions, such as IMTL, SMTL, or $\Pi\mathrm{MTL}$. Another example is provided by theoremhood in the logics $\mathrm{Ł}\Pi$ and $\mathrm{Ł}\Pi\frac{1}{2}$: the problem is known to be in $\mathbf{PSPACE}$, but apparently not known to be complete for that class.

Arithmetical hierarchy. Let N be the standard model of arithmetic. Let $\Phi(x)$ be an arithmetical formula with one free variable; we say $\Phi(x)$ defines a set $A \subseteq \mathsf{N}$ iff for any $n \in \mathsf{N}$, we have $n \in A$ iff $\mathsf{N} \models \Phi(n)$; we say A is definable in N iff there is a Φ that defines it in N. Analogously, one can introduce definable relations in N^k for each natural number k. Via coding, one can consider words over finite alphabets.

An arithmetical formula is *bounded* iff all its quantifiers are bounded (i.e., are of the form $\forall x \leq t$ or $\exists x \leq t$ for some term t). An arithmetical formula is a Σ_1-formula (Π_1-formula) iff it has the form $\exists x\Phi$ ($\forall x\Phi$ respectively) where Φ is a bounded formula. A formula is Σ_2 (Π_2) iff it has the form $\exists x\Phi$ ($\forall x\Phi$ respectively) where Φ is a Π_1-formula (Σ_1-formula respectively). Inductively, one defines Σ_n- and Π_n-formulas for any natural number $n \geq 1$.

A set $A \subseteq \mathsf{N}$ is in the class $\mathbf{\Sigma_n}$ iff there is a Σ_n-formula that defines A in N; analogously for the class $\mathbf{\Pi_n}$. The definition extends to k-tuples and to words over finite alphabets in the obvious fashion. Trivially, any set that is in $\mathbf{\Sigma_n}$ is also in $\mathbf{\Sigma_m}$ and $\mathbf{\Pi_m}$ for $m > n$. If $A \subseteq \mathsf{N}$ is a $\mathbf{\Sigma_n}$-set, then $\overline{A}$ is a $\mathbf{\Pi_n}$-set. $\mathbf{\Sigma_1}$-sets are exactly recursively enumerable sets, while recursive sets are $\mathbf{\Sigma_1} \cap \mathbf{\Pi_1}$. The hierarchy of classes of sets thus defined is called the *arithmetical hierarchy*, and the (complete sets in) classes of sets in the arithmetical hierarchy represent *degrees of undecidability*. The hierarchy is noncollapsing, as it can be shown that for each $n \geq 1$, $\mathbf{\Sigma_{n+1}} \setminus \mathbf{\Sigma_n}$ is nonempty, and so is $\mathbf{\Sigma_1} \setminus (\mathbf{\Sigma_1} \cap \mathbf{\Pi_1})$. A set $A \subseteq \mathsf{N}$ is *arithmetical* iff it is definable by an arithmetical formula in N and so it belongs to the arithmetical hierarchy; otherwise it is *nonarithmetical*.

A suitable notion of reduction is provided by recursive functions. A problem P_1 in an alphabet Σ_1 is *m-reducible* to a problem P_2 in an alphabet Σ_2 (write $P_1 \preceq_{\mathrm{m}} P_2$) iff there is a deterministic Turing machine with input tape alphabet Σ_1 and output tape alphabet Σ_2, halting on all inputs, and such that, for any pair of input x and its output y, we have $x \in P_1$ iff $y \in P_2$. Each of the classes $\mathbf{\Sigma_n}$, $\mathbf{\Pi_n}$ ($n \geq 1$) is closed under m-reducibility. A problem P is $\mathbf{\Sigma_n}$-*hard* (w.r.t. m-reducibility) iff $P' \preceq_{\mathrm{m}} P$ for any $\mathbf{\Sigma_n}$-problem P'. A problem P is $\mathbf{\Sigma_n}$-complete iff it is $\mathbf{\Sigma_n}$-hard and at the same time it is a $\mathbf{\Sigma_n}$-problem. Analogously for $\mathbf{\Pi_n}$.

2.3 Formulas as inputs

If $\mathcal{L}$ is a language, $\mathcal{L}$-expressions $Fm_{\mathcal{L}}$ are well-formed[7] strings, consisting of variables in Var, connectives in $\mathcal{L}$, and and auxiliary symbols (parentheses). First-order algebraic formulas moreover feature the identity symbol $=$, Boolean connectives, quantifiers $\forall$, $\exists$. Again these formulas are well-formed strings.

An essential question is how resources used by an algorithm depend on the size of an input. Inputs are formulas (propositional or first-order), viewed as words in a finite alphabet. We assume a fixed enumeration of the sets of variables and of connectives. Integers are represented in binary, $|n| = \lceil \log(n+1) \rceil$ for $n \geq 1$, so $|n| \in O(\log(n))$.[8]

[7] I.e., they satisfy the usual inductive definition of a propositional formula in the given language.

[8] Throughout we use base 2 logarithm. We write $O(\log(n))$ for $O(\max\{\lceil \log(n) \rceil, 0\}$.

The size of an $\mathcal{L}$-expression φ is the number of tape fields needed to represent it, denoted $|\varphi|$. Given φ, the value $|\varphi|$ is obtained by adding up the sizes of representations of all occurrences of connectives, all occurrences of variables, and all occurrences of auxiliary symbols. Moreover, if φ is an $\mathcal{L}$-expression with n pairwise distinct variables, it is convenient (and equivalent for our purpose) to consider its substitution instance whose variables are indexed with integers up to n; this brings the space needed to represent each of the variables down to $O(\log(n))$.

It is preferable to work with more versatile measures than the actual formula size: in particular, for $\mathcal{L}$-expressions, the number of occurrences of subexpressions, or the overall size of constants in the expression; if we show an algorithm to be polynomial in a measure bounded by $|\varphi|$, then we may conclude it is also polynomial in $|\varphi|$.

On the other hand, $|\varphi|$ is polynomial in the measures mentioned above. Indeed, for an expression φ, denote m the number of occurrences of subexpressions of φ. Each variable takes $O(\log(m))$ tape fields. In a language with finitely many connectives, each connective takes a constant number of fields, so $|\varphi| \in O(m \log(m))$. As for languages with infinitely many connectives, we attend the case of constants for $\mathsf{Q} \cap [0,1]$: for each such q, if $q = \frac{a}{b}$ for some $a, b \in \mathsf{N}$ where $a \leq b$, we have $|q| \in O(|b|)$. Hence, for an expression φ with constants from $\mathsf{Q} \cap [0,1]$, with m occurrences of subformulas, and whose constants have the largest denominator k, we get $|\varphi| \in O(m(\log(m) + \log(k)))$.

For first-order formulas the above considerations are analogous. Validity is a meaningful concept for *sentences*; if a formula Φ is not a sentence, then we consider its universal closure, whose size is polynomial in the size of Φ. Moreover, it is convenient to only consider sentences in prenex form; bringing a given sentence into the prenex form is a routine polynomial-time transformation.

Not all words in the given alphabet are desirable inputs. The assumption of well-formedness in a given language $\mathcal{L}$ (propositional or first-order) is always present. In many cases there are more restrictive assumptions, like the words being universally quantified $\mathcal{L}$-quasiidentities, existential $\mathcal{L}$-sentences, etc. These assumptions are made explicitly for each decision problem. A common trait of these assumptions is that the assumed condition is easy to verify: the class of words $C \subseteq \Sigma^*$ that satisfy the condition is a decision problem in $\mathbf{P}$ (and of limited interest to us).

To illustrate the difference these assumptions make, consider an algorithm accepting $\mathrm{SAT}(\boldsymbol{A})$—the set of satisfiable $\mathcal{L}$-expressions in an $\mathcal{L}$-algebra $\boldsymbol{A}$. The algorithm accepts satisfiable expressions and rejects unsatisfiable ones; what about words that are not $\mathcal{L}$-expressions? In view of definitions presented earlier, the algorithm should reject them. Then—inconveniently—the set of rejected words would consist of a) words that are not $\mathcal{L}$-expressions, and b) $\mathcal{L}$-expressions that are not satisfiable in $\boldsymbol{A}$. Now a) is not at all interesting, while one might be interested in b); indeed b) is the desired complement to $\mathrm{SAT}(\boldsymbol{A})$. This preference can be met by allowing only $\mathcal{L}$-expressions as inputs to the algorithm. (One may think of an auxiliary algorithm that test all input strings for being well-formed $\mathcal{L}$-expressions.) Continuing the given example, suppose we have indeed shown that $\mathrm{SAT}(\boldsymbol{A})$ is in $\mathbf{NP}$, and, for some $\mathcal{L}' \subset \mathcal{L}$, we further want to investigate satisfiability for $\mathcal{L}'$-expressions in $\boldsymbol{A}'$, the $\mathcal{L}'$-reduct of $\boldsymbol{A}$. Can it be argued that $\mathrm{SAT}(\boldsymbol{A}') \preceq_{\mathbf{P}} \mathrm{SAT}(\boldsymbol{A})$ via an identity function (or in other words, can the algorithm deciding $\mathrm{SAT}(\boldsymbol{A})$ be used also for deciding $\mathrm{SAT}(\boldsymbol{A}')$)? Not quite, because the algorithm for $\mathrm{SAT}(\boldsymbol{A})$ accepts all satisfiable $\mathcal{L}$-formulas, some of whom are not (satisfiable) $\mathcal{L}'$-formulas. However, the identity function can be used for reduction if one makes sure that all inputs to the new algorithm are among $\mathcal{L}'$-expressions.

The informal considerations above have a formal counterpart called *promise problems*. A promise problem in an alphabet Σ is a pair (Y, N) where $Y, N \subseteq \Sigma^*$ and $Y \cap N = \emptyset$; the set $Y \cup N$ is called the *promise*. A Turing machine decides the problem (Y, N) iff it accepts all words in Y and rejects all words in N; on inputs outside $Y \cup N$, its behaviour is not specified. Intuitively, an algorithm solving a promise problem is promised that inputs belong to $Y \cup N$; on this condition, it distinguishes the two sets (given computation mode and bounds). Any decision problem P is a promise problem under $Y = P$ and $N = \overline{P}$.

All problems addressed in this chapter come with a promise in $\mathbf{P}$. Consider (Y, N) a promise problem where $Y \cup N$ is in $\mathbf{P}$. If Y is in $\mathbf{NP}$, then $\overline{Y}$ is in $\mathbf{coNP}$, and so is $N = (Y \cup N) \cap \overline{Y}$. If for two promise problems (Y_1, N_1) and (Y_2, N_2) we have $Y_1 \subseteq Y_2$ and $N_1 \subseteq N_2$, then $(Y_1, N_1) \preceq_{\mathbf{P}} (Y_2, N_2)$ via the identity function.

It is useful to generalize the notion of 'fragment' in the following way. If $A \subseteq \Sigma^*$ is any decision problem, the problem $A \cap C$ is called the *C-fragment of A*. Then, if B is a decision problem and A is the C-fragment of B for a condition $C \in \mathbf{P}$, then $A \preceq_{\mathbf{P}} B$.

2.4 Classical logic and Boolean algebras

The usual language of classical logic is $\{\neg, \wedge, \vee, \to, \leftrightarrow, \overline{0}, \overline{1}\}$; we refer to these connectives as the *full language of classical logic*. In classical context, either $\wedge$ or $\&$ is used for conjunction and the two are interchangeable. It is convenient to start with some functionally complete subset[9] of the above set and to define the remaining connectives.

Classical propositional logic can be introduced in a lot of ways, e.g., via its well-known Hilbert- and Gentzen-style proof systems; we will be using neither, but we remark that one can obtain classical logic by adding the axiom $\varphi \vee \neg\varphi$ to the axioms of the logic $\mathrm{FL_{ew}}$ or some of its consistent axiomatic extensions, so classical propositional logic is one of the axiomatic extensions of $\mathrm{FL_{ew}}$. Hence the two-element Boolean algebra $\{0,1\}_{\mathrm{B}}$ is a $\mathrm{FL_{ew}}$-algebra; classical propositional logic is just the logic of $\{0,1\}_{\mathrm{B}}$.

The following sets of formulas are important decision problems in classical propositional logic:

DEFINITION 2.4.1. *Let $\mathcal{L}$ be the full language of classical logic.*

$$\mathrm{SAT}(\{0,1\}_{\mathrm{B}}) = \{\varphi \in \mathit{Fm}_{\mathcal{L}} \mid \exists e \in \mathit{Val}(\{0,1\}_{\mathrm{B}})(e(\varphi) = 1^{\{0,1\}_{\mathrm{B}}})\}$$
$$\mathrm{TAUT}(\{0,1\}_{\mathrm{B}}) = \{\varphi \in \mathit{Fm}_{\mathcal{L}} \mid \forall e \in \mathit{Val}(\{0,1\}_{\mathrm{B}})(e(\varphi) = 1^{\{0,1\}_{\mathrm{B}}})\}$$

It is easy to see that $\mathrm{SAT}(\{0,1\}_{\mathrm{B}})$ is in $\mathbf{NP}$: if a propositional formula φ is classically satisfiable, then a simple proof of the fact is a satisfying evaluation of its propositional variables; this is a piece of information of size polynomial in $|\varphi|$, and the ver-

[9] A set $C \subseteq \mathcal{L}$ of connectives is functionally complete w.r.t. an $\mathcal{L}$-algebra $\boldsymbol{A}$ iff for each $n \in \mathsf{N}$, each n-ary function $f \colon A^n \to A$ is definable by a C-formula. We remark that, unlike in the classical case, no set of the above connectives is functionally complete for the algebras in the focus of our attention, i.e., algebras given by (left-)continuous t-norms. The interesting question *which* functions in these algebras are definable by formulas is addressed in Chapter IX of this book.

ification process is clearly a polynomial affair. Moreover, $\varphi \in \mathrm{TAUT}(\{0,1\}_{\mathrm{B}})$ iff $\neg\varphi \in \overline{\mathrm{SAT}}(\{0,1\}_{\mathrm{B}})$, so $\mathrm{TAUT}(\{0,1\}_{\mathrm{B}})$ is in $\mathbf{coNP}$.

Denote $\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$ a fragment of the above SAT problem for formulas in conjunctive normal form, and $\mathrm{TAUT}^{\mathrm{DNF}}(\{0,1\}_{\mathrm{B}})$ a fragment of the above TAUT problem for formulas in disjunctive normal form. (So for both problems, the propositional language is $\{\neg, \wedge, \vee\}$.) In [9], S.A. Cook established a link between propositional logic and computational complexity theory by presenting $\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$ as a first example of an $\mathbf{NP}$-complete problem:

PROPOSITION 2.4.2 (Cook Theorem).
The $\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$ *problem is* $\mathbf{NP}$*-complete.*

To obtain $\mathbf{NP}$-hardness, Cook considered an arbitrary but fixed set of words $S \subseteq \Sigma^*$ accepted by a (nondeterministic) Turing machine in time polynomial in the input size, and presented a polynomial-time procedure which constructed, for each string $s \in \Sigma^*$, a propositional formula φ^s in conjunctive normal form in such a way that $s \in S$ iff $\varphi^s \in \mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$. Hence, $S \preceq_{\mathbf{P}} \mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$, and $\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$ is $\mathbf{NP}$-complete.

$\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$ is a fragment of $\mathrm{SAT}(\{0,1\}_{\mathrm{B}})$, therefore $\mathrm{SAT}(\{0,1\}_{\mathrm{B}})$ is $\mathbf{NP}$-complete and $\mathrm{TAUT}(\{0,1\}_{\mathrm{B}})$ is $\mathbf{coNP}$-complete.

These complexity results extend immediately to $\mathrm{Th}_{\forall}(\{0,1\}_{\mathrm{B}})$, which is $\mathbf{coNP}$-complete, and to $\mathrm{Th}_{\exists}(\{0,1\}_{\mathrm{B}})$, which is $\mathbf{NP}$-complete; this is observed by realizing that an identity $t = s$ on Boolean expressions can be replaced by the equivalence $t \leftrightarrow s$. By this argument, $\mathrm{Th}(\{0,1\}_{\mathrm{B}})$ is polynomially equivalent to the QBF problem, hence $\mathbf{PSPACE}$-complete.

2.5 Decision problems in the reals

We review some decision problems in the reals, and also in the integers, that are relevant to our purpose.[10]

Consider a system $\mathbf{Ax} \leq \mathbf{b}$ of linear inequalities, where $\mathbf{b}$ is a rational m-vector and $\mathbf{A}$ is a rational $m \times n$-matrix. Assume every rational is represented as a pair of coprime integers and denote k the greatest absolute value of an integer occurring in the representations of $\mathbf{A}$ and $\mathbf{b}$.

The problem of solvability of $\mathbf{Ax} \leq \mathbf{b}$ in the reals is in $\mathbf{P}$. Within this chapter though, we will rely on its $\mathbf{NP}$-containment, which can be observed as follows.

Let $P = \{x \in \mathsf{R}^n \mid \mathbf{Ax} \leq \mathbf{b}\}$ be a nonempty polyhedron in R^n. Each nonempty, inclusion-wise minimal face[11] of P is a solution to $\mathbf{A}'\mathbf{x} = \mathbf{b}'$, where $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$ is a subsystem of the system $\mathbf{Ax} \leq \mathbf{b}$, i.e., $\mathbf{A}'$ is an $m' \times n$-matrix and $\mathbf{b}'$ an m'-vector for some $m' \leq m$. Fix a nonempty, minimal face of P; then its corresponding system of equations $\mathbf{A}'\mathbf{x} = \mathbf{b}'$ is solvable in R. Let $m'' \leq m'$ denote the rank of $\mathbf{A}'$, and let $\mathbf{A}''\mathbf{x} \leq \mathbf{b}''$ be a subsystem of $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$ with m'' linearly independent rows. Then

[10] References for the material presented in this subsection are [40] and [16].

[11] A *face* of P is any set $\{\mathbf{x} \in P \mid \mathbf{c}^T\mathbf{x} = d\}$ for $\mathbf{c} \in \mathsf{R}^n, d \in \mathsf{R}$ chosen in such a way that $\mathbf{c}^T\mathbf{x} \leq d$ holds for all $\mathbf{x} \in P$. A face is *minimal* if it does not contain any other face; a minimal face is an affine subspace of R^n. Since the list of all nonempty faces of (nonempty) P is finite, at least one nonempty, inclusion-wise minimal face exists.

$\mathbf{b}''$ is a linear combination of m'' columns of $\mathbf{A}''$, and hence, there is a solution $\mathbf{x} = \langle x_1, \ldots, x_n \rangle \in \mathsf{R}^n$ to $\mathbf{A}''\mathbf{x} = \mathbf{b}''$ where at most m'' of the x_i's are nonzero. Use Cramer's rule to compute the nonzero values of $\mathbf{x}$ (write m instead of m''). For each determinant, its denominator is at most the product of all denominators in $\mathbf{A}''$; if the largest one is k, then the product is at most k^{m^2}, i.e., of size at most $m^2 \log(k)$. The numerator, being a sum of $m!$ numbers bounded by k^{m^2}, is of size at most $m \log(m) + m^2 \log(k)$. Hence, any of the x_i's, as a fraction of two determinants, is of size at most $O(m \log(m) + m^2 \log(k))$.

Summing up, we arrive at the following statement (where instead of 'model' one might say 'solution' or 'evaluation'):

PROPOSITION 2.5.1 (Small-Model Theorem). *Let $\mathbf{A}$ be a rational $m \times n$-matrix and $\mathbf{b}$ a rational m-vector. Let k be the greatest integer occurring in the representations of $\mathbf{A}$ and $\mathbf{b}$. If the system $\mathbf{Ax} \leq \mathbf{b}$ is solvable in R, then it has a rational solution $\mathbf{x}_0$ with the following properties:*

(i) at most m values in $\mathbf{x}_0$ are nonzero;

(ii) any value in $\mathbf{x}_0$ has size polynomial in $|\mathbf{A}|$, $|\mathbf{b}|$; in particular, for $i = 1, \ldots, n$, $|(\mathbf{x}_0)_i|$ is in $O(m \log(m) + m^2 \log(k))$.

Linear Programming Problem. The linear programming (LP) problem[12] is defined as follows: given a rational $m \times n$-matrix $\mathbf{A}$, a rational m-vector $\mathbf{b}$, a rational n-vector $\mathbf{c}$, and a rational number d, does the system $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{c}^T\mathbf{x} < d$ have a solution in R? Again, while the problem is in $\mathbf{P}$, we need its $\mathbf{NP}$-containment. It is not difficult to see that the added strict inequality does not violate the validity of the small-model theorem above, where of course now k relates also to the integers in the representation of $\mathbf{c}$ and d as well. Indeed, given a solvable LP problem in the above notation, the halfspace $\mathbf{c}^T\mathbf{x} < d$ either contains a minimal face of $\mathbf{Ax} \leq \mathbf{b}$; or it intersects one, and then (by minimality) this face is unbounded and it contains some points that are bounded in size by the coefficients in $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{c}^T\mathbf{x} < d$ in the manner of the small-model theorem.

Modifications of the LP problem are obtained by posing various restrictive conditions; these modifications need not be feasible. In particular, the *integer programming problem*, here referred to as the ILP problem, is obtained by demanding that all variables and coefficients assume integer values. This problem is $\mathbf{NP}$-complete. Containment in $\mathbf{NP}$ can be derived from a small-model theorem for Diophantine equations and inequalities: let $\mathbf{Ax} \leq \mathbf{b}$ be a system of inequalities, where $\mathbf{A}$ is an integral $m \times n$-matrix and $\mathbf{b}$ is an integral m-vector, where the largest absolute value of an integer is k. If the system is solvable in Z, then it has a solution $\mathbf{x}_0$, where for any $1 \leq i \leq n$, $|(\mathbf{x}_0)_i|$ is in $O(m \log(m) + m \log(k))$. The *mixed integer programming* (MIP) problem is a modification of the LP problem demanding that a subset of the variables assume integer values. Particular bounded version of MIP poses the restrictive condition that the variables $x_k, \ldots, x_n$ only take the values 0 or 1. This is also in $\mathbf{NP}$: guess a random assignment of 0's and 1's to $x_k, \ldots, x_n$, then check solvability of the remaining system.

[12] Quite often, the phrase 'linear programming problem' denotes the *task* to either find a maximum of a function $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}$, or to say that none exists. We take the standpoint that a 'problem' is always a decision problem; the optimization task will not be considered in this chapter, so no confusion can arise.

Boolean combinations of linear inequalities. The linear programming problem comes as a conjunction of linear inequalities. Arbitrary Boolean combinations[13] of linear inequalities are apparently more difficult to solve. Consider basic inequalities of the form $\mathbf{a}^T\mathbf{x} \leq b$ for a rational n-vector $\mathbf{a}$ and a rational number b; an *inequality formula* is a Boolean combination of basic inequalities. The INEQ problem is: given an inequality formula, is it solvable in the reals? The fact that the LP problem is in $\mathbf{NP}$ entails $\mathbf{NP}$-containment for the INEQ problem: each inequality formula has a logically equivalent disjunctive normal form, which is solvable in the reals iff so is at least one of its disjuncts. Each of the disjuncts can be equivalently transformed into an LP problem (negative literals will use the strict inequality in the LP problem). Because the solvability of the LP problem can be witnessed by a small evaluation, so can the solvability of an INEQ problem. On the other hand, INEQ is $\mathbf{NP}$-hard because classical SAT can be reduced to it, so it is $\mathbf{NP}$-complete.

Universal theory of RCF. Now let us consider the language of ordered fields, i.e., $\{+, \cdot, 0, 1, =, \leq\}$. Recall that real numbers R with addition, multiplication, and the usual ordering, are an example of a real closed field (RCF). The first-order theory of real closed fields $(\mathrm{Th}(\mathrm{RCF}))$ is complete; hence, each two real closed fields are elementarily equivalent. Moreover, $\mathrm{Th}(\mathrm{RCF})$ is decidable. Both results were proved by A. Tarski by quantifier elimination. J.F. Canny has shown that the existential fragment of the RCF theory is in $\mathbf{PSPACE}$ [7]. As $\mathbf{PSPACE}$ is closed under complementation, also the universal fragment of the RCF theory is in $\mathbf{PSPACE}$. We denote these fragments $\mathrm{Th}_\exists(\mathrm{RCF})$ and $\mathrm{Th}_\forall(\mathrm{RCF})$, respectively.

3 General results and methods

This section presents general statements, applicable for particular examples of logics; while all of the statements are, to a degree, a prerequisite to reading the following sections, this is perhaps especially true about Subsection 3.3, where some technical statements and notation are introduced without which the following sections might be incomprehensible.

Languages in this section are assumed to subsume the language of $\mathrm{FL_{ew}}$, logics are assumed to be at least as strong as $\mathrm{FL_{ew}}$ (with maybe additional connectives, and some connectives definable) and algebras are assumed to be $\mathrm{FL_{ew}}$-algebras (with maybe additional operations). Classes of algebras are assumed nonempty.

3.1 Basic inclusions and reductions

For the next two statements, the discussion of fragments in Subsection 2.3 is relevant.

LEMMA 3.1.1. *Let $\mathcal{L}$ be a language,* L *a logic in the language $\mathcal{L}$, and $\mathbb{K}$ a class of $\mathcal{L}$-algebras. Then*

(i) $\mathrm{THM}(\mathrm{L}) \preceq_{\mathbf{P}} \mathrm{CONS}(\mathrm{L})$; *if* L *enjoys the classical (or the $\triangle$-) deduction theorem, then* $\mathrm{THM}(\mathrm{L}) \approx_{\mathbf{P}} \mathrm{CONS}(\mathrm{L})$;

(ii) $\mathrm{Th_{Eq}}(\mathbb{K}) \preceq_{\mathbf{P}} \mathrm{Th_{QEq}}(\mathbb{K}) \preceq_{\mathbf{P}} \mathrm{Th}_\forall(\mathbb{K}) \preceq_{\mathbf{P}} \mathrm{Th}(\mathbb{K})$ *and* $\mathrm{Th}_\exists(\mathbb{K}) \preceq_{\mathbf{P}} \mathrm{Th}(\mathbb{K})$;

[13] I.e., a formula with any connectives of the full language of classical logic.

