The complexity of admissible rules of Łukasiewicz logic

Emil Jeřábek

Institute of Mathematics of the Academy of Sciences

Žitná 25, 115 67 Praha 1, Czech Republic, email: jerabek@math.cas.cz

February 28, 2012

Abstract

We investigate the computational complexity of admissibility of inference rules in infinite-valued Łukasiewicz propositional logic (**Ł**). It was shown in [13] that admissibility in **Ł** is checkable in PSPACE. We establish that this result is optimal, i.e., admissible rules of **Ł** are PSPACE-complete. In contrast, derivable rules of **Ł** are known to be coNP-complete.

Keywords: Łukasiewicz logic, admissible rule, computational complexity, PSPACE-complete.

1 Introduction

The concept of admissible rules was introduced by Lorenzen [15]: a rule is admissible in a logical system if the set of theorems (tautologies) of the logic is closed under instances of the rule. In contrast to this, a rule is said to be derivable in a logic if it belongs to its usual consequence relation. In classical logic, derivable and admissible rules coincide (such logics are known as structurally complete), but nonclassical logics typically have nonderivable admissible rules, and often admissible rules exhibit much more complicated structure than derivable rules.

Admissible rules are well understood for certain classes of transitive modal and superintuitionistic logics. Admissibility in such logics was investigated in a series of papers by Rybakov, culminating in the monograph [20]. Another impetus was provided by the characterization of unification and admissibility in terms of projective formulas, introduced by Ghilardi [5, 6]. This incited work on bases of admissible rules including Iemhoff [7, 8, 9] and Jeřábek [10, 12]. Rybakov has recently studied admissible rules in some temporal logics, see e.g. [21, 22].

The computational complexity of admissibility of rules in modal and superintuitionistic logics was investigated by Jeřábek [11]. In particular, admissible rules of typical transitive logics (e.g., **IPC**, **K4**, **S4**, **GL**, **Grz**) are coNEXP-complete, in contrast to derivable rules of these logics, which are usually PSPACE-complete. (The coNEXP-hardness part of the result holds for a quite wide class of logics, including even coNP-logics of bounded depth such as $\mathbf{K4BD_3}$.) On the other hand, admissibility has the same complexity as derivability in structurally complete and almost structurally complete logics such as extensions of **S4.3** (for a nontrivial example of another kind, the $\{\to, \neg\}$-fragment of **IPC** has PSPACE-complete admissibility problem by Cintula and Metcalfe [4]). Wolter and Zakharyaschev [23] proved that unification and admissibility in the extension of **K** or **K4** with the universal modality is undecidable.

Admissible rules of Łukasiewicz logic were investigated by Jeřábek [13, 14]. The main result of [13] is a description of a geometric criterion for admissibility of multiple-conclusion rules in **Ł**, which in particular implies that admissibility in **Ł** (of single-conclusion or multiple-conclusion rules, as well as the universal theory of free MV-algebras) is computable in PSPACE. However, no nontrivial lower bound on the complexity of admissibility in **Ł** is given, except that Łukasiewicz tautologies are coNP-complete by Mundici [18]. In [14], an explicit basis of admissible rules of **Ł** is presented, and a description of admissibly saturated formulas of **Ł** is given. Recently, Marra and Spada [16] established that unification in **Ł** is nullary (i.e., of the worst possible type), and Cabrer [2] proved that admissibly saturated formulas in **Ł** are exact.

The purpose of this paper is to show that the PSPACE upper bound on the complexity of admissibility in **Ł** from [13] is in fact optimal: admissibility in **Ł** is PSPACE-complete. The main technical ingredient is a construction of a representation of the configuration graph of a polynomial-space Turing machine by a rational polyhedron which can be described by a polynomial-size Łukasiewicz formula. We also show an exponential lower bound on the length of paths involved in the main criterion for admissibility in **Ł** from [13] (matching an exponential upper bound given there).

The paper is organized as follows. In Section 2 we provide some background and fix the notation. Section 3 presents the criterion for admissibility in **Ł** from [13] and provides an example where the criterion requires exponentially long paths. Section 4 is devoted to the proof of our main result, viz. PSPACE-completeness of admissibility in **Ł**. Section 5 consists of concluding remarks.

2 Preliminaries

We assume the reader is familiar with basic notions from computational complexity theory, such as Turing machines and the definitions of time and space complexity. We recall that NP is the class of languages accepted by polynomial-time nondeterministic Turing machines, and PSPACE is the class of languages accepted by polynomial-space Turing machines (whether deterministic or nondeterministic is immaterial here, by Savitch's theorem). A language L is PSPACE-complete if $L \in$ PSPACE, and every PSPACE-language is polynomial-time reducible to L. The reader can consult e.g. Arora and Barak [1] for details and further background.