(iii) $\mathrm{TAUT}(\mathbb{K}) \approx_{\mathbf{P}} \mathrm{Th}_{\mathrm{Eq}}(\mathbb{K})$ *and* $\mathrm{CONS}(\mathbb{K}) \approx_{\mathbf{P}} \mathrm{Th}_{\mathrm{QEq}}(\mathbb{K})$;

(iv) $\mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K}) \preceq_{\mathbf{P}} \mathrm{Th}_{\mathrm{QEq}}(\mathbb{K})$;

(v) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) \preceq_{\mathbf{P}} \mathrm{Th}_{\exists}(\mathbb{K})$;

(vi) $\mathrm{Th}_{\forall}(\mathbb{K}) \approx_{\mathbf{P}} \overline{\mathrm{Th}_{\exists}}(\mathbb{K})$.

Proof. In the following, consider an $\mathcal{L}$-expression φ with n variables $x_1, \dots, x_n$.
(i) THM(L) is the fragment of CONS(L) obtained by considering only empty theories. If L enjoys the classical (or the $\triangle$-) deduction theorem, then $\{\psi_1, \dots, \psi_n\} \vdash_{\mathrm{L}} \varphi$ iff $\vdash_{\mathrm{L}} \psi_1 \,\&\, \dots \,\&\, \psi_n \to \varphi$ ($\vdash_{\mathrm{L}} \triangle(\psi_1 \,\&\, \dots \,\&\, \psi_n) \to \varphi$ respectively).
(ii) In all cases, we are dealing with fragments defined by polynomial-time conditions.
(iii) $\varphi \in \mathrm{TAUT}(\mathbb{K})$ iff $(\varphi = 1) \in \mathrm{Th}_{\mathrm{Eq}}(\mathbb{K})$; on the other hand, $(\varphi = \psi) \in \mathrm{Th}_{\mathrm{Eq}}(\mathbb{K})$ iff $(\varphi \leftrightarrow \psi) \in \mathrm{TAUT}(\mathbb{K})$. Analogously for CONS and quasiidentities.
(iv) $\varphi \in \mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K})$ iff $\forall x_1 \dots \forall x_n(\varphi = 0 \to 0 = 1) \in \mathrm{Th}_{\mathrm{QEq}}(\mathbb{K})$.
(v) $\varphi \in \mathrm{SAT}(\mathbb{K})$ iff $\exists x_1 \dots \exists x_n(\varphi = 1) \in \mathrm{Th}_{\exists}(\mathbb{K})$; analogously, $\varphi \in \mathrm{SAT}_{\mathrm{pos}}(\mathbb{K})$ iff $\exists x_1 \dots \exists x_n(\varphi > 0) \in \mathrm{Th}_{\exists}(\mathbb{K})$.
(vi) A consequence of classical duality of quantifiers. □

LEMMA 3.1.2. *Let* $\mathcal{L}_1 \subseteq \mathcal{L}_2$ *be languages.*

(i) Assume a logic L_2 *in language* $\mathcal{L}_2$ *expands conservatively a logic* L_1 *in language* $\mathcal{L}_1$ *(so* L_1 *is the* $\mathcal{L}_1$*-fragment of* L_2*). Then* $\mathrm{THM}(\mathrm{L}_1) \preceq_{\mathbf{P}} \mathrm{THM}(\mathrm{L}_2)$ *and* $\mathrm{CONS}(\mathrm{L}_1) \preceq_{\mathbf{P}} \mathrm{CONS}(\mathrm{L}_2)$.

(ii) Assume $\mathbb{K}_2$ *is a class of* $\mathcal{L}_2$*-algebras and* $\mathbb{K}_1$ *is the class of* $\mathcal{L}_1$*-reducts of elements of* $\mathbb{K}_2$*. Then* $\mathrm{Th}(\mathbb{K}_1) \preceq_{\mathbf{P}} \mathrm{Th}(\mathbb{K}_2)$*, and analogously for the equational, quasiequational, universal and existential fragments of the two theories.*

LEMMA 3.1.3. *Let* $\mathcal{L}$ *be a language and* $\mathbb{K}$, $\mathbb{L}$ *classes of* $\mathcal{L}$*-algebras.*

(i) Assume $\mathbb{K} \subseteq \mathbb{L}$. *Then*

  (a) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) \subseteq \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{L})$;

  (b) $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{L}) \subseteq \mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$ *and* $\mathrm{CONS}(\mathbb{L}) \subseteq \mathrm{CONS}(\mathbb{K})$.

(ii) Assume $\mathbb{K}$ *is (partially) embeddable into* $\mathbb{L}$. *Then*

  (a) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) \subseteq \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{L})$;

  (b) $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{L}) \subseteq \mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$ *and* $\mathrm{CONS}(\mathbb{L}) \subseteq \mathrm{CONS}(\mathbb{K})$.

Proof. (i) holds by definition. For (ii), it suffices to recall that 0 is an element of $\mathcal{L}$ and hence preserved by morphisms, and the same is true for $1 = 0 \to 0$. □

LEMMA 3.1.4. *Let* $\mathcal{L}$ *be a language and* $\mathbb{K}$ *a class of* $\mathcal{L}$*-algebras containing a nontrivial algebra. Then*

(i) $\mathrm{SAT}(\{0,1\}_{\mathrm{B}}) \subseteq \mathrm{SAT}(\mathbb{K}) \subseteq \mathrm{SAT}_{\mathrm{pos}}(\mathbb{K})$;

(ii) $\mathrm{TAUT}(\mathbb{K}) \subseteq \mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K}) \subseteq \mathrm{TAUT}(\{0,1\}_{\mathrm{B}})$;

(iii) $\mathrm{CONS}(\mathbb{K}) \subseteq \mathrm{CONS}(\{0,1\}_{\mathrm{B}})$.

Proof. For any nontrivial $\mathcal{L}$-algebra $\boldsymbol{A} \in \mathbb{K}$, its subalgebra $\{0^{\boldsymbol{A}}, 1^{\boldsymbol{A}}\}$ is a two-element Boolean algebra. So $\{0,1\}_{\mathrm{B}}$ is embeddable into $\boldsymbol{A}$, the class $\{\{0,1\}_{\mathrm{B}}\}$ is embeddable into $\mathbb{K}$, and Lemma 3.1.3 applies. □

LEMMA 3.1.5. *Let $\mathcal{L}$ be a language and $\mathbb{K}$ a class of $\mathcal{L}$-algebras. Then*

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbf{I}(\mathbb{K}))$ *and* $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{TAUT}_{(\mathrm{pos})}(\mathbf{I}(\mathbb{K}))$;

(ii) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbf{S}(\mathbb{K}))$ *and* $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{TAUT}_{(\mathrm{pos})}(\mathbf{S}(\mathbb{K}))$;

(iii) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbf{P}(\mathbb{K}))$ *and* $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{TAUT}_{(\mathrm{pos})}(\mathbf{P}(\mathbb{K}))$;

(iv) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbf{P}_{\mathrm{U}}(\mathbb{K}))$ *and* $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{TAUT}_{(\mathrm{pos})}(\mathbf{P}_{\mathrm{U}}(\mathbb{K}))$;

(v) $\mathrm{SAT}_{\mathrm{pos}}(\mathbb{K}) = \mathrm{SAT}_{\mathrm{pos}}(\mathbf{H}(\mathbb{K}))$ *and* $\mathrm{TAUT}(\mathbb{K}) = \mathrm{TAUT}(\mathbf{H}(\mathbb{K}))$.

Proof. For the SAT operators, the left-to-right inclusions are obtained by virtue of $\mathbf{I}$, $\mathbf{S}$, $\mathbf{P}$, $\mathbf{P}_{\mathrm{U}}$, $\mathbf{H}$ being closure operators (in view of Lemma 3.1.3); for the TAUT operators, the converse inclusions hold by the same argument. Indeed for the TAUT operator, equality for all cases is well known. In the following, we set to show the remaining inclusions.
(i) Immediate.
(ii) Let $\boldsymbol{B}$ be a subalgebra of $\boldsymbol{A} \in \mathbb{K}$. Then $\boldsymbol{B}$ is embeddable into $\boldsymbol{A}$ (via identity mapping). Therefore $\mathbf{S}(\mathbb{K})$ is embeddable into $\mathbb{K}$. An application of Lemma 3.1.3 yields the desired inclusions.
(iii) Let $\boldsymbol{B} = \prod_{i \in I} \boldsymbol{A}_i$, where $I \neq \emptyset$ and $\boldsymbol{A}_i \in \mathbb{K}$ for each $i \in I$. If $e_{\boldsymbol{B}}(\varphi) = 1^{\boldsymbol{B}}$ ($e_{\boldsymbol{B}}(\varphi) > 0^{\boldsymbol{B}}$), define $e_{\boldsymbol{A}_i}(x) = \pi_i(e_{\boldsymbol{B}}(x))$ for each $i \in I$ (where π_i is the i-th projection); then $e_{\boldsymbol{A}_i}$ is an evaluation in $\boldsymbol{A}_i$ for each $i \in I$. Clearly $e_{\boldsymbol{A}_i}(\varphi) = 1^{\boldsymbol{A}_i}$ for each $i \in I$ ($e_{\boldsymbol{A}_i}(\varphi) > 0$ for some $i \in I$ respectively), so $\varphi \in \mathrm{SAT}(\boldsymbol{A}_i)$ for each $i \in I$ ($\varphi \in \mathrm{SAT}_{\mathrm{pos}}(\boldsymbol{A}_i)$ for some $i \in I$ respectively). Likewise, if $e_{\boldsymbol{A}_i}(\varphi) > 0^{\boldsymbol{A}_i}$ for each $i \in I$ and each evaluation $e_{\boldsymbol{A}_i}$, then in particular, for an arbitrary evaluation $e_{\boldsymbol{B}}$ in $\boldsymbol{B}$, we have $\pi_i(e_{\boldsymbol{B}}(\varphi)) > 0^{\boldsymbol{A}_i}$, hence $e_{\boldsymbol{B}}(\varphi) > 0^{\boldsymbol{B}}$.
(iv) Let $\boldsymbol{B} = \prod_{i \in I}^{\mathcal{F}} \boldsymbol{A}_i$, where $I \neq \emptyset$, $\mathcal{F}$ is an ultrafilter on I, and $\boldsymbol{A}_i \in \mathbb{K}$ for each $i \in I$. If $e_{\boldsymbol{B}}(\varphi) = 1^{\boldsymbol{B}}$, let, for each $x \in \mathit{Var}$, $e(x)$ be any $f \in [e_{\boldsymbol{B}}(x)]_{\mathcal{F}}$, and for each $i \in I$, define $e_{\boldsymbol{A}_i}(x) = \pi_i(e(x))$; then $e_{\boldsymbol{A}_i}$ is an evaluation in $\boldsymbol{A}_i$ for each $i \in I$. We have $\{i \mid e_{\boldsymbol{A}_i}(\varphi) = 1^{\boldsymbol{A}_i}\} \in \mathcal{F}$, hence for some $i \in I$ we have $\varphi \in \mathrm{SAT}(\boldsymbol{A}_i)$. Assuming $e_{\boldsymbol{B}}(\varphi) > 0^{\boldsymbol{B}}$ ($e_{\boldsymbol{B}}(\varphi) < 1^{\boldsymbol{B}}$, $e_{\boldsymbol{B}}(\varphi) = 0^{\boldsymbol{B}}$), one gets in the same manner $\varphi \in \mathrm{SAT}_{\mathrm{pos}}(\boldsymbol{A}_i)$ ($\varphi \notin \mathrm{TAUT}(\boldsymbol{A}_i)$, $\varphi \notin \mathrm{TAUT}_{\mathrm{pos}}(\boldsymbol{A}_i)$ respectively) for some $i \in I$.
(v) Let $\boldsymbol{B}$ be a nontrivial homomorphic image of $\boldsymbol{A} \in \mathbb{K}$ via an f. Assume $e_{\boldsymbol{B}}(\varphi) > 0^{\boldsymbol{B}}$. Take $e_{\boldsymbol{A}}(x) \in f^{-1}(e_{\boldsymbol{B}}(x))$ for each variable x; then $e_{\boldsymbol{A}}(\varphi) > 0^{\boldsymbol{A}}$. □

On this basis, we may conclude:

THEOREM 3.1.6. *Let $\mathcal{L}$ be a language and $\mathbb{K}$ a class of $\mathcal{L}$-algebras. Then*

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbf{Q}(\mathbb{K}))$;

(ii) $\mathrm{SAT}_{\mathrm{pos}}(\mathbb{K}) = \mathrm{SAT}_{\mathrm{pos}}(\mathbf{V}(\mathbb{K}))$;

(iii) $\mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K}) = \mathrm{TAUT}_{\mathrm{pos}}(\mathbf{Q}(\mathbb{K}))$;

(iv) $\mathrm{TAUT}(\mathbb{K}) = \mathrm{TAUT}(\mathbf{V}(\mathbb{K})) = \mathrm{TAUT}(\mathbf{Q}(\mathbb{K}))$.

The previous theorem says that, in contrast to the case of first-order fuzzy logics, one need not worry about the general/standard semantics distinction in the propositional case. For important examples of logics L considered in this chapter, their equivalent algebraic semantics is a variety that is generated by its standard members (as a quasivariety); then one can apply the above theorem to relate the results obtained on standard algebras also to the general semantics.

3.2 Negations

In classical logic, there is a duality between the SAT and TAUT problems, in the manner of Lemma 3.1.1 (vi). We inspect the conditions under which as much, or at least some of that, may be claimed in a many-valued setting.

A negation $\sim$ in L is *involutive* iff $\sim\sim\varphi \leftrightarrow \varphi$ is a theorem of L; the semantics of $\sim$ is an order-reversing involution. A negation $\neg$ in L is *strict* iff $\neg(\varphi \wedge \neg\varphi)$ is a theorem of L. If both the involutive negation $\sim$ and the strict negation $\neg$ are available in L, then, defining $\triangle\varphi$ as $\neg\sim\varphi$, one can prove the usual axioms for the $\triangle$ connective.

LEMMA 3.2.1. *Let $\mathcal{L}$ be a language and $\mathbb{K}$ a class of $\mathcal{L}$-algebras containing a nontrivial algebra. Then*

(i) $\varphi \in \mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K})$ *iff* $\neg\varphi \in \overline{\mathrm{SAT}}(\mathbb{K})$;

(ii) $\varphi \in \mathrm{SAT}_{\mathrm{pos}}(\mathbb{K})$ *iff* $\neg\varphi \in \overline{\mathrm{TAUT}}(\mathbb{K})$.

If, additionally, $\sim$ is an involutive negation in $\mathbb{K}$, then

(iii) $\varphi \in \mathrm{TAUT}(\mathbb{K})$ *iff* $\sim\varphi \in \overline{\mathrm{SAT}_{\mathrm{pos}}}(\mathbb{K})$;

(iv) $\varphi \in \mathrm{SAT}(\mathbb{K})$ *iff* $\sim\varphi \in \overline{\mathrm{TAUT}_{\mathrm{pos}}}(\mathbb{K})$.

Proof. Items (i), (ii) are easily obtained by observing that in any nontrivial $\mathrm{FL}_{\mathrm{ew}}$-algebra $\boldsymbol{A}$, the equation $\neg x = 1$ has a unique solution, $x = 0^{\boldsymbol{A}}$.
Items (iii), (iv) follow from (i), (ii) by substituting $\sim\varphi$ for φ and using $\sim\sim\varphi \leftrightarrow \varphi$. □

COROLLARY 3.2.2. *Let $\mathcal{L}$ be a language and $\mathbb{K}$ be a class of involutive $\mathcal{L}$-algebras containing a nontrivial algebra. Then*

(i) $\mathrm{TAUT}(\mathbb{K}) \approx_{\mathbf{P}} \overline{\mathrm{SAT}_{\mathrm{pos}}}(\mathbb{K})$;

(ii) $\mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K}) \approx_{\mathbf{P}} \overline{\mathrm{SAT}}(\mathbb{K})$.

Now we explore logics with strict negations.

LEMMA 3.2.3. *Let $\boldsymbol{A}$ be a nontrivial* SMTL-*chain. Then $\{0,1\}_{\mathrm{B}}$ is a homomorphic image of $\boldsymbol{A}$.*

Proof. The mapping $h\colon \boldsymbol{A} \to \boldsymbol{A}$ sending $0^{\boldsymbol{A}}$ to $0^{\boldsymbol{A}}$ and all nonzero elements to $1^{\boldsymbol{A}}$ is a homomorphism of $\boldsymbol{A}$ onto the two-element Boolean subalgebra $\{0^{\boldsymbol{A}}, 1^{\boldsymbol{A}}\}$ of $\boldsymbol{A}$. □

THEOREM 3.2.4. *Let $\mathbb{K}$ be a class of* SMTL-*chains containing a nontrivial one. Then*

(i) $\mathrm{SAT}_{\mathrm{pos}}(\mathbb{K}) = \mathrm{SAT}(\mathbb{K}) = \mathrm{SAT}(\{0,1\}_{\mathrm{B}})$;

(ii) $\mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K}) = \mathrm{TAUT}(\{0,1\}_{\mathrm{B}})$.

Proof. (i) By Lemma 3.1.4, it is sufficient to show that any formula φ positively satisfiable in $\mathbb{K}$ is classically satisfiable. Let for some $\boldsymbol{A} \in \mathbb{K}$ and some $e_{\boldsymbol{A}}$ be $e_{\boldsymbol{A}}(\varphi) > 0^{\boldsymbol{A}}$. Define e' on $\{0,1\}_{\mathrm{B}}$ using the homomorphism from Lemma 3.2.3: for any formula ψ, let $e'_{\boldsymbol{A}}(\psi) = h(e_{\boldsymbol{A}}(\psi))$. Clearly e' is a well-defined evaluation in $\{0,1\}_{\mathrm{B}}$ and $e'(\varphi) = 1^{\{0,1\}_{\mathrm{B}}}$ iff $e_{\boldsymbol{A}}(\varphi) > 0^{\boldsymbol{A}}$.
(ii) Again it is sufficient to show $\mathrm{TAUT}^{\mathrm{Boole}} \subseteq \mathrm{TAUT}^{\boldsymbol{A}}_{\mathrm{pos}}$ for any nontrivial $\boldsymbol{A} \in \mathbb{K}$. For φ a classical tautology, assume $e_{\boldsymbol{A}}(\varphi) = 0^{\boldsymbol{A}}$ in $\boldsymbol{A}$; but then $e'(\varphi) = 0$ in $\{0,1\}_{\mathrm{B}}$ (e' as above), a contradiction; hence $\varphi \in \mathrm{TAUT}_{\mathrm{pos}}(\boldsymbol{A})$. □

3.3 Eliminating compound terms

We start with feasible translations of expressions that preserve satisfiability or tautologousness but eliminate (some) nested connectives in a particular way, at the expense of adding new variables.

Consider a finite language $\mathcal{L}$ and an $\mathcal{L}$-expression $\varphi(x_1, \dots, x_n)$, where $n \geq 1$. To each subexpression ψ of φ, assign a variable y_ψ in the following manner:

– if ψ is a variable x_i, then let y_ψ be the variable x_i;
– otherwise, let y_ψ be a new variable.

For each $\psi \preceq \varphi$, let S_ψ denote the set of all subexpressions of ψ that are not variables. For each $\psi \in S_\varphi$, if ψ is $c(\psi_1, \dots, \psi_k)$ for some $k \in \mathsf{N}$, some $\psi_1, \dots, \psi_k \preceq \varphi$, and some $c \in \mathcal{L}$, let C_ψ stand for $y_\psi \leftrightarrow c(y_{\psi_1}, \dots, y_{\psi_k})$. For each $\psi \preceq \varphi$, let ψ' be the expression $\&_{\chi \in S_\psi} C_\chi$. Then for each $\psi \preceq \varphi$, if ψ is $c(\psi_1, \dots, \psi_k)$ for some $k \in \mathsf{N}$, some $\psi_1, \dots, \psi_k \preceq \psi$, and some $c \in \mathcal{L}$, then ψ' is $\psi'_1 \mathbin{\&} \dots \mathbin{\&} \psi'_k \mathbin{\&} C_\psi$; if ψ is a variable, then ψ' is $\overline{1}$.

Observe that φ' can be obtained from φ in time polynomial in $|\varphi|$: indeed, for each $\psi \preceq \varphi$, $|C_\psi|$ is in $O(\log(|\varphi|))$, and the number of C_ψ's in φ' is bounded by the number of subexpressions in φ, i.e., by $|\varphi|$.

LEMMA 3.3.1. *For each* $\mathrm{FL_{ew}}$*-expression* φ *we have* $\vdash_{\mathrm{FL_{ew}}} \varphi' \to (y_\varphi \leftrightarrow \varphi)$.

Proof. By induction on formula structure. The cases of ψ being a variable or the constant $\overline{0}$ are simple. For the induction step, let $\psi \preceq \varphi$ be $\psi_1 \circ \psi_2$ for $\circ$ one of $\{\&, \to, \wedge, \vee\}$; the induction assumption is $\psi'_i \to (y_{\psi_i} \leftrightarrow \psi_i)$ for $i = 1, 2$. We obtain $\psi'_1 \mathbin{\&} \psi'_2 \mathbin{\&} C_\psi \to (y_{\psi_1} \leftrightarrow \psi_1) \mathbin{\&} (y_{\psi_2} \leftrightarrow \psi_2) \mathbin{\&} C_\psi$. The antecedent of this implication is ψ', while the succedent implies[14] $((y_{\psi_1} \circ y_{\psi_2}) \leftrightarrow (\psi_1 \circ \psi_2)) \mathbin{\&} (y_\psi \leftrightarrow (y_{\psi_1} \circ y_{\psi_2}))$, whence $y_\psi \leftrightarrow \psi$. □

LEMMA 3.3.2. *Let* $\mathbb{K}$ *a class of* $\mathrm{FL_{ew}}$*-algebras,* φ *a* $\mathrm{FL_{ew}}$*-expression, and* φ' *as above. Then*

(i) $\varphi \in \mathrm{TAUT}(\mathbb{K})$ *iff* $\varphi' \to y_\varphi \in \mathrm{TAUT}(\mathbb{K})$;

(ii) $\varphi \in \mathrm{TAUT_{pos}}(\mathbb{K})$ *iff* $\varphi' \to y_\varphi \in \mathrm{TAUT_{pos}}(\mathbb{K})$;

(iii) $\varphi \in \mathrm{SAT}(\mathbb{K})$ *iff* $\varphi' \mathbin{\&} y_\varphi \in \mathrm{SAT}(\mathbb{K})$;

(iv) $\varphi \in \mathrm{SAT_{pos}}(\mathbb{K})$ *iff* $\varphi' \mathbin{\&} y_\varphi \in \mathrm{SAT_{pos}}(\mathbb{K})$.

[14]Using $((\alpha_1 \leftrightarrow \beta_1) \mathbin{\&} (\alpha_2 \leftrightarrow \beta_2)) \to ((\alpha_1 \circ \alpha_2) \leftrightarrow (\beta_1 \circ \beta_2))$ for $\circ$ one of $\{\&, \to, \wedge, \vee\}$; if this congruence is provable for other connectives, then the statement extends to formulas with these connectives.

Proof. (i) If $\varphi \in \mathrm{TAUT}(\mathbb{K})$, then by Lemma 3.3.1 $\varphi' \to (y_\varphi \leftrightarrow \overline{1})$ is a tautology of $\mathbb{K}$. On the other hand, if $\varphi(x_1, \ldots, x_n)$ is not a tautology of $\mathbb{K}$, there is an $\boldsymbol{A} \in \mathbb{K}$ and an evaluation $e_{\boldsymbol{A}}$ such that $e_{\boldsymbol{A}}(\varphi) < 1^{\boldsymbol{A}}$; define $e'_{\boldsymbol{A}}(x_i) = e_{\boldsymbol{A}}(x_i)$ for $1 \leq i \leq n$, and if $\psi \preceq \varphi$, set $e'_{\boldsymbol{A}}(y_\psi) = e_{\boldsymbol{A}}(\psi)$. Then clearly $e'_{\boldsymbol{A}}(\varphi') = 1^{\boldsymbol{A}}$, while $e'_{\boldsymbol{A}}(y_\varphi) < 1^{\boldsymbol{A}}$.
(ii) If $\varphi \in \overline{\mathrm{TAUT}_{\mathrm{pos}}}(\mathbb{K})$, clearly $\varphi' \to y_\varphi \in \overline{\mathrm{TAUT}_{\mathrm{pos}}}(\mathbb{K})$. Conversely, let $\varphi' \to y_\varphi \in \overline{\mathrm{TAUT}_{\mathrm{pos}}}(\mathbb{K})$, so $e_{\boldsymbol{A}}(\varphi' \to y_\varphi) = 0^{\boldsymbol{A}}$ for some $\boldsymbol{A} \in \mathbb{K}$ and some $e_{\boldsymbol{A}}$. By Lemma 3.3.1, $e_{\boldsymbol{A}}(\varphi' \to (\varphi \to y_\varphi)) = 1^{\boldsymbol{A}}$, so $e_{\boldsymbol{A}}(\varphi \to (\varphi' \to y_\varphi)) = 1^{\boldsymbol{A}}$, and hence $e_{\boldsymbol{A}}(\varphi) = 0^{\boldsymbol{A}}$, so $\varphi \in \overline{\mathrm{TAUT}_{\mathrm{pos}}}(\mathbb{K})$.
(iii) If $\varphi \in \mathrm{SAT}(\mathbb{K})$, clearly $\varphi' \,\&\, y_\varphi \in \mathrm{SAT}(\mathbb{K})$. Conversely, for $\boldsymbol{A} \in \mathbb{K}$, if $e_{\boldsymbol{A}}(\varphi') = 1^{\boldsymbol{A}}$, then in particular $e_{\boldsymbol{A}}(y_\psi) = c^{\boldsymbol{A}}(e_{\boldsymbol{A}}(y_{\psi_1}), \ldots, e_{\boldsymbol{A}}(y_{\psi_k})) = e_{\boldsymbol{A}}(\psi)$ whenever $\psi \preceq \varphi$ is $c(\psi_1, \ldots, \psi_k)$. We have $e_{\boldsymbol{A}}(y_\varphi) = e_{\boldsymbol{A}}(\varphi) = 1^{\boldsymbol{A}}$, hence $\varphi \in \mathrm{SAT}(\mathbb{K})$.
(iv) If $\varphi \in \mathrm{SAT}_{\mathrm{pos}}(\mathbb{K})$, clearly $\varphi' \,\&\, y_\varphi \in \mathrm{SAT}_{\mathrm{pos}}(\mathbb{K})$. Conversely, if $\varphi \in \overline{\mathrm{SAT}_{\mathrm{pos}}}(\mathbb{K})$, then $\varphi \leftrightarrow \overline{0}$ is a tautology of $\mathbb{K}$ and hence so is $\varphi' \to (y_\varphi \leftrightarrow \overline{0})$, using Lemma 3.3.1. Hence $\varphi' \,\&\, y_\varphi \to \overline{0}$ is a tautology of $\mathbb{K}$, so $\varphi' \,\&\, y_\varphi$ is unsatisfiable in $\mathbb{K}$. □

THEOREM 3.3.3. *Let $\mathbb{K}$ be a class of $\mathrm{FL_{ew}}$-algebras. Assume $c \in \mathcal{L}$ is term-definable in $\mathbb{K}$, c is not among $\{\&, \to\}$, and let $\mathbb{K}^{c^-}$ be the class of $\mathcal{L} \setminus \{c\}$-reducts of $\mathbb{K}$. Then*

- *(i)* $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) \approx_{\mathbf{P}} \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}^{c^-})$;
- *(ii)* $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}) \approx_{\mathbf{P}} \mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}^{c^-})$.

Proof. We give the proof for TAUT; for the other operators it is analogous. Clearly $\mathrm{TAUT}(\mathbb{K}^{c^-}) \preceq_{\mathbf{P}} \mathrm{TAUT}(\mathbb{K})$ (cf. Lemma 3.1.2). Conversely, if φ is an $\mathcal{L}$-formula, then $\varphi \in \mathrm{TAUT}(\mathbb{K})$ iff $\varphi' \to y_\varphi \in \mathrm{TAUT}(\mathbb{K})$; apply to φ' the desired translation, i.e., if the identity $c(x_1, \ldots, x_k) = \chi(x_1, \ldots, x_k)$ holds in $\mathbb{K}$ for some $\mathcal{L} \setminus \{c\}$-term χ, replace each occurrence of $y_\psi \leftrightarrow c(y_{\psi_1}, \ldots, y_{\psi_k})$ with $y_\psi \leftrightarrow \chi(y_{\psi_1}, \ldots, y_{\psi_k})$. Replace each occurrence of $\leftrightarrow$ using $\&$ and $\to$; denote the resulting formula φ''. Then $\varphi' \to y_\varphi \in \mathrm{TAUT}(\mathbb{K})$ iff $\varphi'' \to y_\varphi \in \mathrm{TAUT}(\mathbb{K}^{c^-})$, and moreover φ'' can be obtained from φ' by a procedure operating in time polynomial in $|\varphi'|$. □

Further we eliminate compound terms from first-order formulas; in particular, we work with existential sentences (using the technique, one can also eliminate compound terms from universal sentences, whose negations are existential sentences).

DEFINITION 3.3.4. *Let Φ be a first-order formula in a language $\mathcal{L}$. We say that Φ is* without compound terms *iff each atomic formula in Φ is either $t_0 = t_1$ or $t_0 \leq t_1$ or $t_0 < t_1$, there being an $i \in \{0, 1\}$ such that t_i is a variable while t_{1-i} is a variable or a term $f(x_1, \ldots, x_n)$ for some n-ary function symbol $f \in \mathcal{L}$ and some variables $x_1, \ldots, x_n$.*

LEMMA 3.3.5. *Let $\mathcal{L}$ be a language, let Φ be an existential $\mathcal{L}$-sentence. Then there is an existential $\mathcal{L}$-sentence Φ', such that:*

- *(i)* *Φ' is of the form $\exists x_1 \ldots \exists x_k(\Phi_1 \wedge \Phi_2)$, where $\Phi_1 \wedge \Phi_2$ is an open formula without compound terms*;
- *(ii)* *$\boldsymbol{A} \models \Phi$ iff $\boldsymbol{A} \models \Phi'$ for each $\mathcal{L}$-algebra $\boldsymbol{A}$*;
- *(iii)* *Φ' can be computed from Φ by an algorithm working in time polynomial in $|\Phi|$.*

Proof. (i) Let Φ be an existential sentence. One may assume Φ is in prenex form (in particular, there is a polynomial-time algorithm which brings a sentence into an equivalent sentence in prenex form).

Let $T = \{t_1, \ldots, t_m\}$ be the collection of all terms in Φ_0. Let S be the collection of T-subterms, i.e., for each $s \in Fm_{\mathcal{L}}$, we have $s \in S$ iff $s \preceq t_i$ for some $i \in \{1, \ldots, m\}$. To each term $s \in S$, assign a variable x_s as follows:

– if s is a variable, let x_s be the variable s
– otherwise, let x_s be a new variable.

Denote S' the terms in S that are not variables.

Now let Φ_1 result from Φ_0 by replacing each atomic formula $t_1 = t_2$ ($t_1 \leq t_2$, $t_1 < t_2$) in Φ_0 with the atomic formula $x_{t_1} = x_{t_2}$ ($x_{t_1} \leq x_{t_2}$, $x_{t_1} < x_{t_2}$ respectively). Then all terms in Φ_1 are variables.

Moreover, let Φ_2 be

$$\bigwedge_{\substack{s \in S' \\ s \text{ is } f(s_1,\ldots,s_n)}} (x_s = f(x_{s_1}, \ldots, x_{s_n})).$$

Then Φ_2 is without compound terms. Finally, let Φ' be the existential closure of $\Phi_1 \wedge \Phi_2$.
(ii) It is elementary to check that Φ and Φ' are equivalent in any $\mathcal{L}$-algebra.
(iii) Identifying all (sub)terms in Φ, introducing new variables for subterms where necessary, building Φ_1 out of variables standing for terms in Φ, and listing all identities obtained for variables from the structure of subterms, is clearly polynomial in $|\Phi|$. □

In the following sections, we will not only be eliminating compound terms, but we will be using the particular translation of existential sentences given in the previous proof. For lack of a better term, we refer to the result of such a translation of an existential sentence Φ as the *existential normal form* of Φ.

3.4 Lower bounds

THEOREM 3.4.1. *Let $\mathcal{L}$ be a language and $\mathbb{K}$ a class of $\mathcal{L}$-algebras containing a non-trivial algebra. Then*

(i) $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$ *is* $\mathbf{coNP}$*-hard*;

(ii) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K})$ *is* $\mathbf{NP}$*-hard.*

Proof. Consider formulas of classical propositional logic in the language $\{\neg, \&, \vee\}$. Recall Proposition 2.4.2: satisfiability in $\{0,1\}_{\mathrm{B}}$ for CNF-formulas is $\mathbf{NP}$-complete, hence tautologousness in $\{0,1\}_{\mathrm{B}}$ for DNF-formulas is $\mathbf{coNP}$-complete. The two problems are denoted $\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_{\mathrm{B}})$ and $\mathrm{TAUT}^{\mathrm{DNF}}(\{0,1\}_{\mathrm{B}})$, respectively.

(i) We show $\mathrm{TAUT}^{\mathrm{DNF}}(\{0,1\}_{\mathrm{B}}) \preceq_{\mathbf{P}} \mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$. If $\varphi(x_1, \ldots, x_n)$ is a formula of classical propositional logic in DNF, define $\varphi^\star$ as

$$((x_1 \vee \neg x_1) \,\&\, \ldots \,\&\, (x_n \vee \neg x_n)) \to \varphi(x_1, \ldots, x_n).$$

We claim $\varphi \in \mathrm{TAUT}^{\mathrm{DNF}}(\{0,1\}_{\mathrm{B}})$ iff $\varphi^\star \in \mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$.

Assume first that φ is a classical tautology in DNF: a formula $D_1 \vee \ldots \vee D_m$, where each D_j, $1 \leq j \leq m$, is a conjunction of literals in variables among $x_1, \ldots, x_n$ and $m \in \mathsf{N}$. Without loss of generality we may assume each D_j contains each of its literals at most once.[15] We will show that $\varphi^\star$ is a theorem of $\mathrm{FL_{ew}}$ (hence a tautology of each $\mathrm{FL_{ew}}$-algebra, and a positive tautology of each nontrivial $\mathrm{FL_{ew}}$-algebra). Recall that

$$\psi \mathbin{\&} \bigvee_{j<k} \chi_j \leftrightarrow \bigvee_{j<k} (\psi \mathbin{\&} \chi_j) \tag{aux}$$

is a theorem of $\mathrm{FL_{ew}}$ for any choice of $\mathrm{FL_{ew}}$-formulas $\{\psi\} \cup \{\chi_j\}_{j<k}$. Assume $e \in \mathit{Val}^{\{x_1,\ldots,x_n\}}(\{0,1\}_\mathrm{B})$ is a (restriction of) Boolean evaluation; for each $i \in \{1, \ldots, n\}$, let x_i^e be the literal x_i if $e(x_i) = 1^{\{0,1\}_\mathrm{B}}$, and the literal $\neg x_i$ otherwise; let E^e be the formula $x_1^e \mathbin{\&} \ldots \mathbin{\&} x_n^e$. Then $e(E^e) = 1^{\{0,1\}_\mathrm{B}}$, and if $e' \in \mathit{Val}^{\{x_1,\ldots,x_n\}}(\{0,1\}_\mathrm{B})$, we have $e'(E^e) = 1^{\{0,1\}_\mathrm{B}}$ iff $e = e'$. Using (aux), the formula $(x_1 \vee \neg x_1) \mathbin{\&} \ldots \mathbin{\&} (x_n \vee \neg x_n)$ is $\mathrm{FL_{ew}}$-equivalent to the formula $\bigvee_{e \in \mathit{Val}^{\{x_1,\ldots,x_n\}}(\{0,1\}_\mathrm{B})} E^e$; we aim at showing

$$\vdash_{\mathrm{FL_{ew}}} \big(\bigvee_{e \in \mathit{Val}^{\{p_1,\ldots,p_n\}}(\{0,1\}_\mathrm{B})} E^e \big) \to \bigvee_{j=1}^{m} D_j \ . \tag{1}$$

As φ is a classical tautology, for each Boolean evaluation e we reason as follows. There is a $j_e \in \{1, \ldots, m\}$ such that $e(D_{j_e}) = 1^{\{0,1\}_\mathrm{B}}$; this implies that the literals in D_{j_e} are among the literals in E^e, and weakening gives the sequent $E^e \Rightarrow D_{j_e}$ in $\mathrm{FL_{ew}}$. Now, for each Boolean evaluation e, if the last sequent is provable, then (introducing $\vee$ to the right) so is $E^e \Rightarrow \varphi$. Finally (introducing $\vee$ to the left repeatedly), we get $\vdash_{\mathrm{FL_{ew}}} \bigvee_{e \in \mathit{Val}^{\{x_1,\ldots,x_n\}}(\{0,1\}_\mathrm{B})} E^e \Rightarrow \bigvee_{j=1}^{m} D_j$, whence (1) follows.

On the other hand, if φ is not a classical tautology, there is a Boolean evaluation e such that $e(\varphi) = 0^{\{0,1\}_\mathrm{B}}$. For each $\boldsymbol{A} \in \mathbb{K}$, define $e_{\boldsymbol{A}}$ in such a way that $e_{\boldsymbol{A}}(x_i) = e(x_i)$ for $1 \leq i \leq n$. Then $e_{\boldsymbol{A}}((x_1 \vee \neg x_1) \mathbin{\&} \ldots \mathbin{\&} (x_n \vee \neg x_n)) = 1^{\boldsymbol{A}}$ (because $e_{\boldsymbol{A}}$ only takes classical values on x_i's), so $e_{\boldsymbol{A}}(\varphi^\star) = 0^{\boldsymbol{A}}$. Hence, $\varphi^\star \notin \mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$.

(ii) We show $\mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_\mathrm{B}) \preceq_{\mathbf{P}} \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K})$. If $\varphi(x_1, \ldots, x_n)$ is a formula of classical propositional logic in CNF, define φ° as

$$(x_1 \vee \neg x_1) \mathbin{\&} \ldots \mathbin{\&} (x_n \vee \neg x_n) \mathbin{\&} \varphi(x_1, \ldots, x_n).$$

We claim $\varphi \in \mathrm{SAT}^{\mathrm{CNF}}(\{0,1\}_\mathrm{B})$ iff $\varphi^\circ \in \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K})$.

If φ is classically satisfiable, there is a Boolean evaluation e such that $e(\varphi) = 1^{\{0,1\}_\mathrm{B}}$. For each $\boldsymbol{A} \in \mathbb{K}$, define $e_{\boldsymbol{A}}$ in such a way that $e_{\boldsymbol{A}}(x_i) = e(x_i)$ for $1 \leq i \leq n$. Then $e_{\boldsymbol{A}}((x_1 \vee \neg x_1) \mathbin{\&} \ldots \mathbin{\&} (x_n \vee \neg x_n)) = 1^{\boldsymbol{A}}$, and $e_{\boldsymbol{A}}(\varphi) = 1^{\boldsymbol{A}}$ by assumption. We get $\varphi^\circ \in \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K})$.

On the other hand, assume φ is a formula in CNF which is not classically satisfiable. Using (aux), the formula φ° is $\mathrm{FL_{ew}}$-equivalent to

$$\bigvee_{e \in \mathit{Val}^{\{x_1,\ldots,x_n\}}(\{0,1\}_\mathrm{B})} (E^e \mathbin{\&} \varphi). \tag{2}$$

[15] More precisely, the problem $\mathrm{TAUT}^{\mathrm{DNF}}(\{0,1\}_\mathrm{B})$ is polynomially equivalent to its modification where no conjunction may contain identical literals.

Now φ is $C_1 \,\&\, \ldots \,\&\, C_m$, where each C_j, $1 \leq j \leq m$ is a disjunction of literals in variables among $x_1, \ldots, x_n$ and $m \in \mathsf{N}$. For each (restriction of) Boolean evaluation e, it is sufficient to show that $E^e \,\&\, \varphi$ in (2) is not (positively) satisfiable in any nontrivial $\boldsymbol{A} \in \mathbb{K}$ (the trivial algebra is omitted from consideration for $\mathrm{SAT}(\mathbb{K})$, and no formula is positively satisfiable in a trivial algebra). For a given e, we may reason as follows. If φ is unsatisfiable, there is a $j_e \in \{1, \ldots, m\}$ such that $e(C_{j_e}) = 0^{\{0,1\}_\mathrm{B}}$; that is, $e(l) = 0^{\{0,1\}_\mathrm{B}}$ for each literal l in C_{j_e}. The formula $E^e \,\&\, \varphi$ is FL_ew-equivalent to

$$E^e \,\&\, C_{j_e} \,\&\, (C_1 \,\&\, \ldots \,\&\, C_{j_e-1} \,\&\, C_{j_e+1} \,\&\, \ldots \,\&\, C_m). \tag{3}$$

If C_{j_e} is $l_1 \vee \ldots \vee l_q$ for some $q \geq 1$, then, using (aux) again, $E^e \,\&\, C_{j_e}$ is FL_ew-equivalent to $\bigvee_{1 \leq k \leq q}(E^e \,\&\, l_k)$. For each $1 \leq k \leq q$, if l_k is an x_i for some i, then $e(x_i) = e(l_k) = 0^{\{0,1\}_\mathrm{B}}$, hence x_i^e is $\neg x_i$, and the latter occurs in E^e; so the formula $E^e \,\&\, l_k$ is a conjunction of literals where both x_i and $\neg x_i$ occur. A dual argument applies when l_k is a $\neg x_i$ for some i. Recall $\varphi \,\&\, \neg\varphi \leftrightarrow \overline{0}$ is a theorem of FL_ew, in particular, $\varphi \,\&\, \neg\varphi$ is unsatisfiable in a nontrivial FL_ew-algebra. So $\bigvee_{1 \leq k \leq q}(E^e \,\&\, l_k)$ is unsatisfiable in a nontrivial FL_ew-algebra and hence, so is (3). Hence, φ° is not (positively) satisfiable in $\mathbb{K}$. □

Part (i) of the above theorem was proved in a stronger way in [29]: theoremhood in a consistent substructural logic is **coNP**-hard, if moreover the logic has the disjunction property, then it is **PSPACE**-hard. We will not be able to use this stronger result as semilinear logics do not have the disjunction property.[16]

4 Łukasiewicz logic

Łukasiewicz logic merits particular attention when studying fuzzy logic, and computational complexity of its propositional fragment is no exception. **NP**-completeness of satisfiability in the standard MV-algebra $[0,1]_Ł$ was proved in 1987 by D. Mundici; many other complexity results in propositional fuzzy logic refer to this result.

Łukasiewicz logic Ł can be viewed as the axiomatic extension of BL with axiom $\neg\neg\varphi \to \varphi$. The equivalent algebraic semantics of Ł is the variety $\mathbb{MV}$ of MV-algebras. Propositional Łukasiewicz logic is strongly complete w.r.t. MV-chains, and finitely strongly complete w.r.t. the standard algebra $[0,1]_Ł$ given by the Łukasiewicz t-norm; these results are due to C.C. Chang. Therefore, the section starts with investigating complexity of decision problems in the standard MV-algebra $[0,1]_Ł$; this happens in Subsection 4.1. Next, relying on Y. Komori's characterization of subvarieties of $\mathbb{MV}$, Subsection 4.2 addresses complexity of decision problems in these subvarieties.

Within this section we work with the language $\{\&, \to, \overline{0}\}$.

4.1 The standard MV-algebra

Our aim is to investigate complexity of the SAT, TAUT and CONS problems in the standard MV-algebra $[0,1]_Ł$, given by the Łukasiewicz t-norm $*_Ł$ and its residuum $\to_Ł$ on $[0,1]$. Thanks to finite strong standard completeness of Ł, the obtained results will apply also to theoremhood and provability in Ł.

[16] However, the logic FL_ew, whose language we borrow and which acts as a basis for many of our considerations, does have the disjunction property and thus, theoremhood in FL_ew is **PSPACE**-hard.

LEMMA 4.1.1. *For each $x, y, z \in [0,1]$ the following hold in the reals:*

(i) $x *_{Ł} y = z$ *iff* $((x+y-1 \geq 0) \wedge (z = x+y-1)) \vee ((x+y-1 < 0) \wedge (z = 0))$;

(ii) $x \rightarrow_{Ł} y = z$ *iff* $((x \leq y) \wedge (z = 1)) \vee ((x > y) \wedge (z = 1 - x + y))$.

We make a statement about complexity of the universal fragment of $\mathrm{Th}([0,1]_{Ł})$, and obtain result for SAT, TAUT, and CONS in $[0,1]_{Ł}$ as a corollary.

THEOREM 4.1.2. $\mathrm{Th}_{\forall}([0,1]_{Ł})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from Theorem 3.4.1, in view of Lemma 3.1.1. The latter also says it suffices to address $\mathrm{Th}_{\exists}([0,1]_{Ł})$, a problem polynomially equivalent to the complement of $\mathrm{Th}_{\forall}([0,1]_{Ł})$. In the rest of the proof, we show containment of $\mathrm{Th}_{\exists}([0,1]_{Ł})$ in $\mathbf{NP}$. We present a nondeterministic algorithm, working in time polynomial in the input size, which uses a subroutine deciding the INEQ problem. (See also discussion below.)

ALGORITHM EX-L `// accepts` $\mathrm{Th}_{\exists}([0,1]_{Ł})$

`input:` Φ `// existential sentence in the language of BL`

`begin`

`normalForm()` Using Lemma 3.3.5, transform Φ into a (logically equivalent) sentence in existential normal form, $\exists x_1 \dots \exists x_n(\Phi_1 \wedge \Phi_2)$. Remove the quantifier prefix, and consider the formula $\Phi_1 \wedge \Phi_2$ as a Boolean combination of equations and inequalities in $[0,1]_{Ł}$.[17]

`guessOrder()` Guess a linear ordering $\leq_0$ of the set $V = \{0, 1, x_1, \dots, x_n\}$, such that $0 \leq_0 x_i \leq_0 1$ for $1 \leq i \leq n$, and $0 <_0 1$; henceforth we take this ordering as fixed, and exploit this piece of information (without assigning exact values to the variables). Let Ψ denote the conjunction of conditions expressing the ordering $\leq_0$, i.e., Ψ is

$$\bigwedge_{\substack{x,y \in V \\ x =_0 y}} (x = y) \wedge \bigwedge_{\substack{x,y \in V \\ x <_0 y}} (x < y)$$

`checkOrder()` Check that Φ_1 is consistent with $\leq_0$. Recall that Φ_1 is a Boolean combination of equations and inequalities between pairs of variables in V. Since $\leq_0$ gives a full information about ordering of all variables in Φ_1, it is easy to perform the check; first assess the validity of the atomic conditions in Φ_1 against $\leq_0$, then compute the validity of Φ_1, using Boolean operations.

`checkInR()` Use Lemma 4.1.1 to replace each equation in Φ_2 of type $x *_{Ł} y = z$ or $x \rightarrow_{Ł} y = z$ with an equivalent condition in the language of linear inequalities in R. Other equations in Φ_2 (i.e., those of the form $x = c$ for c a constant) remain intact. Let Φ_2' denote the conjunction of thus obtained equations. Pass $\Phi_2' \wedge \Psi$ to a subroutine deciding the INEQ problem.[18]

`end`

[17] The sentence $\exists x_1 \dots \exists x_n(\Phi_1 \wedge \Phi_2)$ is valid in $[0,1]_{Ł}$ iff there is an n-tuple $a_1, \dots, a_n \in [0,1]$ that satisfies the Boolean combination of equations and inequalities given by Φ_1 and Φ_2 in $[0,1]_{Ł}$.

[18] A bit more work is needed to rewrite the atomic conditions in $\Phi_2' \wedge \Psi$ as basic inequalities; we leave this as an exercise to the reader.

We claim the algorithm operates in time polynomial in $|\Phi|$, relying in case of the first step on Lemma 3.3.5, for the next two steps the claim is obvious. If the step `checkInR()` is reached, the subroutine for INEQ is called, which (as explained in Subsection 2.5) operates in polynomial time.

Correctness of the algorithm is argued as follows. By Lemma 3.3.5, the formula $\exists x_1 \dots \exists x_n(\Phi_1 \wedge \Phi_2)$, the existential normal form of Φ, is logically equivalent to Φ itself. Assume the existential normal form is true in $[0,1]_{Ł}$. Then there are values $a_1, \dots, a_n \in [0,1]$ such that both Φ_1 and Φ_2 are satisfied under any evaluation e such that $e(x_i) = a_i$. Then one of the computations that guess such an ordering of the x_i's that mirrors the actual ordering of the a_i's within $[0,1]$ will be an accepting computation:[19] Φ_1 will be consistent with the guessed ordering, and by Lemma 4.1.1 and a correctness argument for the algorithm for INEQ (knowing that the conditions in Φ_2' and in Ψ are satisfied by $a_1, \dots, a_n$), we may conclude that the computation will terminate in an accepting state. Conversely, it is clear that any solution found by the algorithm yields a satisfying evaluation of $\Phi_1 \wedge \Phi_2$. □

Discussion. One can simplify the above algorithm to actually show $\mathrm{Th}_\exists([0,1]_{Ł}) \preceq_{\mathbf{P}}$ INEQ (also obtaining $\mathbf{NP}$-containment for $\mathrm{Th}_\exists([0,1]_{Ł})$). This is done as follows: take the existential sentence Φ; transform it into an existential normal form; remove quantifiers, add boundary conditions; replace equations involving $*_{Ł}$, $\rightarrow_{Ł}$ with their equivalents in the language of linear inequalities; pass to INEQ algorithm. We have preferred the version given in the proof because working with an explicit ordering enables us to incorporate, later on, some additional connectives, such as $\triangle$, whose semantics is order-determined.

On the other hand, one could do without the subroutine for the INEQ problem and use a subroutine for the LP problem instead, involving more nondeterminism. We refrain from going into detail, but refer the reader to the proof of Theorem 4.2.5, where this modified version of the algorithm is used for Komori algebras, relying on a subroutine deciding the ILP problem. The algorithm for Komori algebras relies on the discrete order of integers; here, one would have to feed the formula Ψ—with its strict inequalities—to the LP algorithm, which can be done taking a new variable ϵ and replacing each strict inequality $x <_0 y$ in Ψ with $x + \epsilon \leq y$, finally adding $\epsilon > 0$; this yields a formula Ψ' conforming to the criteria on a LP problem.

COROLLARY 4.1.3.