The *standard MV-algebra* is the structure $[0,1]_{\mathbf{Ł}} = \langle [0,1], \cdot_{\mathbf{Ł}}, \to_{\mathbf{Ł}}, \min, \max, 0, 1\rangle$ in the signature $L_{\mathbf{Ł}} = \langle \cdot, \to, \wedge, \vee, \bot, \top\rangle$, where $x \cdot_{\mathbf{Ł}} y = \max\{0, x+y-1\}$ and $x \to_{\mathbf{Ł}} y = \min\{1, 1-x+y\}$. The language of Łukasiewicz logic (**Ł**) consists of propositional formulas built freely from variables x_i, $i \in \omega$, and connectives from $L_{\mathbf{Ł}}$. (We will sometimes employ other letters, such as t, u, v, for propositional variables.) A valuation is a homomorphism e from the free algebra of formulas into $[0,1]_{\text{Ł}}$. A formula φ is an **Ł**-*tautology* if $e(\varphi) = 1$ for every valuation e. A *substitution* is an endomorphism on the algebra of formulas. A substitution σ is a *unifier* of a formula φ if $\sigma(\varphi)$ is an **Ł**-tautology. A *rule* is an expression $\Gamma \,/\, \varphi$, where Γ is a finite set of formulas. Such a rule is *admissible* if every common unifier of Γ is also a unifier of φ. More generally, a *multiple-conclusion rule* is an expression $\Gamma \,/\, \Delta$, where Γ, Δ are finite sets of formulas; it is admissible if every common unifier of Γ is also a unifier of some formula from Δ. We write $\Gamma \mathrel{|\!\!\sim}_{\text{Ł}} \Delta$ if $\Gamma \,/\, \Delta$ is an admissible rule.

McNaughton's theorem [17] states that a function $\varphi\colon [0,1]^m \to [0,1]$ is representable by a Łukasiewicz formula in m variables if and only if it is a *McNaughton function*, i.e., a continuous piecewise linear (more precisely, affine) function with integer coefficients. We will identify formulas with their McNaughton functions when their syntactic shape is not relevant. For any McNaughton function φ, its *truth set* $t(\varphi) := \varphi^{-1}(1)$ is a *rational polyhedron*: we can write $t(\varphi) = \bigcup_{i<k} C_i$, where each C_i is a rational polytope, i.e., the convex hull of a finite subset of $\mathbb{Q}^m$. Conversely, any rational polyhedron $P \subseteq [0,1]^m$ equals $t(\varphi)$ for some formula φ. We will write $t(\Gamma) := \bigcap_{\varphi\in\Gamma} t(\varphi)$, and we denote the convex hull of a set $X \subseteq \mathbb{R}^m$ by $\mathrm{Conv}(X)$. We have the following quantitative version of the easy implication in McNaughton's theorem (see e.g. [13]):

Lemma 2.1 *Let Γ be a finite set of formulas in m variables closed under subformulas, and $n = |\Gamma|$. For all $j < 2^n$, $i < n$, and $\varphi \in \Gamma$, there are linear functions $L_{j,i}$ and $L_{j,\varphi}$ with integer coefficients and L^1-norm at most n such that the polytopes*

$$C_j = \{x \in [0,1]^m : \forall i < n\, L_{j,i}(x) \geq 0\}$$

satisfy

$$\bigcup_{j<2^n} C_j = [0,1]^m,$$

and

$$L_{j,\varphi}(x) = \varphi(x)$$

for each $x \in C_j$ and $\varphi \in \Gamma$. Moreover, we can compute the coefficients of $L_{j,i}$ and $L_{j,\varphi}$ in polynomial time given Γ and j. □

This also implies similar bounds on the expression of $t(\Gamma)$ as a rational polyhedron.

3 Admissible rules of Łukasiewicz logic

The following characterization of admissibility in **Ł** was given in [13]. First, let us say that a set $X \subseteq \mathbb{R}^m$ is *anchored* if its affine hull contains a *lattice point* (i.e., an element of $\mathbb{Z}^m$). Using efficient computability of Herbrand's normal form, it can be seen that given a sequence $x_1, \dots, x_n \in \mathbb{Q}^m$, it is polynomial-time decidable whether $\{x_1, \dots, x_n\}$ is anchored.

Theorem 3.1 (Jeřábek [13]) *Let Γ and Δ be finite sets of formulas in m variables, and let $\{C_j : j < r\}$ be a sequence of rational polytopes such that $\bigcup_{j<r} C_j = t(\Gamma)$. The following are equivalent.*

(*i*) $\Gamma \not\mathrel{|\!\!\sim}_{\text{Ł}} \Delta$.

(*ii*) *There exists* $a \in \{0,1\}^m \cap t(\Gamma)$ *such that for every* $\psi \in \Delta$ *there exists a sequence* $\{j_i : i \le k\}$ *of indices* $j_i < r$ *such that*

(*α*) $a \in C_{j_0}$,

(*β*) C_{j_i} *is anchored for each* $i \le k$,

(*γ*) $C_{j_i} \cap C_{j_{i+1}} \ne \varnothing$ *for each* $i < k$,

(*δ*) *there exists* $x \in C_{j_k}$ *such that* $\psi(x) < 1$. □

We can rephrase this in graph-theoretic language as follows. Given Γ, consider the decomposition $t(\Gamma) = \bigcup_{j<r} C_j$, $r \le 2^n$, from Lemma 2.1. Let the *polytope graph* $G_\Gamma = \langle V_\Gamma, E_\Gamma \rangle$ be the graph with vertex set $V_\Gamma = \{0, \dots, r-1\}$ such that j and j' are connected by an edge in E_Γ iff $C_j \cap C_{j'} \ne \varnothing$. Let the *anchored polytope graph* A_Γ be the induced subgraph of G_Γ consisting of vertices j such that C_j is anchored. Let us call j a *lattice vertex* if $C_j \cap \{0,1\}^m \ne \varnothing$, and j is a *counterexample* to a formula ψ if there exists $x \in C_j$ such that $\psi(x) < 1$.

Corollary 3.2 $\Gamma \not\mathrel{|\!\!\sim}_{\text{Ł}} \Delta$ *iff there exists a connected component of* A_Γ *containing a lattice vertex and a counterexample to* ψ *for every* $\psi \in \Delta$. □

We also have:

Theorem 3.3 (Jeřábek [13]) $\mathrel{|\!\!\sim}_{\text{Ł}}$ *is computable in* PSPACE. □

The original proof of Theorem 3.3 in [13] was a bit complicated due to an effort to optimize the space requirements of the algorithm. However, if we are not interested in a particular polynomial bound, we can easily understand Theorem 3.3 as follows. Since we can check in NP whether a given polytope contains a lattice point or is a counterexample to ψ (the latter is even in P, using linear programming), Corollary 3.2 reduces (non)admissibility in **Ł** to reachability in A_Γ. If an undirected graph is explicitly given by a list of vertices and edges, reachability is computable in logarithmic space (even deterministic, by a breakthrough result of Reingold [19]; however, nondeterministic would do the job for us). Instead of an input tape, the algorithm can be implemented using oracle access to a black box which can tell whether a given label denotes a valid vertex of the graph, and given two vertices, whether they are connected by an edge. Now, our graph is exponentially large, which blows up the complexity from logarithmic to polynomial space. The whole algorithm is PSPACE provided we can simulate the input oracle in polynomial space as well. In fact, we can do it in NP: given j, we can compute the linear functions defining the polytope C_j; then we can check in NP whether it is anchored, and given two such polytopes, we can check whether they intersect.

It should be clear from this description that the only obstacle preventing us from computing $\mathrel{|\!\!\sim}_{\text{Ł}}$ more efficiently is that the path connecting in A_Γ a counterexample to ψ to a lattice vertex may be exponentially long. For example, it is not difficult to see that if we could always find such a path of polynomial length, we could test $\not\mathrel{|\!\!\sim}_{\text{Ł}}$ in NP. Thus, if we intend to prove that $\mathrel{|\!\!\sim}_{\text{Ł}}$ is PSPACE-complete, we had better make sure that there are cases where the distance from any counterexample to ψ to any lattice vertex is exponentially long.

The construction in the proof of our main result will indeed have this property (when applied to an exponential-time PSPACE algorithm). However, we decided to also include a simpler direct construction, since it illustrates more transparently the motivation behind the general case, which may help the reader in understanding the underlying idea. Theorem 3.4 and its proof are not needed for our main result, hence a reader who wants to get straight to the point may safely skip to the next section.

Theorem 3.4 *Given m, we can construct in time* $\mathrm{poly}(m)$ *formulas φ_m, ψ_m of size $O(m^2)$ in m variables such that $\varphi_m \not\vdash_{\mathbf{L}} \psi_m$, but every sequence $\{j_i : i \leq k\}$ as in Theorem* 3.1 *must have length $k = \Omega(2^m)$.*

Proof: Let $G_m = \langle V_m, E_m \rangle$ be the m-dimensional hypercube graph: i.e., $V_m = \mathcal{P}(m)$ (where we use the set-theoretical identity $m = \{0, \dots, m-1\}$ to simplify the notation), and $\langle u, v \rangle \in E_m$ iff $|u \vartriangle v| = 1$, where $\vartriangle$ denotes symmetric difference. We will define an exponentially long path P_m in G_m, and embed G_m in $[0,1]^m$ in such a way that P_m is represented by the graph A_φ for a polynomial-size formula φ.