(i) $\mathrm{SAT}_{(\mathrm{pos})}([0,1]_{Ł})$ *is* $\mathbf{NP}$*-complete.*

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}([0,1]_{Ł})$ *and* $\mathrm{CONS}([0,1]_{Ł})$ *are* $\mathbf{coNP}$*-complete.*

(iii) $\mathrm{THM}(Ł)$ *and* $\mathrm{CONS}(Ł)$ *are* $\mathbf{coNP}$*-complete.*

Proof. (i) Hardness stated in Theorem 3.4.1; containment follows from $\mathbf{NP}$-containment of $\mathrm{Th}_\exists([0,1]_{Ł})$ using Lemma 3.1.1.
(ii) Analogous to (i), with respect to $\mathrm{Th}_\forall([0,1]_{Ł})$.
(iii) Using (finite strong) standard completeness result for Ł. □

[19] One cannot say '*the* computation that guesses the ordering' because another nondeterministic step is still ahead in the INEQ subroutine.

It still remains to show that the decision problems are nontrivial in the sense of being distinct from each other and from the classical case.

LEMMA 4.1.4.

(i) $\mathrm{SAT}(\{0,1\}_{\mathrm{B}}) \subsetneq \mathrm{SAT}([0,1]_{\text{Ł}}) \subsetneq \mathrm{SAT}_{\mathrm{pos}}([0,1]_{\text{Ł}})$;

(ii) $\mathrm{TAUT}([0,1]_{\text{Ł}}) \subsetneq \mathrm{TAUT}_{\mathrm{pos}}([0,1]_{\text{Ł}} \subsetneq \mathrm{TAUT}(\{0,1\}_{\mathrm{B}})$.

Proof. For a variable p:

$$\begin{aligned} p \wedge \neg p &\in \mathrm{SAT}_{\mathrm{pos}}([0,1]_{\text{Ł}}) \setminus \mathrm{SAT}([0,1]_{\text{Ł}}) \\ p \leftrightarrow \neg p &\in \mathrm{SAT}([0,1]_{\text{Ł}}) \setminus \mathrm{SAT}(\{0,1\}_{\mathrm{B}}) \\ p \vee \neg p &\in \mathrm{TAUT}_{\mathrm{pos}}([0,1]_{\text{Ł}}) \setminus \mathrm{TAUT}([0,1]_{\text{Ł}}) \\ (p \vee \neg p)\ \&\ (p \vee \neg p) &\in \mathrm{TAUT}(\{0,1\}_{\mathrm{B}}) \setminus \mathrm{TAUT}_{\mathrm{pos}}([0,1]_{\text{Ł}}). \end{aligned}$$

□

4.2 Axiomatic extensions of Łukasiewicz logic

A result of Y. Komori characterizing subvarieties of $\mathbb{MV}$ makes it possible to extend the complexity results obtained for Łukasiewicz logic also to its axiomatic extensions. For L a consistent axiomatic extension of Łukasiewicz logic, let $\mathbb{MV}^{\mathrm{L}}$ be the subvariety of $\mathbb{MV}$ forming its equivalent algebraic semantics.

We start by introducing notation for finite MV-chains and Komori chains. Denote $\mathsf{N}_0 = \mathsf{N} \setminus \{0\}$ and $\mathsf{N}_1 = \mathsf{N} \setminus \{0,1\}$.

DEFINITION 4.2.1. *For $n \in \mathsf{N}_0$, denote:*

(i) Ł_{n+1} *the subalgebra of* $[0,1]_{\text{Ł}}$ *with the domain* $\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$;

(ii) $\boldsymbol{K}_{n+1}$ *the algebra*

$$\langle \{\langle i,a\rangle \in \mathsf{Z}\times\mathsf{Z} \mid \langle 0,0\rangle \leq_{\mathrm{lex}} \langle i,a\rangle \leq_{\mathrm{lex}} \langle n,0\rangle\}, *_{\boldsymbol{K}_{n+1}}, \rightarrow_{\boldsymbol{K}_{n+1}}, \langle 0,0\rangle\rangle,$$

where $\leq_{\mathrm{lex}}$ *is the lexicographic order on* $\mathsf{Z}\times\mathsf{Z}$ *and* $*_{\boldsymbol{K}_{n+1}}$, $\rightarrow_{\boldsymbol{K}_{n+1}}$ *are given by* $\langle i,x\rangle *_{\boldsymbol{K}_{n+1}} \langle j,y\rangle = \max_{\mathrm{lex}}(\langle 0,0\rangle, \langle i+j-n, x+y\rangle)$ *and* $\langle i,x\rangle \rightarrow_{\boldsymbol{K}_{n+1}} \langle j,y\rangle = \min_{\mathrm{lex}}(\langle n,0\rangle, \langle n-i+j, y-x\rangle)$.

Observe that for each $n \in \mathsf{N}_0$, the algebra Ł_{n+1} is isomorphic to a subalgebra of $\boldsymbol{K}_{n+1}$ (obtained by considering only elements $\langle x,0\rangle$).

DEFINITION 4.2.2. *For $A, B \subseteq \mathsf{N}_1$, denote*

(i) $\mathbb{K}_A = \{\boldsymbol{K}_a \mid a \in A\}$;

(ii) $\mathbb{L}_B = \{\boldsymbol{L}_b \mid b \in B\}$.

PROPOSITION 4.2.3 ([10, 19, 31]). *For* L *a consistent axiomatic extension of* Ł:

(i) $\mathbb{MV}^{\mathrm{L}} = \mathbf{V}(\mathbb{K}_A \cup \mathbb{L}_B)$ *for some* $A, B \subseteq \mathsf{N}_1$;

(ii) if (i) is true for L, *A, and B, then also* $\mathbb{MV}^{\mathrm{L}} = \mathbf{Q}(\mathbb{K}_A \cup \mathbb{L}_B)$;

(iii) $\mathbf{V}(\mathbb{K}_A \cup \mathbb{L}_B) = \mathbb{MV}$ *iff either of A, B is infinite*;

(iv) L *is finitely axiomatizable.*

These results are crucial for us: for L a consistent axiomatic extension of Ł, if $\mathbb{MV}^{\mathrm{L}}$ is a proper subvariety of $\mathbb{MV}$, it is generated by a pair of finite lists of algebras of known structure; it is generated by these algebras as a quasivariety; and, using strong completeness theorem for axiomatic extensions of Ł (or, of BL), results on complexity of TAUT and CONS for $\mathbb{MV}^{\mathrm{L}}$ apply also to THM and CONS in L. The fact that, for given $\mathbb{MV}^{\mathrm{L}}$, the pair of lists need not be unique is of no material importance here.

Now we are interested in complexity of SAT, TAUT, and CONS problems in the algebras defined above, $\mathbf{Ł}_{n+1}$ and $\boldsymbol{K}_{n+1}$. We work with the algebra $\boldsymbol{K}_{n+1}$ for a fixed $n \in \mathsf{N}_0$. For $\mathbf{Ł}_{n+1}$, the upper bound is obvious as the algebra is finite (one can also argue that $\mathbf{Ł}_{n+1}$ is a subalgebra of $\boldsymbol{K}_{n+1}$, as above).

LEMMA 4.2.4. *Let* $n \in \mathsf{N}_1$. *Let* $\langle i_1, x_1\rangle, \langle i_2, x_2\rangle, \langle i_3, x_3\rangle \in [\langle 0,0\rangle, \langle n,0\rangle]_{\mathrm{lex}}$ *in* $\mathsf{Z} \times \mathsf{Z}$. *Then the following holds in the integers:*

(i) $\langle i_1, x_1\rangle *_{\boldsymbol{K}_{n+1}} \langle i_2, x_2\rangle = \langle i_3, x_3\rangle$ *iff either*
$i_1 + i_2 - n \leq -1$ *and* $i_3 = 0$ *and* $x_3 = 0$*, or*
$i_1 + i_2 - n = 0$ *and* $x_1 + x_2 \leq -1$ *and* $i_3 = 0$ *and* $x_3 = 0$*, or*
$i_1 + i_2 - n = 0$ *and* $x_1 + x_2 \geq 0$ *and* $i_3 = 0$ *and* $x_3 = x_1 + x_2$*, or*
$i_1 + i_2 - n \geq 1$ *and* $i_3 = i_1 + i_2 - n$ *and* $x_3 = x_1 + x_2$.

(ii) $\langle i_1, x_1\rangle \to_{\boldsymbol{K}_{n+1}} \langle i_2, x_2\rangle = \langle i_3, x_3\rangle$ *iff either*
$i_1 < i_2$ *and* $i_3 = n$ *and* $x_3 = 0$*, or*
$i_1 = i_2$ *and* $x_1 \leq x_2$ *and* $i_3 = n$ *and* $x_3 = 0$*, or*
$i_1 = i_2$ *and* $x_1 \geq x_2 + 1$ *and* $i_3 = n$ *and* $x_3 = x_2 - x_1$*, or*
$i_1 \geq i_2 + 1$ *and* $i_3 = n - i_1 + i_2$ *and* $x_3 = x_2 - x_1$.

THEOREM 4.2.5. $\mathrm{Th}_\forall(\boldsymbol{K}_{n+1})$ *is* $\mathbf{coNP}$*-complete for each* $n \in \mathsf{N}_0$.

Proof. Hardness follows from Theorem 3.4.1. We argue $\mathbf{NP}$-containment for existential sentences valid in $\boldsymbol{K}_{n+1}$ for an arbitrary but fixed $n \in \mathsf{N}_0$.

ALGORITHM EX-K `// accepts` $\mathrm{Th}_\exists(\boldsymbol{K}_{n+1})$

`input:` Φ `// existential sentence in the language of BL`

`begin`

`normalForm()` Using Lemma 3.3.5, transform Φ into an existential normal form $\exists x_1 \ldots \exists x_n(\Phi_1 \wedge \Phi_2)$. Remove quantifiers. In the open formula $\Phi_1 \wedge \Phi_2$, for $j = 1, \ldots, n$ replace each occurrence of the variable x_j with a pair $\langle i_j, z_j\rangle$ where for $j = 1, \ldots, n$, the pair i_j, z_j are new variables. Consider $\Phi_1 \wedge \Phi_2$ as a Boolean combination of equations and inequalities in $\boldsymbol{K}_{n+1}$.

`guessOrder()` Guess a linear ordering $\leq_0$ of the set $\{\langle i_j, z_j\rangle\}_{j=1}^n \cup \{\langle 0,0\rangle, \langle n,0\rangle\}$, in such a way that $\langle 0,0\rangle \leq_0 \langle i_j, z_j\rangle \leq_0 \langle n,0\rangle$ for $1 \leq j \leq n$, and $\langle 0,0\rangle <_0 \langle n,0\rangle$. Denote Ψ the set of conditions expressing the ordering $\leq_0$.

`checkOrder()` As in the proof of Theorem 4.1.2.

`checkInZ()` Rewrite the conditions in Ψ and in Φ_2 into the language of the o-group of Z and check their solvability, in the following (nondeterministic) manner. Define a new empty system $\mathcal{S}$. Processing the conditions in Ψ one by one, in case of:

– $\langle i_j, z_j\rangle = \langle i_k, z_k\rangle$ in Ψ, put $i_j = i_k$, $x_j = x_k$ into $\mathcal{S}$;
– $\langle i_j, z_j\rangle < \langle i_k, x_k\rangle$ in Ψ, put either $i_j = i_k$ and $z_j + 1 \leq z_k$, or $i_j + 1 \leq i_k$, into $\mathcal{S}$.

Note that this process imposes boundary appropriate conditions on the pairs of variables. Then process the conditions in Φ_2 one by one, for each equation $x *_{\boldsymbol{K}_{n+1}} y = z$, choose exactly one of the four mutually exclusive cases from Lemma 4.2.4 (i), and add it into $\mathcal{S}$; analogously for each atomic formula of type $x \to_{\boldsymbol{K}_{n+1}} y = z$, again using a case from among its equivalents posed by Lemma 4.2.4 (ii). Then process each atomic formula of type $\langle i_j, x_j\rangle = \langle 0, 0\rangle$ as above. Finally, pass $\mathcal{S}$ to an algorithm for the ILP problem.

`end` □

THEOREM 4.2.6. *Let* L *be a consistent axiomatic extension of Łukasiewicz logic. Then* $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{MV}^{\mathrm{L}})$ *is* $\mathbf{NP}$*-complete, whereas* $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{MV}^{\mathrm{L}})$ *and* $\mathrm{CONS}(\mathbb{MV}^{\mathrm{L}})$ *are* $\mathbf{coNP}$*-complete.*

Proof. Hardness by Theorem 3.4.1. Consider L, $\mathbb{MV}^{\mathrm{L}}$ as stated. If L is Ł, then results in the previous subsection apply. Otherwise, using Theorem 4.2.3 (i), we have $\mathbb{MV}^{\mathrm{L}} = \mathbf{V}(\mathbb{K}_A \cup \mathbb{L}_B)$ for some finite $A, B \subseteq \mathsf{N}_1$. Fix such a pair A and B. Recall $\mathbf{V}(\mathbb{K}_A \cup \mathbb{L}_B) = \mathbf{Q}(\mathbb{K}_A \cup \mathbb{L}_B)$ (cf. Theorem 4.2.3 (ii)). Then by Theorem 3.1.6: $\mathrm{SAT}_{(\mathrm{pos})}(\mathbf{V}(\mathbb{K}_A \cup \mathbb{L}_B)) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbf{Q}(\mathbb{K}_A \cup \mathbb{L}_B)) = \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}_A \cup \mathbb{L}_B) = \bigcup_{a\in A} \mathrm{SAT}_{(\mathrm{pos})}(\boldsymbol{K}_a) \cup \bigcup_{b\in B} \mathrm{SAT}_{(\mathrm{pos})}(\boldsymbol{L}_b)$ (last equality by definition of $\mathrm{SAT}_{(\mathrm{pos})}$). Since $\mathbf{NP}$ is closed under finite unions, we may conclude, on the basis of A, B being finite, Theorem 4.2.5 and the remark about finite MV-chains, that $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{MV}^{\mathrm{L}})$ is in $\mathbf{NP}$. Hence, $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{MV}^{\mathrm{L}})$ is in $\mathbf{coNP}$ by Corollary 3.2.2. By an analogous argument we get that $\overline{\mathrm{CONS}}(\mathbb{MV}^{\mathrm{L}})$ is in $\mathbf{NP}$, and hence $\mathrm{CONS}(\mathbb{MV}^{\mathrm{L}})$ is in $\mathbf{coNP}$. □

Using strong completeness for extensions $\mathrm{L} \supseteq$ Ł with respect to the corresponding varieties $\mathbb{MV}^{\mathrm{L}}$, we may conclude

COROLLARY 4.2.7. *Let* L *be a consistent axiomatic extension of Łukasiewicz logic. Then* $\mathrm{THM}(\mathrm{L})$ *and* $\mathrm{CONS}(\mathrm{L})$ *are* $\mathbf{coNP}$*-complete.*

5 Logics of standard BL-algebras

This section studies decision problems in propositional BL and extensions given by continuous t-norms (distinct from Łukasiewicz). We start with Gödel and product logics, proceed to BL and SBL, and then we discuss logics given by single standard BL-algebras (and thus, by single continuous t-norms; hence the section title). Within this section, we work with the language $\{\&, \to, \overline{0}\}$.

5.1 Gödel and product logics

Gödel logic G can be obtained as an axiomatic extension of BL with the axiom $\varphi \to \varphi \,\&\, \varphi$. It is discussed in detail in Chapter VII. The logic is strongly complete w.r.t. its standard algebra $[0,1]_{\mathrm{G}}$, given by the continuous t-norm $x *_{\mathrm{G}} y = \min\{x, y\}$ and its residuum $x \to_{\mathrm{G}} y = y$ for $x > y$.

Product logic extends BL with the axiom $(\varphi \to \chi) \vee ((\varphi \to (\varphi \mathbin{\&} \psi)) \to \psi)$. Product logic is finitely strongly complete w.r.t. its standard algebra $[0,1]_\Pi$, given by the continuous t-norm $x *_\Pi y = xy$; the residuum is $x \to_\Pi y = \frac{y}{x}$ for $x > y$.

Both logics extend SMTL (and SBL), therefore Theorem 3.2.4 applies for the SAT, $\mathrm{SAT}_{\mathrm{pos}}$ and $\mathrm{TAUT}_{\mathrm{pos}}$ operators on the respective standard algebra; it is therefore sufficient to address the set of tautologies and the consequence relation. Moreover, recall that in Gödel logic, we have the classical deduction theorem, i.e., for $\psi_1, \ldots, \psi_n, \varphi$ formulas, we have $\{\psi_1, \ldots, \psi_n\} \vdash_{\mathrm{G}} \varphi$ iff $\vdash_{\mathrm{G}} \psi_1 \mathbin{\&} \ldots \mathbin{\&} \psi_n \to \varphi$. In view of Lemma 3.1.1, it is therefore sufficient to investigate theoremhood for Gödel logic.

THEOREM 5.1.1. *Let $\boldsymbol{L}$ be a Gödel chain. Then* $\mathrm{Th}_\forall(\boldsymbol{L})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from Theorem 3.4.1. We show $\mathbf{NP}$-containment of $\mathrm{Th}_\exists(\boldsymbol{L})$. First, in a Gödel chain $\boldsymbol{L} = \langle *, \to, \wedge, \vee, 0, 1\rangle$, we have $x * y = x \wedge y = \min\{x, y\}$ and $x \to y = 1$ iff $x \leq y$, otherwise $x \to y = y$. Hence for any term $t(x_1, \ldots, x_n)$ and for any evaluation e in $\boldsymbol{L}$, the value $e(t')$ for any subterm $t' \preceq t$ will be among $V = \{0^{\boldsymbol{L}}, e(x_1), \ldots, e(x_n), 1^{\boldsymbol{L}}\}$, and moreover, operations are order-determined, i.e., the value $e(t')$ is fully determined by the ordering of V.

So, determining the validity of an existential sentence $\exists x_1 \ldots \exists x_n \Phi(x_1, \ldots, x_n)$, one can replace the existential quantification over all n-tuples of values in $[0,1]$ by an existential quantification over all such orderings of variables occurring in Φ that are possible in $\boldsymbol{L}$, with respect to its bottom and top elements.

We describe a nondeterministic ALGORITHM EX-G which accepts existential sentences valid in $\boldsymbol{L}$. A sentence $\exists x_1 \ldots \exists x_n \Phi(x_1, \ldots, x_n)$ is given, where Φ is a Boolean combination of identities. Guess a linear ordering $\leq_0$ of the set $V = \{0, x_1, \ldots, x_n, 1\}$, such that $0 \leq_0 x_i \leq_0 1$ for $1 \leq i \leq n$, and in such a way that the length of any strictly increasing $\leq_0$-chain does not exceed the cardinality of $\boldsymbol{L}$; this information is polynomial in the input size. Then compute the Boolean value of each identity in Φ: on the basis of $\leq_0$, evaluate all terms, then determine whether they are $=_0$-equal. Then compute the Boolean value of Φ, accept iff this value is 1.

It is easy to see that the algorithm runs in time polynomial in the input size. It is equally easy to see that it accepts just the set $\mathrm{Th}_\exists(\boldsymbol{L})$. □

COROLLARY 5.1.2.

(i) $\mathrm{TAUT}([0,1]_{\mathrm{G}})$ *is* $\mathbf{coNP}$*-complete.*

(ii) $\mathrm{THM}(\mathrm{G})$ *is* $\mathbf{coNP}$*-complete.*

Now we address product logic, via its standard algebra. We start with a lemma.

LEMMA 5.1.3. *For each $c \in (0,1)$, the standard* MV*-algebra* $[0,1]_{\text{Ł}}$ *is isomorphic to the* cut product algebra $\langle [c,1], *_c, \to_c, c, 1\rangle$ *where*

$$x *_c y = \max\{c, x *_\Pi y\} \qquad\qquad x \to_c y = x \to_\Pi y$$

The element c is called the cut.

Proof. For a fixed c, the isomorphism is given by $f(x) = c^{1-x}$, its inverse by $f^{-1}(y) = 1 - \log_c(y)$. □

THEOREM 5.1.4. $\mathrm{Th}_\forall([0,1]_\Pi)$ *is* $\mathbf{coNP}$*-complete.*

Proof. For hardness, see Theorem 3.4.1. We show $\mathbf{NP}$-containment of $\mathrm{Th}_\exists([0,1]_\Pi)$ by presenting a nondeterministic algorithm, operating on existential sentences, accepting $\mathrm{Th}_\exists([0,1]_\Pi)$, and working in time polynomial in the input formula size. The algorithm uses a subroutine deciding the existential theory of $[0,1]_Ł$.

ALGORITHM EX-PRODUCT `// accepts` $\mathrm{Th}_\exists([0,1]_\Pi)$

`input:` Φ `// existential sentence in the language of BL`

`begin`

`normalForm()` As in the proof of Theorem 4.1.2.

`guessOrder()` As in the proof of Theorem 4.1.2.

`checkOrder()` As in the proof of Theorem 4.1.2.

`eliminateZero()` Partition the equalities in Φ_2 into two classes: let Φ_2^0 contain those equalities in Φ_2 which contain at least one $=_0$-equal variable, let $\Phi_2^{>0}$ contain the remaining equalities. Then check that all equalities in Φ_2^0 are consistent with $\leq_0$, as follows. Processing them one by one, in case of:

- an equality $x = 0$, check $x =_0 0$;
- an equality $x *_\Pi y = z$, if $x =_0 0$, check $z =_0 0$, analogously for y; if $z =_0 0$, check $x =_0 0$ or $y =_0 0$;
- an equality $x \to_\Pi y = z$, if $x =_0 0$, check $z =_0 1$; if $y =_0 0$, check that either $z =_0 0$ and $x \neq_0 0$, or $x =_0 0$ and $z =_0 1$; if $z =_0 0$, check $x \neq_0 0$ and $y =_0 0$.

In the checking process, we have made sure that all atomic formulas in Φ_2^0 are valid under $\leq_0$. Finally, omit from Ψ the variables that are $=_0$-equal to 0, obtaining a $\Psi^{>0}$.

`positiveL()` Test whether the conditions in $\Phi_2^{>0}$ and the conditions in $\Psi^{>0}$ are satisfiable by positive values in $[0,1]_\Pi$; by Lemma 5.1.3, this is iff they are satisfiable by positive values in $[0,1]_Ł$. Use the subroutine `checkInR()` in the proof of Theorem 4.1.2 to test the positive satisfiability of $\Phi_2^{>0} \wedge \Psi^{>0}$ in $[0,1]_Ł$. It follows from that proof that the subroutine works in polynomial time.

`end` □

COROLLARY 5.1.5.

(i) $\mathrm{TAUT}([0,1]_\Pi)$ *and* $\mathrm{CONS}([0,1]_\Pi)$ *are* $\mathbf{coNP}$*-complete.*

(ii) $\mathrm{THM}(\Pi)$ *and* $\mathrm{CONS}(\Pi)$ *are* $\mathbf{coNP}$*-complete.*

5.2 BL and SBL

In this subsection we address theorems and provability in the logic BL and in its axiomatic extension SBL with the axiom $\neg(\varphi \wedge \neg\varphi)$. We denote $\mathbb{BL}^{\mathrm{st}}$ and $\mathbb{SBL}^{\mathrm{st}}$ the classes of all standard BL-algebras and all standard SBL-algebras, respectively. We work with fragments of the respective algebraic theories, relying on finite strong standard completeness results: propositional BL (SBL) is finitely strongly complete with respect to the class $\mathbb{BL}^{\mathrm{st}}$ ($\mathbb{SBL}^{\mathrm{st}}$ respectively), hence, $\mathrm{THM(BL)} = \mathrm{TAUT}(\mathbb{BL}^{\mathrm{st}})$ and $\mathrm{CONS(BL)} = \mathrm{CONS}(\mathbb{BL}^{\mathrm{st}})$, and analogously for SBL.

Apart from that, both BL and SBL are also complete w.r.t. some particular standard BL-algebras. Using the partial embedding technique, it is not difficult to see that any standard BL-algebra which is an ordinal sum with a first component that is an Ł-component and with infinitely many Ł-components generates the full variety $\mathbb{BL}$; the converse also holds.

PROPOSITION 5.2.1 ([1]). *Let* $\boldsymbol{A} = \bigoplus_{i \in I} \boldsymbol{A}_i$ *be a standard* BL*-algebra. Then* $\mathbf{V}(\boldsymbol{A}) = \mathbb{BL}$ *iff there is a first component* $\boldsymbol{A}_{i_0}$*, which is an* Ł*-component, and for infinitely many* $i \in I$*,* $\boldsymbol{A}_i$ *is an* Ł*-component. If that is the case, then also* $\mathbf{Q}(\boldsymbol{A}) = \mathbb{BL}$*. Hence,* $\mathrm{THM(BL)} = \mathrm{TAUT}(\boldsymbol{A})$ *and* $\mathrm{CONS(BL)} = \mathrm{CONS}(\boldsymbol{A})$.

Analogous results hold for SBL and each standard BL-algebra which is an ordinal sum with infinitely many Ł-components, *not* starting with an Ł-component.

PROPOSITION 5.2.2. *Let* $\boldsymbol{A} = \bigoplus_{i \in I} \boldsymbol{A}_i$ *be a standard* BL*-algebra. Then* $\mathbf{V}(\boldsymbol{A}) = \mathbb{SBL}$ *iff for infinitely many* $i \in I$*,* $\boldsymbol{A}_i$ *is an* Ł*-component, and either there is a first component* $\boldsymbol{A}_{i_0}$*, which is* not *an* Ł*-component, or there is no first component. If that is the case, then also* $\mathbf{Q}(\boldsymbol{A}) = \mathbb{SBL}$*. Hence,* $\mathrm{THM(SBL)} = \mathrm{TAUT}(\boldsymbol{A})$ *and* $\mathrm{CONS(SBL)} = \mathrm{CONS}(\boldsymbol{A})$.