The path $P_m = \langle v_{m,0}, \dots, v_{m,2^m-1} \rangle$ will be a Hamiltonian path in G_m starting at the vertex $v_{m,0} = \varnothing$, and we define it inductively as follows: P_0 is the trivial one-vertex path in G_0. If P_m was already constructed, we define P_{m+1} by taking two copies of P_m, one in each of the hyperplanes $\{v \subseteq m+1 : m \notin v\}$ and $\{v \subseteq m+1 : m \in v\}$, and joining them by an edge connecting the two copies of the far end-point of P_m. That is,

$$P_{m+1} = \langle v_{m,0}, \dots, v_{m,2^m-1}, v_{m,2^m-1} \cup \{m\}, \dots, v_{m,0} \cup \{m\} \rangle.$$

We will actually need a more explicit description of the edges belonging to P_m. First, since $v_{m,0} = \varnothing$ for every m, the other end-point of P_m is $v_{m,2^m-1} = \{m-1\}$ for $m > 0$. Then it is easy to show by induction on m that every vertex $v \in V_m$ is connected in P_m to

- $v \vartriangle \{0\}$, and
- $v \vartriangle \{\min(v) + 1\}$ if possible (i.e., if $v \neq \varnothing, \{m-1\}$).

We can identify each $v \subseteq m$ with the binary string describing its characteristic function. That is, we make $V_m = \{0,1\}^m$, and then P_m consists of the following edges, where we denote concatenation by juxtaposition:

- $0w$—$1w$, for $w \in \{0,1\}^{m-1}$,
- $0^k 10w$—$0^k 11w$, for $k < m-1$, $w \in \{0,1\}^{m-k-2}$.

The end-points of P_m are 0^m and $0^{m-1}1$. By abuse of language, we will denote the set of edges of P_m as P_m.

We now construct a representation of G_m in $[0,1]^m$. Put $B_0 = [0, 1/5]$, $B_1 = [3/5, 4/5]$, and $B = [0, 4/5]$. We represent a vertex $v \in \{0,1\}^m$ by the polytope

$$B_v = \prod_{i<m} B_{v_i}.$$

If $e = \{v, w\} \in E_m$, let $j < m$ be the unique position such that $v_j \neq w_j$. We represent e by the polytope

$$C_e = \prod_{i \neq j} B_{v_i} \times B,$$

where the B is supposed to go to the jth position in the product. Let

$$C = \bigcup_{e \in P_m} C_e.$$

The following properties are easy to verify:

Claim 1

(*i*) *Each B_v and C_e is an anchored rational polytope.*

(*ii*) *B_v are pairwise disjoint.*

(*iii*) *If $v \in e$, then $B_v \subseteq C_e$, otherwise $B_v \cap C_e = \varnothing$.*

(*iv*) *C_e are pairwise disjoint, except that $C_e \cap C_{e'} = B_v$ when $e \cap e' = \{v\}$.*

(*v*) *B_v contains a lattice point iff $v = 0^m$. C_e contains a lattice point iff $0^m \in e$.*

(*vi*) *C is connected. If $v \neq 0^m, 0^{m-1}1$, then $C \smallsetminus B_v$ is disconnected, and its two connected components correspond to the two subpaths of P_m on either side of v.*

The key property is that even though there are exponentially many edges in P_m, we can write C in another way using only polynomially many operations, because of the highly uniform way in which P_m can be described. Indeed,

$$C = (B \times B_*^{m-1}) \cup \bigcup_{k < m-1} (B_0^k \times B_1 \times B \times B_*^{m-k-2}),$$

where $B_* = B_0 \cup B_1$. Fix formulas β_0, β_1, β in one variable such that $t(\beta_i) = B_i$, $t(\beta) = B$, and put $\beta_* = \beta_0 \vee \beta_1$. Then we have $C = t(\varphi_m)$, where

$$\varphi_m = \Big(\beta(x_0) \wedge \bigwedge_{i=1}^{m-1} \beta_*(x_i)\Big) \vee \bigvee_{k<m-1} \Big(\bigwedge_{i<k} \beta_0(x_i) \wedge \beta_1(x_k) \wedge \beta(x_{k+1}) \wedge \bigwedge_{i=k+2}^{m-1} \beta_*(x_i)\Big).$$