These facts will be used later when dealing with complexity of individual standard BL-algebras: the two above types of sums will not be considered, as they generate $\mathbb{BL}$ or $\mathbb{SBL}$ and hence the results for BL and SBL (obtained in this subsection) apply.

Let us now investigate TAUT and CONS in $\mathbb{BL}^{\mathrm{st}}$ and $\mathbb{SBL}^{\mathrm{st}}$. The results for SAT, $\mathrm{SAT_{pos}}$, and $\mathrm{TAUT_{pos}}$ for the class $\mathbb{BL}^{\mathrm{st}}$ will be obtained as Corollary 5.3.3; for $\mathbb{SBL}^{\mathrm{st}}$, use Theorem 3.2.4. For TAUT and CONS, we rely on the following lemma.

LEMMA 5.2.3. *Let* $\Phi(x_1, \ldots, x_k)$ *be an open formula in the language of* BL*. Then* $\mathbb{BL}^{\mathrm{st}} \models \Phi$ *iff* $\boldsymbol{A} \models \Phi$ *for each standard* BL*-algebra* $\boldsymbol{A}$ *with at most* $k+1$ *components.*

Proof. The left-to-right implication holds by definition. For the converse one, we give a partial embedding argument. If $\mathbb{BL}^{\mathrm{st}} \not\models \Phi$, there is a standard BL-algebra $\boldsymbol{A}$ and an evaluation $e_{\boldsymbol{A}}$ such that $\boldsymbol{A} \not\models \Phi[e_{\boldsymbol{A}}]$. Write $a_j = e_{\boldsymbol{A}}(x_j)$ for $1 \leq j \leq k$. Possibly re-enumerating, w.l.o.g. assume $a_1 \leq a_2 \leq \cdots \leq a_k$. For $1 \leq j \leq k$, let $\boldsymbol{A}_j$ be the component of $\boldsymbol{A}$ s.t. $a_j \in \boldsymbol{A}_j$; if $a_j \in \boldsymbol{A}_{i_1} \cap \boldsymbol{A}_{i_2}$ for $i_1 < i_2$, let $j = i_2$. It follows from Theorem 2.1.6 that $\{0^{\boldsymbol{A}}\} \cup \bigcup_{1 \leq j \leq k} \boldsymbol{A}_j \cup \{1^{\boldsymbol{A}}\}$ is a BL-subchain of $\boldsymbol{A}$, so for any term t occurring in Φ, we have $e_{\boldsymbol{A}}(t) \in \{0^{\boldsymbol{A}}\} \cup \bigcup_{1 \leq j \leq k} \boldsymbol{A}_j \cup \{1^{\boldsymbol{A}}\}$. Let $\boldsymbol{A}_0$ be the first component of the ordinal sum $\boldsymbol{A}$ if there is one, if not, then let $\boldsymbol{A}_0$ be any component. If $\boldsymbol{A}_j$ is a trivial component for $1 \leq j \leq k$, replace it with a G-component. Define $\boldsymbol{A}' = \bigoplus_{j \leq k} \boldsymbol{A}_j$. If $e(x_j) = a_j$ in $\boldsymbol{A}'$, then $\boldsymbol{A}' \not\models \Phi[e]$. The BL-chain $\boldsymbol{A}'$ is isomorphic to a standard BL-algebra $\boldsymbol{B}$ with at most $k+1$ components via some f, and $\{0^{\boldsymbol{A}}\} \cup \bigcup_{1 \leq j \leq k} \boldsymbol{A}_j \cup \{1^{\boldsymbol{A}}\}$ is isomorphic to a subchain of $\boldsymbol{B}$. Define $e_{\boldsymbol{B}}$ in $\boldsymbol{B}$ s.t. $e_{\boldsymbol{B}}(x_j) = f^{-1}(a_j)$ for $x_1, \ldots, x_k$; then $\boldsymbol{B} \not\models \Phi[e_{\boldsymbol{B}}]$, hence $\boldsymbol{B} \not\models \Phi$. □

Combining Lemma 5.1.3 with the above proof, one can replace each copy of $[0,1]_\Pi$ with two copies of $[0,1]_Ł$ (and it is easy to replace copies of $[0,1]_G$ by a suitable number of $[0,1]_Ł$ copies as well). Then the partial embedding gives us the easy implication from

Proposition 5.2.1. For SBL, reasoning in full analogy (with the proviso that the first component—if any—is not Ł), we get $\mathbb{SBL}^{\mathrm{st}} \models \Phi$ iff $\boldsymbol{A} \models \Phi$ for each standard SBL-algebra $\boldsymbol{A}$ with at most $k+1$ components.

THEOREM 5.2.4. $\mathrm{Th}_\forall(\mathbb{BL}^{\mathrm{st}})$ *and* $\mathrm{Th}_\forall(\mathbb{SBL}^{\mathrm{st}})$ *are* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from Theorem 3.4.1; we address $\mathbf{coNP}$-containment for the case of BL, with comments on modifications for the SBL case.

We work with the complement of $\mathrm{Th}_\forall(\mathbb{BL}^{\mathrm{st}})$. The complement consists of universal sentences Φ that do not hold in at least one standard BL-algebra; by Lemma 5.2.3, one can limit oneself to a class $\mathbb{C}$ of finite sums of cardinality bounded polynomially by $|\Phi|$. Equivalently, one can consider the set of existential sentences Φ that are valid in at least one standard BL-algebra in the class $\mathbb{C}$. We give a nondeterministic algorithm that accepts $\mathrm{Th}_\exists(\mathbb{BL}^{\mathrm{st}})$ (its modification accepts $\mathrm{Th}_\exists(\mathbb{SBL}^{\mathrm{st}})$) and works in time polynomial in the input size.

ALGORITHM EX-BL `//accepts` $\mathrm{Th}_\exists(\mathbb{BL}^{\mathrm{st}})$

`input:` Φ `// existential sentence in the language of BL`

`begin`

`normalForm()` Using Lemma 3.3.5, transform Φ into a (logically equivalent) sentence in existential normal form, $\exists x_1 \ldots \exists x_n(\Phi_1 \wedge \Phi_2)$. Note that n is bounded polynomially by $|\Phi|$, in particular, $n \in O(|\Phi|)$.

`guessOrdinalSum()` Guess a $k \in \mathsf{N}$, $k \leq n+1$. Guess an ordinal sum $\boldsymbol{A}$ of k components (i.e., a sequence of k symbols out of Ł, G, Π).
`// For SBL, the first component is Π.`

`componentDelimiters()` Introduce constants $\frac{1}{k}, \ldots, \frac{k-1}{k}, 1$ for the idempotent elements of **A** that delimit its components, in their real order, (0 we already have). Set $V = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\} \cup \{x_1, \ldots, x_n\}$.

`guessOrder()` Guess a linear ordering $\leq_0$ of elements of V, in such a way that $\leq_0$ preserves the strict ordering of the constants imposed by the delimiters they represent.

`For` $i < k$, we say that any variable $x \in V$ such that $\frac{i}{k} \leq_0 x \leq_0 \frac{i+1}{k}$ *belongs to* i.

`checkOrder()` Check that $\leq_0$ is consistent with Ψ_1, as in the proof of Theorem 4.1.2.

`checkExternal()` Check that $\leq_0$ is compatible with all identities in Φ_2, as far as the operations are order-determined, as follows. Consider each atomic formula in Φ_2. In case of:

- $x = 0$ for some variable x, check $x =_0 0$ ($x =_0 1$ respectively).
- $x * y = z$ for some variables x, y, z, then if, for some $l \leq k$, we have $x \leq_0 \frac{l}{k} \leq_0 y$, then check that $z =_0 x$; analogously for $y \leq_0 \frac{l}{k} \leq_0 x$. If, on the other hand, for some $l < k$, we have $\frac{l}{k} \leq x, y \leq \frac{l+1}{k}$, then check $\frac{l}{k} \leq z \leq \frac{l+1}{k}$.
- $x \to y = z$ for some variables x, y, z, then if $x \leq_0 y$, check $z =_0 1$. If, $x >_0 y$, then if, for some $l \leq k$, we have $x \geq_0 \frac{l}{k} >_0 y$, then check $z =_0 y$; if, on the other hand, for some $l < k$ we have $\frac{l}{k} \leq x, y \leq \frac{l+1}{k}$, then check $\frac{l}{k} \leq z \leq \frac{l+1}{k}$.

`checkInternal()` For each $\leq_0$-interval $[\frac{l}{k}, \frac{l+1}{k}]$, $l = 0, \ldots, k-1$, check that $\leq_0$ is compatible with all identities in Φ_2 for the variables belonging to l. Working for a fixed l, consider Φ_2 restricted to those identities where at least one variable belongs to l. Construct a system $\mathcal{S}_l$ of identities, as follows. $\mathcal{S}_l$ is initially empty. Consider each identity in Φ_2. In case of:

- $x = 0$ for some variable x, add $x = 0$ ($x = 1$ respectively) into $\mathcal{S}_l$.
- $x * y = z$ for some variables x, y, z, then if x and y belong to l, add $x * y = z$ into $\mathcal{S}_l$.
- $x * y = z$ for some variables x, y, z, then if x and y belong to l, add $x \to y = z$ into $\mathcal{S}_l$.

Further, add a conjunction of atomic formulas defining $\leq_0$ for $\frac{l}{k}$, $\frac{l+1}{k}$, and the variables in l, into $\mathcal{S}_l$; replace $\frac{l}{k}$ with 0 and $\frac{l+1}{k}$ with 1. In case the l-th component of $\boldsymbol{A}$ is:

- an Ł-component, use ALGORITHM EX-L to check satisfiability of $\mathcal{S}_l$ in $[0,1]_{Ł}$.
- a Π-component, use ALGORITHM EX-PRODUCT to check satisfiability of $\mathcal{S}_l$ in $[0,1]_{\Pi}$.
- a G-component, use the external check based on $\leq_0$.

`end` □

Because both BL and SBL enjoy finite strong standard completeness, we may conclude:

COROLLARY 5.2.5. *The sets* THM(BL), CONS(BL), THM(SBL), CONS(SBL) *are* **coNP***-complete.*

5.3 Other logics of standard BL-algebras

The term 'logics of standard BL-algebras' may be ambiguous. Suppose that, given a class $\mathbb{K}$ of standard BL-algebras, we do have a clear idea what is meant by a 'logic of $\mathbb{K}$'. Then one reading of the term is that for each standard BL-algebra $\boldsymbol{A}$, we consider the logic of $\boldsymbol{A}$; another reading is that we take arbitrary nonempty classes of standard BL-algebras and for each, we consider the logic of $\mathbb{K}$. Most of this section is dedicated to discussing the former meaning; at the end, we give some remarks on the latter. All of this is going to happen in view of previously presented results that addressed particularly important choices of standard BL-algebras; here, we cater for the "remaining cases".

PROPOSITION 5.3.1 ([14]). *Let $\boldsymbol{A}$ be a standard* BL*-algebra. Then the logic of $\boldsymbol{A}$ is an axiomatic extension of* BL *obtained by adding finitely many axioms.*

We investigate each given standard BL-algebra as to the complexity of its SAT, TAUT, and CONS problems. We start with some easy results; recall that in the ordinal-sum decomposition of a standard BL-algebra $\boldsymbol{A}$, either $\boldsymbol{A}$ has a first component Ł, or $\boldsymbol{A}$ is an SBL-algebra.

THEOREM 5.3.2. *Let $\mathbb{K}$ be a nonempty class of standard* BL*-algebras and let each $\boldsymbol{A} \in \mathbb{K}$ be of the type* Ł $\oplus$ X *for some ordinal sum X (possibly void). Then*

(i) $\mathrm{TAUT}_{\mathrm{pos}}(\mathbb{K}) = \mathrm{TAUT}_{\mathrm{pos}}([0,1]_{Ł})$;

(ii) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \mathrm{SAT}_{(\mathrm{pos})}([0,1]_{Ł})$.

Proof. $[0,1]_{Ł}$ is isomorphic to a subalgebra of each $\boldsymbol{A} \in \mathbb{K}$; use Lemma 3.1.3 to obtain the left-to-right inclusion in (i) and the right-to-left inclusion in (ii). Moreover, for each $\boldsymbol{A} \in \mathbb{K}$, a mapping f sending x to $\neg\neg x$ in $\boldsymbol{A}$ is a homomorphism of $\boldsymbol{A}$ onto (an isomorphic copy of) $[0,1]_{Ł}$; we have $f(x) = 0$ iff $x = 0$ and $f(x) = 1$ if $x = 1$, which yields the converse inclusions. □

COROLLARY 5.3.3.

(i) $\mathrm{TAUT_{pos}}(\mathbb{BL}^{\mathrm{st}}) = \mathrm{TAUT_{pos}}([0,1]_{Ł})$;

(ii) $\mathrm{SAT_{(pos)}}(\mathbb{BL}^{\mathrm{st}}) = \mathrm{SAT_{(pos)}}([0,1]_{Ł})$.

Proof. We have $\mathbb{BL}^{\mathrm{st}} = \mathbb{SBL}^{\mathrm{st}} \cup \mathbb{L}$, where $\mathbb{L}$ denotes the class of standard BL-algebras with a first component Ł.
(i) $\mathrm{TAUT_{pos}}(\mathbb{BL}^{\mathrm{st}}) = \mathrm{TAUT_{pos}}(\mathbb{SBL}^{\mathrm{st}}) \cap \mathrm{TAUT_{pos}}(\mathbb{L}) =$
$\mathrm{TAUT}(\{0,1\}_{\mathrm{B}}) \cap \mathrm{TAUT_{pos}}([0,1]_{Ł})$, where the last equality holds by combining Theorem 3.2.4 with the theorem above. The statement then follows from Lemma 3.1.4.
(ii) $\mathrm{SAT_{(pos)}}(\mathbb{BL}^{\mathrm{st}}) = \mathrm{SAT_{(pos)}}(\mathbb{SBL}^{\mathrm{st}}) \cup \mathrm{SAT_{(pos)}}(\mathbb{L}) =$
$\mathrm{SAT}(\{0,1\}_{\mathrm{B}}) \cup \mathrm{SAT_{(pos)}}([0,1]_{Ł})$, analogously to the above case. □

Apply Lemma 4.1.4 to show that the SAT, $\mathrm{SAT_{pos}}$, and $\mathrm{TAUT_{pos}}$ problems for the class of standard BL-algebras are distinct from each other and from the classical case. Compare this to Theorem 3.2.4 for classes of standard SBL-algebras (i.e., not starting with an Ł-component).

5.3.1 Finite sums

THEOREM 5.3.4. *Let* $\boldsymbol{A}$ *be a standard* BL*-algebra which is a finite ordinal sum of* Ł*,* G*, and* Π*-components. Then* $\mathrm{Th}_{\forall}(\boldsymbol{A})$ *is* **coNP***-complete.*

Proof. Hardness follows from Theorem 3.4.1. We prove **NP**-containment for $\mathrm{Th}_{\exists}(\boldsymbol{A})$; the type and cardinality of the sum of $\boldsymbol{A}$ is used as a built-in information. This is the only difference from Algorithm EX-BL, which guesses a finite ordinal sum $\boldsymbol{B}$ and tests whether the given formula is in $\mathrm{Th}_{\exists}(\boldsymbol{B})$. Here, we fix a standard BL-algebra $\boldsymbol{A}$ which is a finite ordered sum of k components, so the `guessOrdinalSum()` step is omitted.

ALGORITHM EX-FIN `//accepts` $\mathrm{Th}_{\exists}(\boldsymbol{A})$

`input:` Φ `// existential sentence in the language of BL`

`begin`

`normalForm()` As in the proof of Theorem 5.2.4.

`componentDelimiters()` As in the proof of Theorem 5.2.4.

`guessOrder()` As in the proof of Theorem 5.2.4.

`checkOrder()` As in the proof of Theorem 5.2.4.

`checkExternal()` As in the proof of Theorem 5.2.4.

`checkInternal()` As in the proof of Theorem 5.2.4.

`end` □

COROLLARY 5.3.5. *Let $\boldsymbol{A}$ be a standard* BL*-algebra that is a finite ordinal sum and let* $\mathrm{L}(\boldsymbol{A})$ *be the logic of* $\boldsymbol{A}$*. Then*

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\boldsymbol{A})$ *is* $\mathbf{coNP}$*-complete*;

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}(\boldsymbol{A})$ *and* $\mathrm{CONS}(\boldsymbol{A})$ *are* $\mathbf{coNP}$*-complete*;

(iii) $\mathrm{THM}(\mathrm{L}(\boldsymbol{A}))$ *and* $\mathrm{CONS}(\mathrm{L}(\boldsymbol{A}))$ *are* $\mathbf{coNP}$*-complete.*

Proof. For SAT, $\mathrm{SAT}_{\mathrm{pos}}$, $\mathrm{TAUT}_{\mathrm{pos}}$, consider that if $\boldsymbol{A}$ has a first component Ł, and then one can use Theorem 5.3.2, otherwise one can use Theorem 3.2.4. □

5.3.2 Infinite sums

Now we consider those standard BL-algebras that are ordinal sums of infinitely many components. As stated in Propositions 5.2.1 and 5.2.2, a standard BL-algebra $\boldsymbol{A}$ that is an ordinal sum with infinitely many Ł-components generates either the variety $\mathbb{BL}$ (in case the sum has a first component Ł) or the variety $\mathbb{SBL}$ (otherwise). In both cases, the respective variety is generated as a quasivariety by $\boldsymbol{A}$. Hence, it is sufficient to deal with standard BL-algebras with *finitely many* Ł*-components*.

A standard BL-algebra with finitely many, say n, Ł-components seems much more tangible than the general case: one can think about it in terms of $n+1$ ordinal subsums that are without Ł-components and sit inbetween the n Ł-components. Some of these sums may be finite ordinal sums of G- and Π-components, some others may be infinite sums thereof, and some may be void. The infinite sums might be a problem, because there are too many such infinite sums for a finite description. Fortunately, from the point of view of the (quasi)equational theory of the algebra, we need not describe the infinite sums of G- and Π-components exactly, as the following analysis shows.

Consider two algebras $\boldsymbol{X}$, $\boldsymbol{Y}$, where either is a standard BL-algebra or the trivial one-element algebra. Let us take standard BL-algebras $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ that are two arbitrary infinite sums of G- and Π-components. It is easy to see that $\boldsymbol{Z}_1$ is partially embeddable into $\boldsymbol{Z}_2$ and vice versa, and therefore the standard BL-algebras $\boldsymbol{X} \oplus \boldsymbol{Z}_1 \oplus \boldsymbol{Y}$ and $\boldsymbol{X} \oplus \boldsymbol{Z}_2 \oplus \boldsymbol{Y}$ have the same universal theory. So the universal theory of any standard BL-algebra can be encoded by a finite string in the alphabet $Ł, \mathrm{G}, \Pi, \infty Ł, \infty\Pi$.

DEFINITION 5.3.6 (Canonical Standard BL-algebra). *A standard* BL*-algebra is canonical iff it is an ordinal sum that is either* ∞Ł, $\Pi \oplus \infty$Ł*, or a finite sum of* Ł, G, Π*, and* $\infty\Pi$*, where no* G *is preceded or followed by another* G*, and no* $\infty\Pi$ *is preceded or followed by a* G*, a* Π*, or another* $\infty\Pi$*.*

PROPOSITION 5.3.7. *For each standard* BL*-algebra $\boldsymbol{A}$ there is a canonical standard* BL*-algebra $\boldsymbol{A}'$ such that* $\mathrm{Th}_\forall(\boldsymbol{A}) = \mathrm{Th}_\forall(\boldsymbol{A}')$.

Not only canonical standard BL-algebras generate all possible subvarieties of $\mathbb{BL}^{\mathrm{st}}$ that are generated by any single standard BL-algebra, but, as shown in [14], nonisomorphic canonical standard BL-algebras generate distinct subvarieties. Hence, the strings in the alphabet $\{Ł, \infty Ł, \mathrm{G}, \Pi, \infty\Pi\}$ give a finite-string representation of all varieties (and all sets of propositional tautologies) given by a single standard BL-algebra. Moreover each of these varieties is generated as a quasivariety by the standard BL-algebra, so this consideration extends also to the respective quasivarieties (finite consequence relations).

As in previous cases, we address the universal theory of each of the canonical standard BL-algebras with finitely many L-components (and thus, no ∞Ł-components).

THEOREM 5.3.8. *Let $\boldsymbol{A}$ be a canonical standard* BL*-algebra which is an infinite ordinal sum with finitely many* L*-components. Then* $\mathrm{Th}_\forall(\boldsymbol{A})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from Theorem 3.4.1. We prove $\mathbf{NP}$-containment of $\mathrm{Th}_\exists(\boldsymbol{A})$, using an algorithm that (accepts the given set and) relies on a built-in information about $\boldsymbol{A}$ in terms of a string of symbols. It works as follows: first, it guesses another finite sum $\boldsymbol{B}$ of Ł, G, and Π, whose cardinality is linear in the input size. Then, it checks that the ordinal sum $\boldsymbol{B}$ is a subsum of the sum of $\boldsymbol{A}$, in such a way that a first Ł-component in $\boldsymbol{B}$ is also a first component in $\boldsymbol{A}$; if this is true, then $\boldsymbol{B}$ is a subalgebra of $\boldsymbol{A}$ and thus, any solution found in $\boldsymbol{B}$ will be a solution in $\boldsymbol{A}$ also. Finally, it tests the input existential formula for validity in $\boldsymbol{B}$.

ALGORITHM EX-INF `//accepts` $\mathrm{Th}_\exists(\boldsymbol{A})$

`input:` Φ `// existential sentence in the language of BL`

`begin`

`normalForm()` As in the proof of Theorem 5.2.4 (in particular, n is the number of variables in the formula $\Phi_1 \wedge \Phi_2$).

`guessOrdinalSum()` Guess a $k \in \mathsf{N}$, $k \leq n+1$.
Guess an ordinal sum $\boldsymbol{B}$ of k components (a sequence of k symbols out of Ł, G, Π).

`checkEmbedding()` Check whether the sum of $\boldsymbol{B}$ is embeddable into the sum of $\boldsymbol{A}$ (as a sequence of symbols into a sequence of symbols), in such a way that an initial Ł of $\boldsymbol{B}$ (if any) is mapped to an initial Ł in $\boldsymbol{A}$.
From now on, work with $\boldsymbol{B}$ instead of $\boldsymbol{A}$.

`componentDelimiters()` As in the proof of Theorem 5.2.4.

`guessOrder()` As in the proof of Theorem 5.2.4.

`checkOrder()` As in the proof of Theorem 5.2.4.

`checkExternal()` As in the proof of Theorem 5.2.4.

`checkInternal()` As in the proof of Theorem 5.2.4.

`end`

Let us look at the `checkEmbedding()` step. First of all, we discuss this is a subroutine working in time polynomial in k, hence in $|\Phi|$. As a matter of fact, the check can be done deterministically in polynomial time; but it is simpler and sufficient to present the nondeterministic check. Take the finite-string representation of $\boldsymbol{B}$ (in the alphabet Ł, G, Π), for each element of the sum of $\boldsymbol{B}$, guesses a natural number points into the finite-string representation of $\boldsymbol{A}$ (in the alphabet Ł, G, Π, ∞Π), then checks that this assignment is a one-one embedding in terms of components (more than one Π-component in $\boldsymbol{B}$ can be mapped onto a single ∞Π-component of $\boldsymbol{A}$), and that it satisfies the condition that if Ł is initial in $\boldsymbol{B}$, then it is mapped onto an initial L-component in $\boldsymbol{A}$. This is a polynomial-time procedure: the cardinality of $\boldsymbol{B}$ is bounded polynomially by $|\Phi|$ and the cardinality of $\boldsymbol{A}$ (as a finite string in the alphabet Ł, G, Π, ∞Π) is fixed.

If there is a satisfying evaluation in $\boldsymbol{A}$, then one can find a finite subsum harbouring it; we know by Lemma 5.2.3 it is enough to search all finite subsums up to length $k+1$. The algorithm works with each such subsum as finite sum and works in exactly the same way as in the case for finite sums.

It is perhaps worth remarking that under this construction, the algorithm "throws away" some of the constructed sums which actually could supply a satisfying evaluation embeddable into $\boldsymbol{A}$; for example, if we permitted G-components in $\boldsymbol{B}$ to map onto $\infty\Pi$ components, the algorithm would still be correct because, if later the algorithm guesses an ordering that assigns finitely many idempotent values into this G-component, then these could map onto some delimiting idempotent values in the $\infty\Pi$-segment. We prefer, however, to work solely with the string representations and not to go back to the structure of the standard BL-algebras. □

COROLLARY 5.3.9. *Let $\boldsymbol{A}$ be a standard* BL*-algebra that is an infinite ordinal sum and let* $\mathrm{L}(\boldsymbol{A})$ *be the logic of* $\boldsymbol{A}$*. Then*

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\boldsymbol{A})$ *is* $\mathbf{coNP}$*-complete*;

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}(\boldsymbol{A})$ *and* $\mathrm{CONS}(\boldsymbol{A})$ *are* $\mathbf{coNP}$*-complete*;

(iii) $\mathrm{THM}(\mathrm{L}(\boldsymbol{A}))$ *and* $\mathrm{CONS}(\mathrm{L}(\boldsymbol{A}))$ *are* $\mathbf{coNP}$*-complete.*

5.3.3 Logics given by classes of standard BL-algebras

Let us first discuss the case that $\mathbb{K}$ is finite. The following is a consequence of a more general result of [17].