Notice that $|\varphi_m| = O(m^2)$. Let δ_i be fixed formulas in one variable such that $t(\delta_i) = [0,1] \smallsetminus \mathrm{int}(B_i)$, and put

$$\psi_m = \bigvee_{i<m-1} \delta_0(x_i) \vee \delta_1(x_{m-1}),$$

so that

$$t(\psi_m) = D := [0,1]^m \smallsetminus \mathrm{int}(B_{0^{m-1}1}).$$

Since C is a connected union of anchored polytopes, contains a lattice point $\vec{0}$, and a counterexample to ψ_m, we have

$$\varphi_m \not\vdash_{\text{Ł}} \psi_m.$$

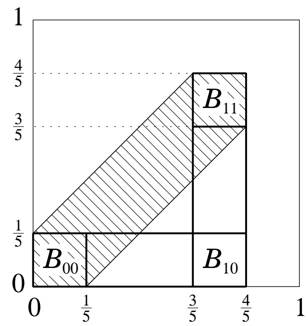

Figure 1: The convex hull of $B_{00} \cup B_{11}$ is disjoint from B_{10}

On the other hand, if we write $t(\varphi)$ as $\bigcup_{e\in P_m} C_e$, then it follows from Claim 1 that the only path in A_φ connecting a lattice vertex to a counterexample to ψ_m traces P_m all the way from one end to the other end, hence it has length $2^m - 1$.

A subtle issue (which will not arise in the PSPACE-completeness proof below) is that in principle it may be possible to write C as a union of polytopes $\bigcup_{i<r} C'_i$ in a different way so that there is a shorter path from a lattice vertex to a counterexample to ψ_m. However, we have:

Claim 2 *Any convex subset of C intersects at most two B_v.*

Proof: Let $X \subseteq C$ be convex. If $x \in X \cap B_u$ and $y \in X \cap B_w$, the line segment $\mathrm{Conv}(x, y)$ is included in $X \subseteq C$ and it is connected, hence by Claim 1 it hits B_v for every v lying on the subpath of P_m joining u to w. Thus, if we assume for contradiction that X intersects three or more B_v, we can find u, v, w such that $\{u, v\}, \{v, w\} \in P_m$, $x \in X \cap B_u$, $y \in X \cap B_w$, $\mathrm{Conv}(x, y) \cap B_v \neq \varnothing$. Let $i \neq j$ be the unique coordinates such that $u_i \neq v_i$ and $v_j \neq w_j$, and let π be the projection to the ith and jth coordinates. Then $\pi(B_u) = B_{u_i u_j}$ and similarly for B_v, B_w, and π preserves convex hulls, hence there exist $u', v', w' \in \{0,1\}^2$ such that $\{u', v'\}, \{v', w'\} \in P_2$, and $\mathrm{Conv}(B_{u'} \cup B_{w'}) \cap B_{v'} \neq \varnothing$. However, this is easily seen to be false, see Figure 1. □ (Claim 2)

By Claim 1, removing any B_v from C disconnects the unique lattice point $\vec{0}$ from $C \smallsetminus D$, hence any path using the C'_i witnessing $\varphi_m \not\vdash_{\mathbf{L}} \psi_m$ as in Theorem 3.1 must intersect every B_v. By Claim 2, such a path has to have length at least 2^{m-1}. □

4 PSPACE-completeness

We will use an idea similar to the proof of Theorem 3.4 to simulate a computation of a polynomial-space Turing machine. In a nutshell, we will embed in $[0, 1]^m$ the configuration graph of the machine. (This subsumes the ability to create exponentially long paths as a polynomial-space computation may take exponential time.) In order to get a description of the graph by a polynomial-size formula, we will exploit the locality of Turing machines: the behaviour of the machine in a particular configuration is determined by a constant-size subset of the configuration, and anything outside this subset is passed unchanged to the next step.

In order to simplify the construction, we will not simulate completely general polynomial-space Turing machines, but we will first reduce to a special case that is more manageable. Let us say that a deterministic Turing machine M is in a *normal form* if it has the following properties. M has a single tape with alphabet $\Sigma = \{0, 1\}$ (using no extra blank symbol) which serves both as the input tape and as a work tape. M has states with labels from $Q = \{0, \ldots, s\}$, $s \geq 1$, where 0 is the initial state, and 1 is the unique accepting state. There is no rejecting state, on non-accepted inputs M eventually enters an infinite loop. The tape head moves left or right in every step. Let $T \colon Q \times \Sigma \to Q \times \Sigma \times \{1, -1\}$ be the transition function of M (i.e., when M is in state q with the tape head in position h reading symbol $x \in \Sigma$, and $T(q, x) = \langle r, y, d \rangle$, then M writes y to the tape, moves head to position $h + d$, and enters state r). We require $T(1, x) = \langle 1, y, d \rangle$; i.e., T is defined in such a way that once M enters the accepting state, it can never leave it. (This is only a formal technical requirement, as after entering the accepting state M is supposed to stop anyway. However, it will be convenient for our simulation to pretend that the machine continues to work in order to reduce the number of exceptions.) On an input $w \in \{0, 1\}^n$, M starts with head at position 0 of the tape and $w = w_0 \ldots w_{n-1}$ written at positions $0, \ldots, n-1$ of the tape. A *normal run* of M on input of length n is a computation during which M does not attempt to access positions -1 or n of the tape (which in particular implies that it is confined to space n). We consider acceptance by M as a promise problem, whose positive instances are inputs accepted by a normal run of M, and negative instances are inputs that make M enter an infinite normal run avoiding the accepting state.