PROPOSITION 5.3.10. *If* $\mathbb{K}$ *is a finite, nonempty class of standard* BL*-algebras, the logic of* $\mathbb{K}$ *is an axiomatic extension of* BL *obtained by adding finitely many axioms.*

Recall that $\mathbf{NP}$ is closed under finite unions, and consider that $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K}) = \bigcup_{\boldsymbol{A}\in\mathbb{K}} \mathrm{SAT}_{(\mathrm{pos})}(\boldsymbol{A})$; $\mathbf{coNP}$ is closed under finite intersections, and $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K}) = \bigcap_{\boldsymbol{A}\in\mathbb{K}} \mathrm{TAUT}_{(\mathrm{pos})}(\boldsymbol{A})$ and $\mathrm{CONS}(\mathbb{K}) = \bigcap_{\boldsymbol{A}\in\mathbb{K}} \mathrm{CONS}(\boldsymbol{A})$. On that basis, and in view of Theorem 3.4.1, we may conclude:

THEOREM 5.3.11. *If* $\mathbb{K}$ *is a finite, nonempty class of standard* BL*-algebras and* $\mathrm{L}(\mathbb{K})$ *is the logic of* $\mathbb{K}$*, then*

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K})$ *is* $\mathbf{NP}$*-complete*;

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$ *and* $\mathrm{CONS}(\mathbb{K})$ *are* $\mathbf{coNP}$*-complete*;

(iii) $\mathrm{THM}(\mathrm{L}(\mathbb{K}))$ *and* $\mathrm{CONS}(\mathrm{L}(\mathbb{K}))$ *are* $\mathbf{coNP}$*-complete.*

Now let us address the case when $\mathbb{K}$ is an arbitrary (possibly infinite) class of standard BL-algebras. We rely on the following statement.

PROPOSITION 5.3.12 ([24]). *Let* $\mathbb{K}$ *be a class of standard* BL*-algebras. Then there is a finite class* $\mathbb{L}$ *of standard* BL*-algebras such that* $\mathbf{V}(\mathbb{K}) = \mathbf{V}(\mathbb{L})$.

COROLLARY 5.3.13. *Let* $\mathbb{K}$ *be a nonempty class of standard* BL*-algebras. Then*

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{K})$ *is* $\mathbf{NP}$*-complete*;

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{K})$ *is* $\mathbf{coNP}$*-complete.*

Proof. For TAUT the statement follows from the previous one. For SAT, $\mathrm{SAT_{pos}}$, and $\mathrm{TAUT_{pos}}$, recall that we can partition $\mathbb{K}$ into a class $\mathbb{K}_{Ł}$ of those algebras in $\mathbb{K}$ that have a first component Ł, and a class $\mathbb{K}_{\bar{Ł}}$ of the remaining algebras in $\mathbb{K}$. Then use Theorem 5.3.2 and Theorem 3.2.4 for the two classes, and Lemma 3.1.4 on their union/intersection. □

6 Logics in modified languages

In this section we discuss some fragments and expansions of BL and MTL and their extensions. As regards fragments, we limit ourselves to dropping $\overline{0}$, i.e., we consider the falsehood-free language of hoops. For expansions, we consider in particular logics with $\triangle$, logics with new propositional constants, logics with an involutive negation as an independent connective, and the logics ŁΠ and $Ł\Pi\frac{1}{2}$.

6.1 Falsehood-free fragments

MTLH (monoidal t-norm hoop logic) is obtained from MTL by dropping $\overline{0}$ from the language and dropping the axiom $\overline{0} \to \varphi$. BLH (basic hoop logic) is obtained from BL in the same manner. BLH can be extended to ŁH, GH, and ΠH in exactly the same way as BL extends to Ł, G, and Π. Besides, *cancellative hoop logic* CHL is obtained from BLH by adding the axiom $(\varphi \to (\varphi \,\&\, \psi)) \to \psi$.

PROPOSITION 6.1.1 (Conservativeness [13]).

(i) MTL, IMTL, *and* SMTL *are conservative expansions of* MTLH;

(ii) BL *and* SBL *are conservative expansions of* BLH;

(iii) Ł *is a conservative expansion of* ŁH;

(iv) G *is a conservative expansion of* GH;

(v) Π *is a conservative expansion of* ΠH.

So each of the falsehood-free fragments inherits the complexity class of its counterpart in the full language (cf. Lemma 3.1.2). For CHL, we use the following fact:

PROPOSITION 6.1.2. *Let* $\mathcal{L}$ *be the language of hoops and* $T \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$. *Then* $T \vdash_{\mathrm{CHL}} \varphi(p_1, \ldots, p_n)$ *iff* $T \vdash_{\Pi} (\bigwedge_{i=1}^{n} \neg\neg p_i) \to \varphi$.

COROLLARY 6.1.3. $\mathrm{CONS(CHL)} \preceq_{\mathbf{P}} \mathrm{CONS}(\Pi)$, *so* CONS(CHL) *is in* $\mathbf{coNP}$.

In the following we prove $\mathbf{coNP}$-hardness for theoremhood in BLH, ŁH, GH, ΠH, CHL. Let p be a new variable, and consider the following translation function, operating on BL-formulas:

- if φ is $\overline{0}$ then φ° is p
- if φ is atomic then φ° is $\varphi \vee p$
- if φ is $\psi \,\&\, \chi$ then φ° is $(\psi^\circ \,\&\, \chi^\circ) \vee p$
- if φ is $\psi \to \chi$ then φ° is $\psi^\circ \to \chi^\circ$

PROPOSITION 6.1.4 (Interpretation in falsehood-free fragments [13]). *Let* L *be* MTL, BL, Ł, *or* G, LH *its falsehood-free fragment, and* φ *a formula in the language of* MTL. *Then*

(i) $\vdash_{\mathrm{L}} \varphi$ *iff* $\vdash_{\mathrm{LH}} \varphi^\circ$;

(ii) $\vdash_{\Pi\mathrm{H}} \varphi^\circ$ *iff* $\vdash_{\Pi} \varphi$ *and* $\vdash_{Ł} \varphi$;

(iii) $\vdash_{\mathrm{CHL}} \varphi^\circ$ *iff* $\vdash_{Ł} \varphi$.

COROLLARY 6.1.5. *Theoremhood (and hence, provability from finite theories) in the logics* BLH, $Ł\mathrm{H}$, GH, $\Pi\mathrm{H}$, *and* CHL *is* $\mathbf{coNP}$*-hard, hence* $\mathbf{coNP}$*-complete. Moreover,* $\mathrm{MTL} \approx_{\mathbf{P}} \mathrm{MTLH}$.

Proof. The set $\mathrm{THM}(Ł) \cap \mathrm{THM}(\Pi) = \mathrm{TAUT}([0,1]_{Ł}) \cap \mathrm{TAUT}([0,1]_{\Pi})$ can be observed to be $\mathbf{coNP}$-hard by Theorem 3.4.1; hence $\mathbf{coNP}$-hardness for $\Pi\mathrm{H}$. The other cases are straightforward. □

Satisfiability for (classes of) hoops is not an interesting problem. If $\mathcal{L}$ is the language of hoops and $\mathbb{K}$ is a nonempty class of $\mathcal{L}$-hoops, then $\mathrm{SAT}(\mathbb{K}) = \mathrm{SAT}_{\mathrm{pos}}(\mathbb{K}) = Fm_{\mathcal{L}}$ (clearly in any hoop, each formula is satisfiable by evaluating all of its variables with the value 1).

6.2 Logics with $\triangle$

Let L be MTL or its extension. The $\triangle$-expansion $\mathrm{L}_\triangle$ of L is obtained by adding the rule of $\triangle$-generalization: from φ derive $\triangle\varphi$, and the following axioms:

($\triangle$1) $\triangle\varphi \vee \neg\triangle\varphi$

($\triangle$2) $\triangle(\varphi \vee \psi) \to (\triangle\varphi \vee \triangle\psi)$

($\triangle$3) $\triangle\varphi \to \varphi$

($\triangle$4) $\triangle\varphi \to \triangle\triangle\varphi$

($\triangle$5) $\triangle(\varphi \to \psi) \to (\triangle\varphi \to \triangle\psi)$

Recall that the deduction theorem for a logic $\mathrm{L}_\triangle$ reads as follows: $T \cup \{\varphi\} \vdash_{\mathrm{L}_\triangle} \psi$ iff $T \vdash_{\mathrm{L}_\triangle} \triangle\varphi \to \psi$. Hence $\mathrm{CONS}(\mathrm{L}_\triangle) \approx_{\mathbf{P}} \mathrm{THM}(\mathrm{L}_\triangle)$ and it is sufficient to investigate complexity of the set $\mathrm{THM}(\mathrm{L}_\triangle)$ (cf. Lemma 3.1.1).

Adding $\triangle$ in the above manner expands the logic L conservatively; in particular, we can expand each standard L-algebra into a $\mathrm{L}_\triangle$-algebra, and then prove standard completeness results.

We remark that for any logic L, in any $\mathrm{L}_\triangle$-chain $\boldsymbol{A}$, the semantic counterpart of $\triangle$ is the function given by $\triangle(1) = 1$, $\triangle(x) = 0$ for $x \neq 1$.

THEOREM 6.2.1. *Let* L *be the logic of a standard* BL*-algebra. Then* $\mathrm{THM}(\mathrm{L}_\triangle)$ *is* $\mathbf{coNP}$*-complete.*

Proof. For hardness, combine Lemma 3.1.2 with results on standard L-algebras. Containment by an inspection of containment proof for L; it is sufficient to note that the $\triangle$-operation is order-determined. In this way, we actually obtain $\mathbf{coNP}$-completeness of the universal fragment of the theory of the appropriate standard $\mathrm{L}_\triangle$-algebra. □

6.3 Logics with constants

Expanding the language with constants is addressed in a comprehensive manner in Chapter VIII. We rely on [11] for the general framework and on [22] for results. Admittedly, the results are rather fragmentary and there are some open and many unattempted problems. We discuss three examples: Łukasiewicz logic, Gödel logic, and product logic, each expanded with new propositional constants as explained below.

In a general setting, one expands the language with names for some elements of a chosen algebra, often a standard one. In that case, one takes an arbitrary $C \subseteq [0,1]$, countable and closed under all operations, to be the canonical semantics of new constants. But in order to be able to reason about complexity, we need much more: elements of C should be representable by finite words, C should be decidable (preferably in $\mathbf{P}$), and the C-words should admit feasible evaluation of operations. Thus we restrict our attention to the case $C = \mathsf{Q} \cap [0,1]$.

Let $[0,1]_*$ be a standard BL-algebra. Let $\mathcal{Q} = \{\overline{q} \mid q \in \mathsf{Q} \cap (0,1)\}$ be a set of new propositional constants. If $\mathcal{L}$ is the language of BL, define $\mathcal{L}^{\mathcal{Q}} = \mathcal{L} \cup \mathcal{Q}$. If L is the logic of $[0,1]_*$, then $\mathrm{L}(\mathcal{Q})$ in the language $\mathcal{L}^{\mathcal{Q}}$ expands L with the *bookkeeping axioms*

$$\overline{r} \mathbin{\&} \overline{s} \leftrightarrow \overline{r * s} \quad \text{and} \quad \overline{r} \to \overline{s} \leftrightarrow \overline{r \to s}$$

for each $r, s \in \mathsf{Q} \cap [0,1]$. Each logic $\mathrm{L}(\mathcal{Q})$ has its equivalent algebraic semantics (L-algebras enriched with constants, satisfying the axioms), it has standard semantics (standard L-algebras, ditto), but we are interested primarily in its *canonical* semantics, which is given by the initial standard algebra $[0,1]_*$ and the canonical interpretation of $\overline{q}$ with q for each $q \in \mathsf{Q} \cap [0,1]$. If $[0,1]_*$ is a standard BL-algebra, L is the logic of $[0,1]_*$, then $[0,1]_*^{\mathcal{Q}}$ denotes the canonical $\mathrm{L}(\mathcal{Q})$-algebra. Now let us see where bookkeeping axioms can take us, completeness-wise.

PROPOSITION 6.3.1 ([21, 39]). *Let* $[0,1]_*$ *and* L *be as above. Then*

(i) $\mathrm{L}(\mathcal{Q})$ *has the canonical FSSC iff* $*$ *is the Łukasiewicz t-norm*;

(ii) $\mathrm{L}(\mathcal{Q})$ *has the canonical SC if* $*$ *is the Gödel or the product t-norm.*

Note also that finite strong *standard* completeness (i.e., completeness with respect to the class of all standard algebras) holds for any $\mathrm{L}(\mathcal{Q})$ obtained in this manner, and consequently, each $\mathrm{L}(\mathcal{Q})$ is conservative over L and L is the $\mathcal{L}$-fragment of $\mathrm{L}(\mathcal{Q})$.

Now for complexity: we work in the canonical standard algebras, starting with the expansion of Łukasiewicz logic.

THEOREM 6.3.2. $\mathrm{Th}_\forall([0,1]_{\text{Ł}}^{\mathcal{Q}})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from the result for $[0,1]_{\text{Ł}}$ without constants (Theorem 4.1.2): we have $\mathrm{Th}_\forall([0,1]_{\text{Ł}}) \preceq_{\mathbf{P}} \mathrm{Th}_\forall([0,1]_{\text{Ł}}^{\mathcal{Q}})$ since the former is the fragment of the latter given by restriction to the BL-language; cf. also Lemma 3.1.2.

We show $\mathbf{NP}$-containment of $\mathrm{Th}_\exists([0,1]_{\text{Ł}}^{\mathcal{Q}})$ by a discussion of ALGORITHM EX-L accepting $\mathrm{Th}_\exists([0,1]_{\text{Ł}})$. In the case with constants, the input is an existential sentence Φ with constants, represented as two integers in binary. As before, we first transform Φ into an existential normal form (see Lemma 3.3.5). In the normal form for Φ we have, in addition to the former, also identities of the form $x = c$ for x a variable, c a constant. Then we guess an ordering $\leq_0$ of $V = \{x_1, \ldots, x_n, c_1, \ldots, c_k\}$, where $c_1, \ldots, c_k$ are those constants in Φ, and 0 and 1. Then we check that Φ_1 is compatible with $\leq_0$. Then we need to translate each equation $x = c$ into the language of linear programming. But that is easy, since for c being $\frac{p}{q}$, we may translate the equation $x = \frac{p}{q}$ with $xq = p$. The rest is as before. □

COROLLARY 6.3.3.

(i) $\mathrm{SAT}_{(\mathrm{pos})}([0,1]^{\mathcal{Q}}_{\text{Ł}})$ *is* $\mathbf{NP}$*-complete.*

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}([0,1]^{\mathcal{Q}}_{\text{Ł}})$ *and* $\mathrm{CONS}([0,1]^{\mathcal{Q}}_{\text{Ł}})$*, are* $\mathbf{coNP}$*-complete.*

(iii) $\mathrm{THM}(\text{Ł}(\mathcal{Q}))$ *and* $\mathrm{CONS}(\text{Ł}(\mathcal{Q}))$ *are* $\mathbf{coNP}$*-complete.*

Proof. For hardness, use Lemma 3.1.2. For (i), (ii), consider Lemma 3.1.1. For (iii), use finite strong canonical completeness of Ł$(\mathcal{Q})$. □

THEOREM 6.3.4. $\mathrm{Th}_\forall([0,1]^{\mathcal{Q}}_{\mathrm{G}})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from follows from Theorem 5.1.1 and Lemma 3.1.2. We show $\mathbf{NP}$-containment of $\mathrm{Th}_\exists([0,1]^{\mathcal{Q}}_{\mathrm{G}})$, modifying ALGORITHM EX-G (we are only considering the standard algebra) to cater for truth constants. Let Φ be an existential sentence $\exists x_1 \ldots \exists x_n \Phi(x_1, \ldots, x_n)$ in the language of $([0,1]^{\mathcal{Q}}_{\mathrm{G}})$, where Φ is an open Boolean combination of identities. Let $\frac{k_1}{l_1}, \ldots, \frac{k_m}{l_m}$ be a list of truth constants in Φ distinct from 0, 1. Guess a linear ordering $\leq_0$ of the set $V = \{0, 1, x_1, \ldots, x_n, \frac{k_1}{l_1}, \ldots, \frac{k_m}{l_m}\}$, such that 0 is at the bottom, 1 is at the top, and the natural order of the constants is preserved. Compute the value of each atomic formula in Φ: on the basis of $\leq_0$, evaluate all atomic formulas. Evaluate the Boolean combination Φ, accept iff the result is 1. □

THEOREM 6.3.5.

(i) $\mathrm{SAT}_{(\mathrm{pos})}([0,1]^{\mathcal{Q}}_{\mathrm{G}})$ *is* $\mathbf{NP}$*-complete.*

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}([0,1]^{\mathcal{Q}}_{\mathrm{G}})$ *and* $\mathrm{CONS}([0,1]^{\mathcal{Q}}_{\mathrm{G}})$ *are* $\mathbf{coNP}$*-complete.*

(iii) $\mathrm{THM}(\mathrm{G}(\mathcal{Q}))$ *is* $\mathbf{coNP}$*-complete*

(iv) $\mathrm{CONS}(\mathrm{G}(\mathcal{Q}))$ *is* $\mathbf{coNP}$*-complete.*

Proof. (i) to (iii) are clear, but we prove (iv), because CONS in the canonical algebra need not correspond to provability (due to lack of canonical FSSC). For $\mathrm{CONS}(\mathrm{G}(\mathcal{Q}))$, recall that $\mathrm{G}(\mathcal{Q})$ enjoys the FSSC, and observe that for each standard $\mathrm{G}(\mathcal{Q})$-algebra $\boldsymbol{A}$, there is a filter F on the G-algebra of constants $\mathsf{Q} \cap [0,1]$ such that $\boldsymbol{A}$ interprets the elements of F with 1, whereas other constants are interpreted with pairwise distinct elements. If $\{\varphi_1, \ldots, \varphi_n\}, \psi \in \overline{\mathrm{CONS}}(\mathrm{G}(\mathcal{Q}))$, then this shows in some standard $\mathrm{G}(\mathcal{Q})$-algebra given by a filter F on the algebra of constants, and the only information needed about F is which constants *occurring in the input* belong to it. So we may guess a rational (inbetween two constants occurring in the input) and use a variant of the algorithm given above on $\boldsymbol{A}$ obtained in this manner. □

THEOREM 6.3.6. $\mathrm{Th}_\forall([0,1]^{\mathcal{Q}}_\Pi)$ *is in* $\mathbf{PSPACE}$.

Proof. We show $\mathrm{Th}_\exists([0,1]^{\mathcal{Q}}_\Pi) \preceq_{\mathbf{P}} \mathrm{Th}_\exists(\mathrm{RCF})$. Consider, for $k \in \mathsf{N}$, its binary representation $(c_{\lfloor\log(k)\rfloor}\dots c_0)$. Then $k = \sum_{i \leq \lfloor\log(k)\rfloor} c_i.2^i$; let this be the *term* $\overline{k}$ *corresponding to* k in the language of RCF.

Let Φ be an existential $\Pi(\mathcal{Q})$-sentence. Transform it into an existential normal form $\exists x_1 \dots \exists x_n(\Phi_1 \wedge \Phi_2)$. Define Φ_2' as follows. Process the identities in Φ_2 one by one, for $i_1, i_2, i_3 \leq n$, in case of:

– $x_{i_1} = \frac{c}{d}$, replace with $x_{i_1} \cdot \overline{d} = \overline{c}$;
– $x_{i_1} * x_{i_2} = x_{i_3}$, replace with $x_{i_1} \cdot x_{i_2} = x_{i_3}$;
– $x_{i_1} \to x_{i_2} = x_{i_3}$, replace with $((x_{i_1} \leq x_{i_2}) \wedge (x_{i_3} = 1)) \vee$
$((x_{i_1} > x_{i_2}) \wedge (x_{i_1} \cdot x_{i_3} = x_{i_2}))$.

Let Φ_3 denote the formula $\bigwedge_{i=1}^n 0 \leq x_i \wedge x_i \leq 1$ for $1 \leq i \leq n$ (boundary conditions).

Let Φ' denote the formula $\Phi_1 \wedge \Phi_2' \wedge \Phi_3$. Then $\Phi \in \mathrm{Th}_\exists[0,1]^{\mathcal{Q}}_\Pi)$ if and only if $\exists x_1 \dots \exists x_n \Phi' \in \mathrm{Th}_\exists(\mathrm{RCF})$. Moreover, the latter formula can be computed from Φ in time polynomial in $|\Phi|$. □

EXAMPLE 6.3.7. There is a standard BL-algebra $[0,1]_*$ which is an infinite sum of Ł-components such that both $\mathrm{TAUT}([0,1]^{\mathcal{Q}}_*)$ and $\mathrm{SAT}([0,1]^{\mathcal{Q}}_*)$ are nonarithmetical.

Proof. Let $A \subseteq \mathsf{N}$ be any set. Let $[0,1]_*$ be a standard BL-algebra with idempotents 0, 1, and $\frac{1}{n}$ for $n \in A$, and all components isomorphic to $[0,1]_{\text{Ł}}$. Let us introduce rational constants into $[0,1]_*$. Now observe the following: for each $n \in \mathsf{N}$, we have $n \in A$ iff the formula

$$\overline{\tfrac{1}{n}} \mathbin{\&} \overline{\tfrac{1}{n}} \leftrightarrow \overline{\tfrac{1}{n}}$$

is in $\mathrm{TAUT}([0,1]^{\mathcal{Q}}_*)$ (or $\mathrm{SAT}([0,1]^{\mathcal{Q}}_*)$). Therefore, $A \preceq_{\mathrm{m}} \mathrm{TAUT}([0,1]^{\mathcal{Q}}_*)$ and $A \preceq_{\mathrm{m}} \mathrm{SAT}([0,1]^{\mathcal{Q}}_*)$. Fixing A as a nonarithmetical set closes the proof. □

6.4 Logics with an involutive negation

We discuss expansions of a given logic L with a new unary connective $\sim$, which behaves as a decreasing involution. The resulting logic $\mathrm{L}_\sim$ is particularly interesting when the definable negation $\neg$ in the logic is the strict negation, because of the two negations' interplay. This means the cases when L is an extension of SMTL. Then $\triangle\varphi$ is defined as $\neg{\sim}\varphi$. In particular, if L extends SMTL, then $\mathrm{L}_\sim$ results from L by adding the rule $\varphi/\triangle\varphi$ and the axioms

$$\sim\sim\varphi \leftrightarrow \varphi$$
$$\triangle(\varphi \to \psi) \to \triangle({\sim}\psi \to {\sim}\varphi)$$
$$\neg\varphi \to {\sim}\varphi$$

The semantics of $\sim$ on $[0,1]$ is given by decreasing involutions; a prominent example is the function $1-x$ on $[0,1]$, the *canonical* involutive negation. We say that $\mathrm{L}_\sim$-algebra is standard iff its $\mathrm{FL_{ew}}$-reduct is a standard MTL-algebra, no matter what the $\sim$-operation is. If $\boldsymbol{A}$ is a standard MTL-algebra and $\sim$ is an involutive negation, denote $\boldsymbol{A}^\sim$ the algebra that is an expansion of $\boldsymbol{A}$ with the involutive negation $\sim$. Further, denote $\mathbb{A}^\sim$ the class of all algebras that expand $\boldsymbol{A}$ with some involutive negation $\sim$.

Let A be a set totally ordered by $\leq$. Let $\langle a_1, b_1\rangle, \langle a_2, b_2\rangle, \ldots, \langle a_n, b_n\rangle$ be a finite number of pairs from A. We say that these pairs are nested (w.r.t. $\leq$) iff there is a permutation σ on $\{1, \ldots, n\}$ such that $a_{\sigma(1)} \leq a_{\sigma(2)} \leq \cdots \leq a_{\sigma(n)} \leq b_{\sigma(n)} \leq \cdots \leq a_{\sigma(2)} \leq a_{\sigma(1)}$ and for each $i = 1, \ldots, n-1$ we have $a_{\sigma(i)} = a_{\sigma(i+1)}$ iff $b_{\sigma(i)} = b_{\sigma(i+1)}$.

LEMMA 6.4.1. *Let $0 < a_0 < \cdots < a_k < 1$ be real numbers. Then there is a decreasing involution $\sim$ on $[0, 1]$ such that $\sim(a_i) = a_{k-i}$ for $i = 0, \ldots, k$.*

PROPOSITION 6.4.2. *If* L *is the logic of some standard* BL*-algebra, then* $\mathrm{L}_\sim$ *enjoys finite strong standard completeness.*

This entails that, if L is the logic of some standard BL-algebra, $\mathrm{L}_\sim$ is a conservative expansion of L.

THEOREM 6.4.3. *Let $\boldsymbol{A}$ be a standard* SBL*-algebra which is a finite ordinal sum. Then* $\mathrm{Th}_\forall(\mathbb{A}^\sim)$ *is* **coNP***-complete.*

Proof. Hardness follows from Lemma 3.1.2, considering that $\boldsymbol{A}$ is the $\sim$-free reduct of each $\boldsymbol{A}^\sim \in \mathbb{A}^\sim$. We prove **NP**-containment of $\mathrm{Th}_\exists(\mathbb{A}^\sim)$. Since $\boldsymbol{A}$ is a standard SBL-algebra, we modify ALGORITHM EX-FIN to cater for the involutive negation. We rely on Lemma 6.4.1.