Lemma 4.1 *Every $L \in$ PSPACE is polynomial-time reducible to the acceptance problem of a Turing machine in normal form.*

Proof: Let $L \subseteq \Sigma_0^*$, and let M_1 be a deterministic Turing machine accepting L in space $p(n) \geq n$ using k work tapes (along with the input tape) with alphabet $\Sigma_1 \supseteq \Sigma_0 \cup \{\epsilon\}$, where ϵ is the blank symbol, and p is a polynomial. Let $\Sigma_1' = \{a' : a \in \Sigma_1\}$ be a disjoint copy of Σ_1, and $\Diamond \notin \Sigma_1 \cup \Sigma_1'$ an auxiliary symbol. We can represent a configuration c of M_1 by the string

$$\tilde{c} = \Diamond \tilde{a}_0^0 \tilde{a}_1^0 \ldots \tilde{a}_{p(n)-1}^0 \Diamond \tilde{a}_0^1 \tilde{a}_1^1 \ldots \tilde{a}_{p(n)-1}^1 \Diamond \cdots \Diamond \tilde{a}_0^k \tilde{a}_1^k \ldots \tilde{a}_{p(n)-1}^k \Diamond,$$

where a_i^j is the ith symbol on the jth tape (the input tape being the 0th tape), and $\tilde{a}_i^j = (a_i^j)'$ if the head of tape j is on position i, $\tilde{a}_i^j = a_i^j$ otherwise. We can simulate easily the computation of M_1 by a single-tape Turing machine M_2 with alphabet $\Sigma_2 = \Sigma_1 \cup \Sigma_1' \cup \{\Diamond\}$ operating with the representations $\tilde{c}$ of configurations of M_1 in such a way that M_2 never attempts to move past the first or last $\Diamond$ delimiters. Choose $d \in \omega$ and pairwise distinct $\overline{a} \in \{0, 1\}^d$ for each $a \in \Sigma_2$. We can simulate M_2 by a machine M in normal form by translating each symbol a of the simulated tape of M_2 with the sequence $\overline{a}$ of d binary symbols. A run of M is normal whenever it starts with the tape containing the translation of a valid representation $\tilde{c}$ of a configuration of M_1. Then L is reducible to the acceptance problem of M via the polynomial-

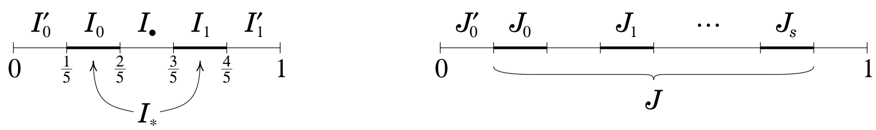

Figure 2: The layout of auxiliary intervals

time function $f(x)$ which computes the translation of $\tilde{c}$, where c is the initial configuration of M_1 on input x. □

Theorem 4.2 *Admissibility of either single-conclusion or multiple-conclusion rules in* $\mathbf{Ł}$ *is* PSPACE*-complete.*

Proof: That $\mathrel{|\!\!\sim}_{\mathbf{Ł}} \in$ PSPACE was established in [13], hence it suffices to show that non-admissibility of single-conclusion rules in $\mathbf{Ł}$ is PSPACE-hard. Given a PSPACE language L, let f be a polynomial-time function and M a Turing machine in normal form such that $x \in L$ iff M accepts $f(x)$, and the run of M on any $w = f(x)$ is normal.

Let n be given. A configuration of M is a sequence $c = \langle q, h, x_0, \dots, x_{n-1} \rangle$, where $q \in Q = \{0, \dots, s\}$ is the current state, $h < n$ is the position of the head, and $x_0, \dots, x_{n-1}$ is the content of the tape. Put $I_0 = [1/5, 2/5]$, $I_1 = [3/5, 4/5]$, $I_* = I_0 \cup I_1$, $I_\bullet = [2/5, 3/5]$, $I'_0 = [0, 1/5]$, $I'_1 = [4/5, 1]$, $J_q = [(2q+1)/(2s+3), (2q+2)/(2s+3)]$ for $q \le s$, $J_* = \bigcup_{q \le s} J_q$, $J = [1/(2s+3), (2s+2)/(2s+3)] = \mathrm{Conv}(J_*)$, $J'_0 = [0, 1/(2s+3)]$ (cf. Figure 2). We represent a configuration $c = \langle q, h, x_0, \dots, x_{n-1} \rangle$ by the polytope