ALGORITHM EX-INV `//accepts` $\mathrm{Th}_\exists(\mathbb{A}^\sim)$

`input:` Φ `// ex. sentence in the language of BL with` $\sim$

```
begin
normalForm()
componentDelimiters()
guessOrder()
checkOrder()
```

`checkInvolution()` Let k be the number of variables in the normal form of Φ. For each x_i, x_j, $i, j \leq k'$, if $x_i = \sim x_j$ occurs in Φ_2, put down a pair $\{x_i, x_j\}$. Do this for every occurrence of $\sim$ in *Eq*. Check that the pairs thus created are nested w.r.t. $\leq_0$.

```
checkExternal()
checkInternal()
end
```

We discuss why this works correctly: assume for an algebra $\boldsymbol{A}^\sim$ obtained from $\boldsymbol{A}$ by adding some involutive involution, we have $\models_{\boldsymbol{A}} \Phi$. Then one guess of $\leq_0$ will be the real ordering of $\boldsymbol{A}$ on all values of subformulas. The properties of the involution on **A** warrant that the conditions in the step `checkInvolution()` will be satisfied. (And the equations in the remaining operations will be solvable.) Hence, there is an accepting computation. On the other hand, if there is an accepting computation on Φ, then the values of all variables determine a complete evaluation of Φ; by Lemma 6.4.1, there is a decreasing involution on $[0, 1]$ which satisfies all the identities prescribed by φ on the given values. Note that the check of soundness of the ordering w.r.t. involution is independent of the other steps and can be performed at any stage (after the ordering $\leq_0$ is established). $\square$

The algorithm can be modified to work also for $\mathrm{SBL}_{\sim}$: instead of working with a fixed algebra, the algorithm first (transforms the input formula into a normal form, and then) guesses an ordinal sum whose first component is Π and whose number of components is bounded by $2m+1$, where m is the number of variables in the normal form (for each variable, we consider its component and the component of its $\sim$-negation).

Finally, for infinite ordinal sums, we may restrict our attention to the canonical ones; if $\boldsymbol{A}$ is any standard BL-algebra and $\boldsymbol{A}'$ is the canonical standard BL-algebra with the same universal (equational, quasiequational) theory, then by completeness, the logic of $\mathbb{A}^{\sim}$ coincides with the logic of $\mathbb{A}'^{\sim}$.

COROLLARY 6.4.4. *Let* L *be the logic of a standard* SBL*-algebra. Then* $\mathrm{THM}(\mathrm{L}_{\sim})$ *and* $\mathrm{CONS}(\mathrm{L}_{\sim})$ *are* $\mathbf{coNP}$*-complete.*

One might further ask about complexity of fragments of the theory of *particular* (standard) $\mathrm{SBL}_{\sim}$ algebras (or $\mathrm{SMTL}_{\sim}$-algebras); this is an interesting open problem. Research in this area will be framed by available results on the structure of the lattice of subvarieties of the variety of $\mathrm{SBL}_{\sim}$-algebras. Chapter VIII gives details. We mention an important example, the case of standard $\Pi_{\sim}$-algebras given by a combination of the standard product algebra $[0,1]_{\Pi}$ and an arbitrary involutive negation $\sim$ on $[0,1]$; such an algebra will be denoted $[0,1]_{\Pi}^{\sim}$.

For $[0,1]_{\Pi}$ and $\sim_1$, $\sim_2$ two involutive negations on $[0,1]$, we say $\sim_1$ and $\sim_2$ are *isomorphic* w.r.t. $[0,1]_{\Pi}$ iff there is an isomorphism of $[0,1]_{\Pi}^{\sim_1}$ onto $[0,1]_{\Pi}^{\sim_2}$. Any such isomorphism must obviously be an automorphism of the product t-norm. It is not difficult to see that the cardinality of the class of isomorphism classes of algebras $[0,1]_{\Pi}^{\sim}$, for all choices of $\sim$ on $[0,1]$, is that of the continuum. The following result can be obtained:

PROPOSITION 6.4.5 ([18]). *Let* $\sim_1$, $\sim_2$ *be two involutive negations on* $[0,1]$. *Then we have* $\mathrm{TAUT}([0,1]_{\Pi}^{\sim_1}) = \mathrm{TAUT}([0,1]_{\Pi}^{\sim_2})$ *iff* $[0,1]_{\Pi}^{\sim_1}$ *is isomorphic to* $[0,1]_{\Pi}^{\sim_2}$.

Hence, nice complexity results can only be obtained for a minority of such algebras. This result concerning the standard product algebra can be generalized to particular ordinal sums of Ł- and Π-components, as shown in [27]. In the next section, we give an upper bound for the particular case of $\sim$ being the function $1-x$.

6.5 The logics $\text{Ł}\Pi$ and $\text{Ł}\Pi\frac{1}{2}$

The logics $\text{Ł}\Pi$ and $\text{Ł}\Pi\frac{1}{2}$ are discussed in detail in Chapter VIII. The logic $\text{Ł}\Pi$ is a result of combining the Łukasiewicz and the product connectives in a single logical system. We use subscripts to distinguish between the two sets of connectives where necessary, writing $\&_{\text{Ł}}$, $\&_{\Pi}$, etc.; $\sim$ denotes the involutive negation in Ł. The logic $\text{Ł}\Pi\frac{1}{2}$ has, in addition, a constant $\frac{1}{2}$ in the language. All the connectives of the three important schematic extensions of BL are available in both $\text{Ł}\Pi$ and $\text{Ł}\Pi\frac{1}{2}$. Moreover, the $\triangle$ connective is defined by $\triangle\varphi$ being $\neg{\sim}\varphi$. Then the $\triangle$-deduction theorem entails polynomial equivalence of THM and CONS for both $\text{Ł}\Pi$ and $\text{Ł}\Pi\frac{1}{2}$ (cf. Lemma 3.1.1).

The logics $\text{Ł}\Pi$ and $\text{Ł}\Pi\frac{1}{2}$ can be presented as expansions of Łukasiewicz logic; another approach to axiomatizing $\text{Ł}\Pi$ is the following one. Take the logic $\Pi_{\sim}$ and define $\varphi \to_{\text{Ł}} \psi$ as ${\sim}(\varphi \,\&_{\Pi} {\sim}(\varphi \to_{\Pi} \psi))$. Add $(\varphi \to_{\text{Ł}} \psi) \to ((\psi \to_{\text{Ł}} \chi) \to (\varphi \to_{\text{Ł}} \chi))$ to the axioms of $\Pi_\sim$. Then one can define all the other connectives of ŁΠ, in particular, $\&_{Ł}$. For the logic $Ł\Pi\frac{1}{2}$, introduce a new constant $\frac{1}{2}$ and add the axiom $\frac{1}{2} \leftrightarrow_{Ł} \sim\frac{1}{2}$.

The standard semantics of ŁΠ and $Ł\Pi\frac{1}{2}$ on $[0,1]$ is obtained by combining the standard semantics for both sets of connectives: $[0,1]_{Ł\Pi} = \langle *_{Ł}, \rightarrow_{Ł}, *_{\Pi}, \rightarrow_{\Pi}, 0\rangle$. The standard algebra $[0,1]_{Ł\Pi\frac{1}{2}}$ has additionally a constant operation $\frac{1}{2}$. Completeness of ŁΠ with respect to $[0,1]_{Ł\Pi}$ and of $Ł\Pi\frac{1}{2}$ with respect to $[0,1]_{Ł\Pi\frac{1}{2}}$ can be proved; it follows that ŁΠ expands its Ł-, G-, and Π-fragment conservatively, and that $Ł\Pi\frac{1}{2}$ is a conservative expansion of ŁΠ. This also shows what has not been mentioned in the previous subsection, that the logic ŁΠ is a complete axiomatization of the standard $\Pi_\sim$-algebra given by the product t-norm and the involutive negation $1-x$.

Owing to the presence of an involutive negation, Lemma 3.2.1 holds in full, establishing usual reductions between tautologousness and satisfiability. Summing up, we only need to investigate TAUT and SAT for ŁΠ and $Ł\Pi\frac{1}{2}$.

LEMMA 6.5.1.

(i) $\mathrm{TAUT}([0,1]_{Ł\Pi}) \approx_{\mathbf{P}} \mathrm{TAUT}([0,1]_{Ł\Pi\frac{1}{2}})$.

(ii) $\mathrm{SAT}([0,1]_{Ł\Pi}) \approx_{\mathbf{P}} \mathrm{SAT}([0,1]_{Ł\Pi\frac{1}{2}})$.

Proof. (i) It is sufficient to show $\mathrm{TAUT}([0,1]_{Ł\Pi\frac{1}{2}}) \preceq_{\mathbf{P}} \mathrm{TAUT}([0,1]_{Ł\Pi})$, the other reduction follows from the fact that the latter set is the $\frac{1}{2}$-free fragment of the former one (cf. Lemma 3.1.2). Let φ be an $Ł\Pi\frac{1}{2}$-formula, p be a new variable; then

$$\varphi \in \mathrm{TAUT}([0,1]_{Ł\Pi\frac{1}{2}}) \text{ iff } \triangle(p \leftrightarrow_{Ł} \sim p) \rightarrow_{Ł} \varphi(\tfrac{1}{2}/p) \in \mathrm{TAUT}([0,1]_{Ł\Pi}).$$

Observe that $\varphi(\frac{1}{2}/p)$ is an ŁΠ-formula. Now assume $\varphi \in \mathrm{TAUT}([0,1]_{Ł\Pi\frac{1}{2}})$; for any evaluation e in $[0,1]_{Ł\Pi}$, either $e(p \leftrightarrow_{Ł} \sim p) = 1$, then $e(p) = 0.5$, and $e(\varphi(\frac{1}{2}/p)) = 1$ by assumption, or $e(p \leftrightarrow_{Ł} \sim p) < 1$, so $e(\triangle(p \leftrightarrow_{Ł} \sim p)) = 0$, and hence $e(\triangle(p \leftrightarrow_{Ł} \sim p) \rightarrow_{Ł} \varphi(\frac{1}{2}/p)) = 1$. Conversely, if $\triangle(p \leftrightarrow_{Ł} \sim p) \rightarrow_{Ł} \varphi(\frac{1}{2}/p) \in \mathrm{TAUT}([0,1]_{Ł\Pi})$, then in particular all evaluations e in $[0,1]_{Ł\Pi}$ such that $e(p) = 0.5$ give $e(\varphi(\frac{1}{2}/p)) = 1$, so $\varphi \in \mathrm{TAUT}([0,1]_{Ł\Pi\frac{1}{2}})$.

(ii) Similarly, for p new variable, $\varphi \in \mathrm{SAT}([0,1]_{Ł\Pi\frac{1}{2}})$ iff $(p \leftrightarrow_{Ł} \sim p) \,\&_{Ł}\, \varphi(\frac{1}{2}/p) \in \mathrm{SAT}([0,1]_{Ł\Pi})$. □

Tautologousness and satisfiability in $[0,1]_{Ł\Pi}$ and $[0,1]_{Ł\Pi\frac{1}{2}}$ can be shown to be in **PSPACE** using a polynomial reduction to $\mathrm{Th}_{\exists}(\mathrm{RCF})$. Since the TAUT problems for both algebras are polynomially equivalent (and the same goes for SAT), we henceforth work only with $[0,1]_{Ł\Pi\frac{1}{2}}$.

THEOREM 6.5.2. *$\mathrm{Th}_{\forall}([0,1]_{Ł\Pi\frac{1}{2}})$ and $\mathrm{Th}_{\exists}([0,1]_{Ł\Pi\frac{1}{2}})$ are in* **PSPACE**.

Proof. Both proofs are conducted by replacing atomic formulas in the language of $Ł\Pi\frac{1}{2}$-algebras with their equivalents in the ordered field of reals. We show that

$$\mathrm{Th}_{\exists}([0,1]_{Ł\Pi\frac{1}{2}}) \preceq_{\mathbf{P}} \mathrm{Th}_{\exists}(\mathrm{RCF}).$$

Let Φ be an existential sentence in the language of $Ł\Pi\frac{1}{2}$. Using Lemma 3.3.5, transform Φ into a formula Φ' of the form $\exists x_1 \ldots \exists x_n(\Phi_1 \wedge \Phi_2)$, where Φ_1 is a Boolean

combination of atomic formulas $x_i = x_j, x_i \leq x_j, x_i < x_j$ for some pairs $1 \leq i, j \leq n$, while Φ_2 is a conjunction of identities without compound terms.
Define a formula Φ'' as $\exists x_1 \dots \exists x_n(\Phi_1 \wedge \Phi_2' \wedge \Phi_3)$, where

- Φ_1 is as before,
- Φ_3 is $0 \leq x_i \leq 1$ for each $1 \leq i \leq n$ (boundary conditions), and
- Φ_2' results from Φ_2 by processing all identities in Φ_2 ($i, j, k \in \{1, \dots, n\}$) in the following way:
  - (i) keep any identity $x_i = c$, where c is a constant;
  - (ii) replace any $x_i *_{Ł} x_j = x_k$ with $(x_i + x_j - 1 \leq 0 \wedge x_k = 0) \vee (x_i + x_j - 1 > 0 \wedge x_k = x_i + x_j - 1)$;
  - (iii) replace any $x_i *_{\Pi} x_j = x_k$ with $x_i.x_j = x_k$;
  - (iv) replace any $x_i \rightarrow_{Ł} x_j = x_k$ with $(x_i \leq x_j \wedge x_k = 1) \vee (x_i > x_j \wedge 1 - x_i + x_j = x_k)$;
  - (v) replace any $x_i \rightarrow_{\Pi} x_j = x_k$ with $(x_i \leq x_j \wedge x_k = 1) \vee (x_i > x_j \wedge x_i * x_k = x_j)$.

Clearly Φ holds in $[0,1]_{Ł\Pi\frac{1}{2}}$ iff Φ'' holds in R; for replacement of Łukasiewicz connectives, see Lemma 4.1.1. Moreover, it is obvious that the translation process can be performed in time polynomial in $|\Phi|$. □

Interestingly, a converse reducibility also holds in a stronger way; namely, we are going to show next that $\mathrm{Th}_\forall(\mathrm{RCF}) \preceq_{\mathbf{P}} \mathrm{TAUT}([0,1]_{Ł\Pi\frac{1}{2}})$.

Let us start with defining a bijection f of $(0,1)$ onto R. Take

$$f_{\mathrm{neg}}(x) = \frac{4x}{2x-1} \quad f_{\mathrm{pos}}(x) = \frac{4-4x}{2x-1}$$

Then the inverse functions to $f_{\mathrm{neg}}, f_{\mathrm{pos}}$ are

$$f_{\mathrm{neg}}^{-1}(x) = \frac{x}{2x-4} \quad f_{\mathrm{pos}}^{-1}(x) = \frac{x+4}{2x+4}$$

The function f is defined from $f_{\mathrm{neg}}, f_{\mathrm{pos}}$ as follows:

$$f(x) = \begin{cases} f_{\mathrm{neg}}(x) & \text{if } 0 < x < \frac{1}{2} \\ 0 & \text{if } x = \frac{1}{2} \\ f_{\mathrm{pos}}(x) & \text{if } \frac{1}{2} < x < 1 \end{cases}$$

and the inverse function to f is

$$f^{-1}(x) = \begin{cases} f_{\mathrm{neg}}^{-1}(x) & \text{if } x < 0 \\ \frac{1}{2} & \text{if } x = 0 \\ f_{\mathrm{pos}}^{-1}(x) & \text{if } x > 0 \end{cases}$$

Using f, define an isomorphic copy of the ordered field of reals on $(0,1)$: let $\mathsf{R}^0 = \langle(0,1), +^0, \cdot^0, 0^0, 1^0\rangle$, where for $x, y \in (0,1)$ the operations of R^0 are as follows:

$$\begin{aligned} x +^0 y &= f^{-1}(f(x) + f(y)) \\ x \cdot^0 y &= f^{-1}(f(x) \cdot f(y)) \\ 0^0 &= \tfrac{1}{2} \\ 1^0 &= \tfrac{5}{6} \end{aligned}$$

and $x \leq^0 y$ is strictly order-reversing on $(0, \frac{1}{2})$ and on $(\frac{1}{2}, 1)$ and order-preserving otherwise.

As R^0 is an isomorphic copy of the ordered field of reals, their theories coincide. We now define a function that assigns to each open RCF-formula Φ a term in the language of $[0,1]_{\text{ŁΠ}\frac{1}{2}}$. The function is defined by induction on formula structure.

A first observation to be made is that comparisons $=$, $\leq$ and $<$ on $[0,1]$ can be expressed by equations in the language of ŁΠ$\frac{1}{2}$-algebras. In particular, $x = y$ corresponds to $\triangle(x \leftrightarrow_{Ł} y) = 1$ (denote $t^{=}(x,y)$ the term on the left-hand side), $x \leq y$ corresponds to $\triangle(x \to_{Ł} y) = 1$ and $x < y$ corresponds to $\triangle(x \to_{Ł} y) \,\&_{Ł}\, {\sim}\triangle(x \leftrightarrow_{Ł} y) = 1$.

Recall that for $x, y \in (0,1)$ we have $x \leq^0 y$ iff either $x \leq \frac{1}{2} \leq y$ or $y \leq x$ and either $x, y < \frac{1}{2}$ or $\frac{1}{2} < x, y$. So $x \leq^0 y$ on $(0,1)$ can be expressed as $t^{\leq^0}(x,y) = 1$, where $t^{\leq^0}(x,y)$ is

$$\triangle(x \vee {\sim}y \to_{Ł} \tfrac{1}{2}) \vee (\triangle(y \to_{Ł} x) \wedge ({\sim}\triangle(\tfrac{1}{2} \to_{Ł} x \vee y) \vee {\sim}\triangle(x \wedge y \to_{Ł} \tfrac{1}{2})))$$

Suppose Φ is an open RCF-formula. We will need to assume that Φ is without compound terms; Subsection 3.3 shows how to polynomially eliminate compound terms from a given formula (the statement there is given for existential sentences; apply it to a negation of the universal closure of Φ).

LEMMA 6.5.3. *The functions* 0^0, 1^0, $+^0$, $\cdot^0$ *are term-definable in* $[0,1]_{\text{ŁΠ}\frac{1}{2}}$.

Proof. Clearly $0^\star$ is $\frac{1}{2}$ and $1^\star$ is $\frac{5}{6}$.[20] For $+^0$, $\cdot^0$, it is enough to recall that these functions are piecewise rational; their definability in $[0,1]_{\text{ŁΠ}\frac{1}{2}}$ follows. □

So for each term t in the language of RCF, we have a defining term $t_\star$ in $[0,1]_{\text{ŁΠ}\frac{1}{2}}$; if Φ has no compound terms, then in particular for each term t in Φ, $|t_\star|$ is polynomial in $|t|$. Now assume Φ is an open RCF-formula without compound terms. We define t^Φ as follows:

- if Φ is $s = u$, then t^Φ is $t^{=}(s_\star, u_\star)$;
- if Φ is $s \leq u$, then t^Φ is $t^{\leq^0}(s_\star, u_\star)$;
- if Φ is $\neg\Theta$, then t^Φ is ${\sim}t^\Theta$;
- if Φ is $\Theta \wedge \Psi$, then t^Φ is $t^\Theta \wedge t^\Psi$.

LEMMA 6.5.4. *Let* Φ *be an open* RCF*-formula. Then* Φ *holds in the ordered field of reals iff* $t^\Phi = 1$ *holds in* $[0,1]_{\text{ŁΠ}\frac{1}{2}}$.

[20] See the proof of Lemma 6.5.7 for the definition of rationals in $[0,1]_{\text{ŁΠ}\frac{1}{2}}$.

We have shown:

THEOREM 6.5.5. $\mathrm{Th}_\forall(\mathsf{R}) \preceq_{\mathbf{P}} \mathrm{THM}([0,1]_{Ł\Pi\frac{1}{2}})$.

Combining the results, we get

THEOREM 6.5.6. $\mathrm{THM}(Ł\Pi\frac{1}{2}) \approx_{\mathbf{P}} \mathrm{Th}_\forall(\mathsf{R})$.

We close this subsection by pointing out how logics with rational constants can be interpreted in the logic $Ł\Pi\frac{1}{2}$ (and hence also $Ł\Pi$).

LEMMA 6.5.7. $\mathsf{Q} \cap [0,1]$ *is polynomially term-definable in* $[0,1]_{Ł\Pi\frac{1}{2}}$.

Proof. For each $q = \frac{k}{l} \in \mathsf{Q} \cap [0,1]$ we seek to find an $Ł\Pi\frac{1}{2}$-expression φ such that for each evaluation e in $[0,1]_{Ł\Pi\frac{1}{2}}$ we have $e(\varphi) = q$. Moreover, we demand that $|\varphi|$ be polynomial in $|q| \leq 2\log(l)$.

Choose $n \in \mathsf{N}$ the least such that $l < 2^n$; then $2^{n-1} \leq l < 2^n \leq 2l$, therefore $\log(l) < n \leq \log(l) + 1$ and $n \in O(\log(l))$.

First, for each $n \in \mathsf{N}$: $\frac{1}{2^n} = \frac{1}{2} *_\Pi \cdots *_\Pi \frac{1}{2}$ (n times); the number of factors is n.

To define $\frac{k}{2^n}$ for $k > 1$, we cannot use $\frac{k}{2^n} = \frac{1}{2^n} \oplus \cdots \oplus \frac{1}{2^n}$ (k times), because then the cardinality of the sum would be the value of k (which is exponential in $|k|$). Instead, consider that $k = \sum_{i \leq \lfloor \log(k) \rfloor} c_i . 2^i$, where $(c_{\lfloor \log(k) \rfloor} \ldots c_0)$ is the binary representation of k. For $k \leq 2^n$, define $\frac{k}{2^n} = \bigoplus_{i \leq \lfloor \log(k) \rfloor} (c_i *_\Pi \frac{1}{2^{n-i}})$. Here, the cardinality of the sum is $|k|$ and each summand is in $O(n)$.

Finally, put $\frac{k}{l} = \frac{k}{2^n} \rightarrow_\Pi \frac{l}{2^n}$. □

The following corollary entails Theorem 6.3.6 concerning the logic $\Pi(\mathcal{Q})$; however, for $Ł(\mathcal{Q})$ and $\mathrm{G}(\mathcal{Q})$, we were able to obtain a better upper bound in Subsection 6.3 than the one implied by the following corollary.

COROLLARY 6.5.8. *Let* L *be one of* Ł, G, Π. *Then* $\mathrm{THM}(\mathrm{L}(\mathcal{Q})) \preceq_{\mathbf{P}} \mathrm{THM}(Ł\Pi\frac{1}{2})$.

7 MTL and its axiomatic extensions

Despite the tumultuous research in the family of MTL and its extensions, complexity results leave much to be desired for those logics that do not also extend BL. It follows from [5] that MTL and its axiomatic extensions SMTL and IMTL are decidable since the corresponding varieties of algebras enjoy the finite embeddability property (FEP); in fact this implies that the universal theories of the respective varieties of algebras are decidable. Hence, due to (strong) completeness of MTL (IMTL, SMTL) w.r.t. the class of MTL-algebras (IMTL-algebras, SMTL-algebras respectively), both theorems and provability from finite theories for each of the three logics are decidable. Apart from that, theoremhood and provability in axiomatic extension ΠMTL of MTL is decidable (though ΠMTL-algebras do not have the FEP), as shown in [28].

The problem with improving this upper bound seems to be our insufficient knowledge of the structure of MTL-algebras (or even standard MTL-algebras). However, for particular (classes of) MTL-algebras, whose structure is known, usual complexity results can be obtained, as shown below.

Within this section, we work with the language $\{\&, \rightarrow, \wedge, \overline{0}\}$.

7.1 (Weak) nilpotent minimum logic

MTL was introduced in [12], along with its axiomatic extensions WNM (weak nilpotent minimum) and NM (nilpotent minimum). WNM is an axiomatic extension of MTL with the axiom

$$(\varphi \mathbin{\&} \psi \to \overline{0}) \vee (\varphi \wedge \psi \to \varphi \mathbin{\&} \psi)$$

and NM is the involutive extension of WNM, obtained by adding to WNM the axiom

$$\neg\neg\varphi \leftrightarrow \varphi.$$

Let us look at the semantics of these logics. A unary function $n\colon [0,1] \to [0,1]$ is a weak negation iff it is order-reversing, $n(0) = 1$, $n(1) = 0$, and $x \leq n(n(x))$ for all $x \in [0,1]$). Given a weak negation n, one defines

$$x *_n y = \begin{cases} 0 & \text{if } x \leq n(y) \\ \min\{x, y\} & \text{otherwise.} \end{cases}$$

Then $*_n$ is a left-continuous t-norm and the (MTL-)algebra $[0,1]_{*_n}$ is a standard WNM-algebra. If one starts with the function $1 - x$ in the role of n, the resulting t-norm is

$$x *_{\mathrm{NM}} y = \begin{cases} 0 & \text{if } x + y \leq 1 \\ \min\{x, y\} & \text{otherwise} \end{cases}$$

and the corresponding standard algebra is denoted $[0,1]_{\mathrm{NM}}$. If one starts from a different involutive negation, then the whole algebra determined by it is isomorphic to $[0,1]_{\mathrm{NM}}$. The logic NM is strongly complete w.r.t. the algebra $[0,1]_{\mathrm{NM}}$.

Also the logic WNM enjoys strong completeness w.r.t. the above standard algebras, but this class is not tangible enough for our purpose. It turns out that we can prove completeness with respect to a narrower class.

For $k \in \mathsf{N} \setminus \{0\}$, define $I_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$. Let $S \subseteq I_k$ be arbitrary such that $\frac{i}{k} \in S$ iff $\frac{k-i-1}{k} \in S$ for $i \leq \frac{k}{2}$. Define on $[0,1]$ the following function n_k^S:

$$n_k^S(x) = \begin{cases} 1 - x & \text{if } x \in [\frac{i}{k}, \frac{i+1}{k}] \text{ for } \frac{i}{k} \in S \\ \frac{k-i-1}{k} & \text{otherwise.} \end{cases}$$

It is easy to check that n_k^S is a weak negation, which determines a corresponding standard WNM-algebra $[0,1]_{\mathrm{WNM}_k^S}$.