$$H_c = J_q \times \prod_{i<n} I_{\delta_{h,i}} \times \prod_{i<n} I_{x_i} \subseteq [0,1]^{2n+1},$$

where $\delta_{h,i}$ is Kronecker's delta. We represent the input $w = f(x)$ of length n by

$$F_w = J'_0 \times \prod_{i<n} I'_{\delta_{0,i}} \times \prod_{i<n} I'_{w_i}.$$

Acceptance by M will be represented by (the complement of) the polyhedron

$$B = [0,1]^{2n+1} \smallsetminus \mathrm{int}(J_1 \times I_*^{2n}).$$

Finally, we have to find a representation for transition edges. For any configuration c, let $\sigma(c)$ be its successor configuration (which is unique, as M is deterministic). We will construct a polyhedron E_c representing an edge connecting c to $\sigma(c)$ as follows.

Claim 1 *For every $q \in Q$ and $x \in \{0,1\}$, we can choose a rational polyhedron $C_{q,x} \subseteq [0,1]^4$ with the following properties, where $T(q,x) = \langle r, y, d \rangle$ is the transition function of M:*

(*i*) *$C_{q,x}$ is connected, and it is a finite union of polytopes of dimension* 4.

(*ii*) *$C_{q,x}$ intersects $J_q \times \{\langle 3/5, 2/5, (2+x)/5 \rangle\}$ and $J_r \times \{\langle 2/5, 3/5, (2+y)/5 \rangle\}$.*

(*iii*) $C_{q,x}$ *is included in* $J \times I_\bullet^3$, *and more precisely, in*

$$(J_q \times \{\langle 3/5, 2/5, (2+x)/5\rangle\}) \cup (J_r \times \{\langle 2/5, 3/5, (2+y)/5\rangle\}) \cup (J \times \operatorname{int}(I_\bullet)^3).$$

(*iv*) *The sets* $C_{q,x}$ *are pairwise disjoint.*

Proof: The reader may well take it on faith that there is room enough in the 4-dimensional space to embed a finite collection of edges, but for definiteness, we can construct $C_{q,x}$ explicitly as follows. Let us enumerate $Q \times \{0,1\} = \{\langle q_i, x_i\rangle : i < m\}$ (hence $m = 2(s+1)$), and put $\langle r_i, y_i, d_i\rangle = T(q_i, x_i)$. Denote $[a \pm \varepsilon] = [a - \varepsilon, a + \varepsilon]$ and $c(t, x, y) = (1-t)x + ty$. We put $z_{q,i} = c((1+i)/(2m+1), \min(J_q), \max(J_q))$, $\overline{z}_{q,i} = z_{q,m+i}$, $h_i = c((1+i)/(m+1), 2/5, 3/5)$. Let C'_{q_i,x_i} be the broken line with end-points $\langle z_{q_i,i}, 3/5, 2/5, (2+x_i)/5\rangle$, $\langle z_{q_i,i}, 1/2, 1/2, h_i\rangle$, $\langle \overline{z}_{r_i,i}, 1/2, 1/2, h_i\rangle$, $\langle \overline{z}_{r_i,i}, 2/5, 3/5, (2+y_i)/5\rangle$. Then C'_{q_i,x_i} satisfies all the requirements above except that it has only dimension 1. Let $\varepsilon > 0$, $\varepsilon \in \mathbb{Q}$ be such that the L^∞-distance of C'_{q_i,x_i} and $C'_{q_{i'},x_{i'}}$ is at least 3ε for each $i \neq i'$. We can define C_{q_i,x_i} to be the union of the following three polytopes:

(i) The convex hull of $\langle z_{q_i,i}, 3/5, 2/5, (2+x_i)/5\rangle$ and $[z_{q_i,i} \pm \varepsilon] \times [1/2 \pm \varepsilon]^2 \times [h_i \pm \varepsilon]$,

(ii) $[z_{q_i,i}, \overline{z}_{r_i,i}] \times [1/2 \pm \varepsilon]^2 \times [h_i \pm \varepsilon]$,

(iii) The convex hull of $\langle \overline{z}_{r_i,i}, 2/5, 3/5, (2+y_i)/5\rangle$ and $[\overline{z}_{r_i,i} \pm \varepsilon] \times [1/2 \pm \varepsilon]^2 \times [h_i \pm \varepsilon]$.