PROPOSITION 7.1.1 ([12]). *The class of* WNM*-chains is partially embeddable into the class of standard* WNM*-algebras* $[0,1]_{\mathrm{WNM}_k^S}$. *In particular, each n-element partial subalgebra of a* WNM*-chain is embeddable into a* $[0,1]_{\mathrm{WNM}_{n'}^S}$ *for some $n' \leq 2n + 2$ and some choice of S.*

COROLLARY 7.1.2. *The variety* $\mathbb{WNM}$ *of* WNM*-algebras is generated by the class of algebras* $[0,1]_{\mathrm{WNM}_k^S}$ *(where $k \in \mathsf{N} \setminus 0$, S is an arbitrary subset of* $\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$*) as a quasivariety, and the logic* WNM *enjoys finite strong standard completeness with respect to this class of algebras.*

We now explore complexity of the two logics; the results first appeared in [32].

THEOREM 7.1.3. $\mathrm{Th}_\forall([0,1]_{\mathrm{NM}})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from Theorem 3.4.1. We show $\mathbf{NP}$-containment of the set $\mathrm{Th}_\exists([0,1]_{\mathrm{NM}})$.[21]

For any term $t(x_1,\ldots,x_n)$ and for any evaluation e in $[0,1]_{\mathrm{NM}}$, the value $e(t')$ for any subterm $t' \preceq t$ will be among

$$V = \{0, e(x_1), \ldots, e(x_n), \neg e(x_1), \ldots, \neg e(x_n), 1\}$$

and that operations on V are order-determined, i.e., the value $e(t')$ is fully determined by the ordering of V. Note that the negation $\neg$ is involutive, and so $e(x) < e(y)$ iff $\neg e(y) < \neg e(x)$.

It follows that, given an existential sentence in the language of MTL, we may replace the existential quantification over all evaluations by an existential quantification over all orderings of variables occurring in the input formula and of their negations (with respect to the bottom and top elements of the algebra). The ordering must respect the involutive negation, i.e., the pairs $\langle e(x_1), \neg e(x_1)\rangle$, $\langle e(x_2), \neg e(x_2)\rangle$, …, $\langle e(x_n), \neg e(x_n)\rangle$ must be nested.

We describe a nondeterministic ALGORITHM EX-NM which accepts existential sentences valid in $[0,1]_{\mathrm{NM}}$. An existential sentence $\exists x_1 \ldots \exists x_n \Phi(x_1,\ldots,x_n)$ is given, where Φ is a Boolean combination of identities. Guess a linear ordering $\leq_0$ of the set $\{0, x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n, 1\}$, such that $0 \leq_0 x_i \leq_0 1$ for $1 \leq i \leq n$, and in such a way that the pairs induced by $\neg$ are nested. This is clearly polynomial in the input size. Then compute the value of each identity in Φ: on the basis of $\leq_0$, evaluate all terms and subsequently also all atomic formulas. Then compute the value of the Boolean combination in Φ, accept iff this value is 1. □

Using finite strong standard completeness theorem for NM, proved in [12], we get

COROLLARY 7.1.4.

- *(i)* $\mathrm{SAT}_{(\mathrm{pos})}([0,1]_{\mathrm{NM}})$ *is* $\mathbf{NP}$*-complete.*
- *(ii)* $\mathrm{TAUT}_{(\mathrm{pos})}([0,1]_{\mathrm{NM}})$ *and* $\mathrm{CONS}([0,1]_{\mathrm{NM}})$ *are* $\mathbf{coNP}$*-complete.*
- *(iii)* $\mathrm{THM}(\mathrm{NM})$ *and* $\mathrm{CONS}(\mathrm{NM})$ *are* $\mathbf{coNP}$*-complete.*

THEOREM 7.1.5. *For each choice of k and S,* $\mathrm{Th}_\forall([0,1]_{\mathrm{WNM}_k^S})$ *is* $\mathbf{coNP}$*-complete.*

Proof. Let k and S be fixed and let $[0,1]_{\mathrm{WNM}_k^S}$ be the standard WNM-algebra given by these parameters. Hardness follows from Theorem 3.4.1. We show $\mathbf{NP}$-containment of $\mathrm{Th}_\exists([0,1]_{\mathrm{WNM}_k^S})$, in a manner similar to the proof of Theorem 7.1.3.

Clearly for any term $t(x_1,\ldots,x_n)$ and for any evaluation e in $[0,1]_{\mathrm{WNM}_k^S}$, the value $e(t')$ for any subterm $t' \preceq t$ will be among

$$V = \{0, e(x_1), \ldots, e(x_n), \neg e(x_1), \ldots, \neg e(x_n), \neg\neg e(x_1), \ldots, \neg\neg e(x_n), 1\}$$

and operations on V are order-determined, i.e., the value $e(t')$ is fully determined by the ordering of V. Therefore again, we take an existential quantification over particular orderings.

[21] Cf. also the proof of Theorem 5.1.1.

We describe a nondeterministic ALGORITHM EX-WNM which accepts existential sentences valid in $[0,1]_{\mathrm{WNM}_k^S}$. Let an existential sentence $\exists x_1 \ldots \exists x_n \Phi(x_1,\ldots,x_n)$ be given, where Φ is a Boolean combination of identities. Guess a linear ordering $\leq_0$ of a set consisting of 0, 1 and terms x_i, $\neg x_i$ and $\neg\neg x_i$ for $1 \leq i \leq n$ such that $0 \leq_0 x_i \leq_0 1$ for $1 \leq i \leq n$, such that the pairs $\langle \neg x_1, \{x_1, \neg\neg x_1\}\rangle$, $\langle \neg x_2, \{x_2, \neg\neg x_2\}\rangle$, $\ldots$, $\langle \neg x_n, \{x_n, \neg\neg x_n\}\rangle$ are nested, satisfy $x_i \leq \neg\neg x_i$ and conform to functionality of $\neg$, namely, if for any expressions a, b we have $a =_0 b$, then $\neg a =_0 \neg b$. This can be done in time polynomial in $|\Phi|$. Then compute the value of each atomic formula in Φ: on the basis of $\leq_0$, evaluate all terms and subsequently also all atomic formulas. Then compute the value of the Boolean formula Φ, accept iff the value is 1. □

THEOREM 7.1.6.

(i) $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{WNM})$ *is* $\mathbf{NP}$*-complete.*

(ii) $\mathrm{TAUT}_{(\mathrm{pos})}(\mathbb{WNM})$ *and* $\mathrm{CONS}(\mathbb{WNM})$ *are* $\mathbf{coNP}$*-complete.*

(iii) $\mathrm{THM}(\mathrm{WNM})$ *and* $\mathrm{CONS}(\mathrm{WNM})$ *are* $\mathbf{coNP}$*-complete.*

Proof. Hardness follows from Theorem 3.4.1. Due to Proposition 7.1.1, we may limit ourselves to algebras $[0,1]_{\mathrm{WNM}_k^S}$ for k bounded polynomially by the length of the input, as follows. (i) Given a formula φ with n variables as an instance of $\mathrm{SAT}_{(\mathrm{pos})}(\mathbb{WNM})$, guess a $k \leq 2n+2$ and ask whether $\varphi \in \mathrm{SAT}_{(\mathrm{pos})}([0,1]_{\mathrm{WNM}_k^S})$; the latter is in $\mathbf{NP}$. This algorithm solves the problem correctly: If a formula φ is (positively) satisfiable in any WNM-algebra, then it is also (positively) satisfiable in some WNM-chain. The evaluation of subterms forms a partial subalgebra, which is embeddable into an algebra $[0,1]_{\mathrm{WNM}_k^S}$ for some $k \leq 2n+2$ and some choice of S by Proposition 7.1.1. Hence $\varphi \in \mathrm{SAT}_{(\mathrm{pos})}(\mathbb{WNM})$ iff there are k and S (both polynomial in $|\varphi|$) such that $\varphi \in \mathrm{SAT}_{(\mathrm{pos})}([0,1]_{\mathrm{WNM}_k^S})$. (ii) If a formula is not a tautology, then this shows in some WNM-chain. The rest is as in (i). The case for CONS is analogous. □

We remark that one can obtain these complexity bounds also for tautologies of each of $[0,1]_{\mathrm{NM}}$ and of $[0,1]_{\mathrm{WNM}_k^s}$ for $k \in \mathsf{N} \setminus 0$ and a choice of S enriched with constants for Q; hence, e.g., the logic $\mathrm{NM}(\mathcal{Q})$ (expansion of NM with rational constants, axiomatized by adding bookkeeping axioms as valid in the standard NM-algebra) is $\mathbf{coNP}$-complete. Details can be found in [15].

8 Overview of results and open problems

Table 1 gives a summary of results obtained for logics (i.e., theoremhood and provability from finite theories); it omits results obtained for the corresponding classes of algebras (i.e., satisfiability, tautologousness, and finite consequence relation), both for spatial reasons and because these results do not add much new information.

The presentation is rather condensed, therefore it merits some explanation. The rows of the table represent logics considered in this chapter. For each of these, the column entries specify the complexity result for theoremhood/provability (these are, for all cases, identical). The '–' character means the logic has not been considered within this chapter. However, one can still use Lemma 3.1.2 to obtain bounds on complexity of fragments/expansions.

Logic L	THM(L)	CONS(L)	0-free fragment	$\triangle$ expansion	Q expansion	$\sim$ expansion
BL	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.	–	–
SBL	**coNP**-c.	**coNP**-c.	**coNP**-c	**coNP**-c.	–	**coNP**-c.
Ł	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.	–
L $\supset$Ł	**coNP**-c.	**coNP**-c.	–	–	–	–
G	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.
Π	**coNP**-c.	**coNP**-c.	**coNP**-c.	**coNP**-c.	$\in$ **PSPACE**	**coNP**-c.
L$(*) \supset$ BL	**coNP**-c.	**coNP**-c.	–	**coNP**-c.	–	**coNP**-c.
Ł$\Pi\frac{1}{2}$	$\in$ **PSPACE**	$\in$ **PSPACE**	–	–	–	–
MTL	decidable	decidable	decidable	–	–	–
IMTL	decidable	decidable	decidable	–	–	–
SMTL	decidable	decidable	decidable	–	–	–
ΠMTL	decidable	decidable	–	–	–	–
NM	**coNP**-c.	**coNP**-c.	–	–	**coNP**-c.	–
WNM	**coNP**-c.	**coNP**-c.	–	–	–	–

Table 1. Complexity of Propositional Logics

Open problems and directions. As already observed, BL and its extensions have received quite a thorough treatment as to basic questions of computational complexity of theoremhood and provability in the propositional case. (Chapter XI addresses arithmetical complexity of first-order calculi.) From these basic results, roughly speaking one can move in several directions.

The greatest itch, no doubt, is the lack of complexity results for MTL and its extensions (that are not at the same time extensions of BL). The same is true for semilinear extensions of other substructural logics introduced in this book, some of whom are not known to be decidable; this is the case of Uninorm Logic UL. Another track of research can be taken by modifying the language. This has been done in a degree, but the picture is still very incomplete both ways—shifting to fragments of the basic language, or to its expansions, or both. Moreover, one can pass from tautologousness and provability to more intricate problems involving propositional or first-order formulas. This includes admissible rules, quantified propositional formulas, etc. One can also look at (fragments of) algebraic theories for classes of algebras among the equivalent algebraic semantics for the logics. This is what has been, to a degree, presented in this chapter, but our approach has been a utilitarian one, while the topic is of independent interest.

9 Historical remarks and further reading

Earlier chapters of this book present fuzzy logics within the broader framework of substructural logics. In particular, the concept of *semilinearity* is introduced as an essential trait of fuzzy logics, and it is shown how some prototypical fuzzy logics can be obtained as semilinear extensions of well-known substructural logics, or as axiomatic extensions thereof. To exemplify, MTL is the semilinear extension of the logic $\mathrm{FL}_{\mathrm{ew}}$ and G is the semilinear extension of intuitionistic logic INT.

Semilinearity put aside, there is a substantial amount of complexity results on substructural logics (out of the scope of this chapter). Remarkably, the full Lambek calculus FL, and logics obtained by adding the structural rules of exchange, weakening and contraction to FL, possess streamlined Gentzen-style calculi, some of whom admit proofsearch in $\mathbf{PSPACE}$ (due to the *subformula property:* each formula occurring in the proof of φ is a subformula of φ). This is the case of FL itself, as well as $\mathrm{FL}_{\mathrm{ew}}$; for both these logics, theoremhood is in fact $\mathbf{PSPACE}$-complete. Unfortunately, the virtue of polynomial-space proof search seems to be lost when semilinearity is assumed. Instead, we have relied on completeness with respect to a suitable class of algebras and on sufficient understanding of the structure of these algebras. It has hopefully been demonstrated why, for many logics discussed in this chapter, one can work with the universal fragment of the full algebraic theory of standard algebras to obtain the same complexity result as for the equational, or the quasiequational, fragment. We stress that in a general case, provability from finite theories may be computationally much harder than theoremhood. For example, provability from finite theories in FL is undecidable.

Interestingly, the paper [6] also shows $\mathbf{PSPACE}$-containment for theoremhood and provability in the logic $\mathrm{GBL}_{\mathrm{ewf}}$. This logic is obtained from $\mathrm{FL}_{\mathrm{ew}}$ by adding the divisibility axiom; adding semilinearity to $\mathrm{GBL}_{\mathrm{ewf}}$, one obtains BL. It is shown in the paper that provability in $\mathrm{GBL}_{\mathrm{ewf}}$ is $\mathbf{PSPACE}$-hard.

Finally, let us mention a classic result of R. Statman [41]: theoremhood in intuitionistic logic INT (which is exactly $\mathrm{FL_{ewc}}$) is **PSPACE**-complete (and, because this logic enjoys the deduction theorem, so is provability).

We now turn to the story of unravelling computational complexity of propositional fuzzy logics presented in this chapter. It begins with D. Mundici's result establishing **NP**-completeness of satisfiability of propositional formulas in the standard Lukasiewicz algebra, presented in [37] in 1987. Mundici's method was proving an upper bound by reasoning about the functions represented by formulas. Our method is a rather straightforward reduction to the INEQ problem; however, we try to recover the geometry behind the small-model theorem that gives upper bounds for both LP and INEQ. Thus our approach rather resembles the proof presented in [20], where a reduction to the bounded version of MIP problem is used. Mundici's pioneering work laid a basis for results in other logics given by continuous t-norms, to be obtained much later; at the time of Mundici's result, mathematical fuzzy logic, as a homogeneous branch of formal nonclassical logics, had yet to be developed.

It was not until almost ten years later that the fundamentals of Hájek's basic logic were laid, which was the starting point of a focused and tumultuous research involving also computational complexity issues. Theoremhood was addressed via tautologousness in the corresponding (class of) standard algebras indicated by completeness theorems. In our presentation, we use the ideas of the proofs in the respective papers to obtain a more general result, shifting from identities in the algebraic theory to quasiidentities and universal theories.

In 1998, the paper [2] determined the complexity of theoremhood in product logic, showing it to be **coNP**-complete using a reduction to Łukasiewicz logic. This paper also mentions Gödel logic, but disclaims the result as common knowledge. As to the fact that (positively) satisfiable formulas and positive tautologies in the two standard algebras coincide with classical satisfiability and tautologousness respectively, this step was taken in [21]. The last reference gives a good overview of complexity results for Łukasiewicz, Gödel, and product logics.

In 2002, the paper [3] settled the complexity of theoremhood for propositional BL, relying on its standard completeness, Mostert–Shields theorem, and a simple but crucial observation on the number of components needed to represent potential counterexamples to tautologousness (incarnated here as Lemma 5.2.3). Later in the same year, a similar method was adapted to work also for any propositional logic given by a standard BL-algebra, in the paper [25]. The latter result, which implies there are only countably many subvarieties of $\mathbb{BL}$ given by a single standard BL-algebra, was anticipated in the strong result of [1] which characterizes those standard BL-algebras that generate the full variety $\mathbb{BL}$.

Another complexity result in the family of BL and its extensions concerned axiomatic extensions of Łukasiewicz logic and was presented in [8]; it relies on Komori's characterization of subvarieties of $\mathbb{MV}$, presented in [31]. The paper [30] also belongs to this family, showing that admissibility of rules in Łukasiewicz logic is in **PSPACE**.

For logics in expanded or restricted languages, results are rather fragmentary and some are negative. The computational complexity results for falsehood-free logics, as presented here, are covered by the comprehensive paper [13]. For involutive negations added as an independent new connective, the result [26] shows that theoremhood and provability in a logic given by some standard SBL-algebra and the class of all involutive negations is **coNP**-complete. However, this result says nothing about individual combinations of standard algebras and involutive negations, most notably, it says nothing about $[0,1]_{Ł\Pi}$. As to the logics $Ł\Pi$ and $Ł\Pi\frac{1}{2}$, containment in **PSPACE** for theoremhood was shown in [23] using the universal/existential fragment of the RCF-theory; the latter was shown to be in **PSPACE** in [7]. The result of polynomial equivalence of theorems of $Ł\Pi\frac{1}{2}$ and the universal fragment of the RCF theory comes from [33]. Intriguingly, these problems are apparently not known to be **PSPACE**-complete. Computational complexity for logics with rational constants has been addressed in [22]. In fact, the scope of the paper is slightly broader than presented here, dealing also with finite sums of particular properties. Moreover, rational expansions of standard WNM-algebras have been studied in [15].

For MTL and its extensions (expansions), results are mostly limited to decidability, obtained as in [5] for MTL and its axiomatic extensions SMTL and IMTL via finite embeddability property, and in [28] for ΠMTL. Results given here on NM and WNM are from [32].

We also mention Abelian logic, the logic of Abelian ℓ-groups (see [35]). This logic not only belongs to the family of semilinear substructural logics, but relates both to Łukasiewicz logic (see [34]) and to product logic (because of its semantics). This logic proves exactly all expressions φ in the ℓ-group language for which $\varphi \geq 0$ holds in all ℓ-groups, or equivalently, in Z. Using a variant of the small-model theorem for integers, as given in Section 2.5, one can show that the universal theory of the additive group on Z (and hence, due to a classic result of Y.S. Gurevich and A.I. Kokorin, of all Abelian o-groups) is in **coNP**. V. Weispfenning ([42]) extended this result to ℓ-groups by extending the small-model theorem to incorporate also the width of the lattice ordering: if a universal formula Φ is not valid in the class of all ℓ-groups, then this can be shown in an ℓ-group with of width polynomially bounded by $|\Phi|$. Hence, universal theory of Abelian ℓ-groups is **coNP**-complete.

To complement the account, we relied on [38] as a reference text on computational complexity.

Acknowledgements

The preparation of this text was supported by grants ICC/08/E018 (a part of ESF Eurocores-LogICCC project FP006) and P202/10/1826 of the Czech Science Foundation and by Institutional Research Plan AV0Z10300504. The author is indebted to her reviewers, Brunella Gerla and Emil Jeřábek, for reading and commenting upon a draft of the text and suggesting many improvements.

BIBLIOGRAPHY

[1] Paolo Aglianò and Franco Montagna. Varieties of BL-algebras I: General properties. *Journal of Pure and Applied Algebra*, 181(2–3):105–129, 2003.

[2] Matthias Baaz, Petr Hájek, Jan Krajíček, and David Švejda. Embedding logics into product logic. *Studia Logica*, 61(1):35–47, 1998.

[3] Matthias Baaz, Petr Hájek, Franco Montagna, and Helmut Veith. Complexity of t-tautologies. *Annals of Pure and Applied Logic*, 113(1–3):3–11, 2002.

[4] Willem J. Blok and Don L. Pigozzi. *Algebraizable Logics*, volume 396 of *Memoirs of the American Mathematical Society*. American Mathematical Society, Providence, RI, 1989. Freely downloadable from `http://orion.math.iastate.edu/dpigozzi/`.

[5] Willem J. Blok and Clint J. van Alten. The finite embeddability property for residuated lattices, pocrims and BCK-algebras. *Algebra Universalis*, 48(3):253–271, 2002.

[6] Simone Bova and Franco Montagna. The consequence relation in the logic of commutative GBL-algebras is PSPACE-complete. *Theoretical Computer Science*, 410(12–13):1143–1158, 2009.

[7] John F. Canny. Some algebraic and geometric computations in PSPACE. In *ACM Symposium on Theory of Computing*, pages 460–467, 1988.

[8] Petr Cintula and Petr Hájek. Complexity issues in axiomatic extensions of Łukasiewicz logic. *Journal of Logic and Computation*, 19(2):245–260, 2009.

[9] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[10] Antonio Di Nola and Ada Lettieri. Equational characterization of all varieties of MV-algebras. *Journal of Algebra*, 221(2):463–474, 1999.

[11] Francesc Esteva, Joan Gispert, Lluís Godo, and Carles Noguera. Adding truth-constants to logics of continuous t-norms: Axiomatization and completeness results. *Fuzzy Sets and Systems*, 158(6):597–618, 2007.

[12] Francesc Esteva and Lluís Godo. Monoidal t-norm based logic: Towards a logic for left-continuous t-norms. *Fuzzy Sets and Systems*, 124(3):271–288, 2001.

[13] Francesc Esteva, Lluís Godo, Petr Hájek, and Franco Montagna. Hoops and fuzzy logic. *Journal of Logic and Computation*, 13(4):532–555, 2003.

[14] Francesc Esteva, Lluís Godo, and Franco Montagna. Equational characterization of subvarieties of BL-algebras generated by t-norm algebras. *Studia Logica*, 76(2):161–200, 2004.

[15] Francesc Esteva, Lluís Godo, and Carles Noguera. On expansions of WNM t-norm based logics with truth-constants. *Fuzzy Sets and Systems*, 161(3):347–368, 2010.

[16] Ronald Fagin, Joseph Y. Halpern, and Nimrod Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1–2):78–128, 1990.

[17] Nikolaos Galatos. Equational bases for joins of residuated-lattice varieties. *Studia Logica*, 76(2): 227–240, 2004.

[18] Mai Gehrke, Carol Walker, and Elbert Walker. Fuzzy logics arising from strict de Morgan systems. In Steve Rodabaugh and Erich Peter Klement, editors, *Topological and Algebraic Structures in Fuzzy Sets: A Handbook of Recent Developments in the Mathematics of Fuzzy Sets*, pages 257–276. Kluwer, Dordrecht, 2003.

[19] Joan Gispert and Daniele Mundici. MV-algebras: A variety for magnitudes with archimedean units. *Algebra Universalis*, 53(1):7–43, 2005.

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

[21] Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer, Dordrecht, 1998.

[22] Petr Hájek. Computational complexity of t-norm based propositional fuzzy logics with rational truth constants. *Fuzzy Sets and Systems*, 157(5):677–682, 2006.

[23] Petr Hájek and Sauro Tulipani. Complexity of fuzzy probability logics. *Fundamenta Informaticae*, 45(3):207–213, 2001.

[24] Zuzana Haniková. Varieties generated by standard BL-algebras. Submitted.

[25] Zuzana Haniková. A note on the complexity of propositional tautologies of individual t-algebras. *Neural Network World*, 12(5):453–460, 2002. Special issue on SOFSEM 2002.

[26] Zuzana Haniková. On the complexity of propositional logic with an involutive negation. In M. Wagenknecht and R. Hampel, editors, *Proceedings of the 3rd EUSFLAT Conference*, pages 636–639, Zittau, 2003.

[27] Zuzana Haniková and Petr Savický. Distinguishing standard SBL-algebras with involutive negations by propositional formulas. *Mathematical Logic Quarterly*, 54(6):579–596, 2008.

[28] Rostislav Horčík. Decidability of cancellative extension of monoidal t-norm based logic. *Logic Journal of the Interest Group of Pure and Applied Logic*, 14(6):827–843, 2006.

[29] Rostislav Horčík and Kazushige Terui. Disjunction property and complexity of substructural logics. *Theoretical Computer Science*, 412(31):3992–4006, 2011.

[30] Emil Jeřábek. Admissible rules of Łukasiewicz logic. *Journal of Logic and Computation*, 20(2):425–447, 2010.

[31] Yuichi Komori. Super-Łukasiewicz propositional logics. *Nagoya Mathematical Journal*, 84:119–133, 1981.
[32] Enrico Marchioni. On computational complexity of semilinear varieties. *Journal of Logic and Computation*, 18(6):941–958, 2008.
[33] Enrico Marchioni and Franco Montagna. Complexity and definability issues in $\text{Ł}\Pi\frac{1}{2}$. *Journal of Logic and Computation*, 17(2):311–331, 2007.
[34] George Metcalfe, Nicola Olivetti, and Dov Gabbay. Sequent and hypersequent calculi for Abelian and Łukasiewicz logics. *ACM Transactions on Computational Logic*, 6(3):578–613, 2005.
[35] Robert K. Meyer and John K. Slaney. Abelian logic from A to Z. In Graham Priest, Richard Routley, and Jean Norman, editors, *Paraconsistent Logic: Essays on the Inconsistent*, pages 245–288. Philosophia Verlag, 1989.
[36] Paul S. Mostert and Allen L. Shields. On the structure of semigroups on a compact manifold with boundary. *The Annals of Mathematics, Second Series*, 65:117–143, 1957.
[37] Daniele Mundici. Satisfiability in many-valued sentential logic is NP-complete. *Theoretical Computer Science*, 52(1–2):145–153, 1987.
[38] Christos H. Papadimitriou. *Computational Complexity*. Theoretical Computer Science. Addison Wesley, 1993.
[39] Petr Savický, Roberto Cignoli, Francesc Esteva, Lluís Godo, and Carles Noguera. On product logic with truth constants. *Journal of Logic and Computation*, 16(2):205–225, 2006.
[40] Alexander Schrijver. *Theory of Linear and Integral Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Chichester, 1998.
[41] Richard Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9(1):67–72, 1979.
[42] Volker Weispfenning. The complexity of the word problem for Abelian ℓ-groups. *Theoretical Computer Science*, 48(1):127–132, 1986.

ZUZANA HANIKOVÁ
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8, Czech Republic
Email: zuzana@cs.cas.cz