Notice that C_{q_i,x_i} is contained within the closed ε-neighbourhood of C'_{q_i,x_i} (in the L^∞-norm). Then it is easy to see that C_{q_i,x_i} satisfies all our requirements. □ (Claim 1)

Given a configuration $c = \langle q, h, x_0, \dots, x_{n-1}\rangle$, let $T(q, x_h) = \langle r, y, d\rangle$, so that

$$\sigma(c) = \langle r, h+d, x_0, \dots, x_{h-1}, y, x_{h+1}, \dots, x_{n-1}\rangle.$$

We define

$$E_c = C_{q,x_h} \times \prod_{i \neq h, h+d} I_0 \times \prod_{i \neq h} I_{x_i},$$

where the four coordinates of C_{q,x_h} are supposed to go to the 0th, $(h+1)$st, $(h+d+1)$st, and $(h+n+1)$st coordinates in the product; that is, more precisely,

$$(1) \quad E_c = \{\langle t, u_0, \dots, u_{n-1}, v_0, \dots, v_{n-1}\rangle : u_i \in I_0 \ (i \neq h, h+d), \\ v_i \in I_{x_i} \ (i \neq h), \langle t, u_h, u_{h+d}, v_h\rangle \in C_{q,x_h}\}$$

(cf. the definition of H_c). We put $H = \bigcup_c H_c$, $E = \bigcup_c E_c$, $A_w = H \cup E \cup F_w$. Notice that we have

$$H = J_* \times \bigcup_{h<n} (I_0^h \times I_1 \times I_0^{n-h-1}) \times I_*^n,$$

$$(2) \qquad E = \bigcup_{q,h,x} (C_{q,x} \times I_0^{n-2} \times I_*^{n-1}),$$

$$B = \left(([0,1] \smallsetminus \operatorname{int}(J_1)) \times [0,1]^{2n}\right) \cup \bigcup_{i=1}^{2n} \left([0,1]^i \times ([0,1] \smallsetminus \operatorname{int}(I_*)) \times [0,1]^{2n-i}\right),$$

where the products in E have coordinates permuted as in the definition of E_c above.

Claim 2

(*i*) *H_c and F_w are full-dimensional* (*hence anchored*) *polytopes. E_c is a connected finite union of full-dimensional polytopes.*

(*ii*) *There is no lattice point in $H \cup E$, and there is one in F_w. F_w is disjoint from E, and it intersects H_c iff c is the initial configuration $\langle 0, 0, w \rangle$.*

(*iii*) *H_c are pairwise disjoint.*

(*iv*) *E_c intersects H_d iff $d = c$ or $d = \sigma(c)$.*

(*v*) *$E_c \smallsetminus H$ are pairwise disjoint.*

(*vi*) *$B \supseteq E \cup F_w$. B includes H_c iff c is not an accepting configuration.*

(*vii*) *The connected component of A_w containing F_w is included in B if and only if M does not accept w.*

Proof: (i), (ii), (iii), and (vi) are immediate from the definition.

(iv): Let $c = \langle q, h, x_0, \dots, x_{n-1} \rangle$. C_{q,x_h} intersects $J_q \times \{\langle 3/5, 2/5, (2 + x_h)/5 \rangle\} \subseteq J_q \times I_1 \times I_0 \times I_{x_h}$, hence E_c intersects H_c. Similarly, C_{q,x_h} intersects $J_r \times I_0 \times I_1 \times I_y$, where $\langle r, y, d \rangle = T(q, x_h)$, hence E_c intersects $H_{\sigma(c)}$. The remaining part of C_{q,x_h} is contained in $J \times \mathrm{int}(I_\bullet)^3$, and as $\mathrm{int}(I_\bullet) \cap I_* = \varnothing$, the corresponding part of E_c is disjoint from H.

(v): By the proof of (iv), $E_c \smallsetminus H$ corresponds to the part of C_{q,x_h} included in $J \times \mathrm{int}(I_\bullet)^3$. Let $c' = \langle q', h', x'_0, \dots, x'_{n-1} \rangle$, $T(q', x'_{h'}) = \langle r', y', d' \rangle$ be such that $E_c \cap E_{c'} \nsubseteq H$. If $h \neq h'$, the projection of $E_{c'}$ to the v_h-coordinate (using the notation of (1)) is included in I_*, whereas $E_c \smallsetminus H$ projects to the disjoint interval $\mathrm{int}(I_\bullet)$, a contradiction. Thus $h = h'$. If $d \neq d'$, then similarly the projections of $E_c \smallsetminus H$ and $E_{c'}$ to the u_{h+d}-coordinate are included in $\mathrm{int}(I_\bullet)$ and I_0, respectively, hence we may assume $d = d'$. If $x_i \neq x'_i$ for some $i \neq h$, then the projections of E_c and $E_{c'}$ to the v_i-coordinate are I_{x_i} and $I_{x'_i}$. Finally, if $x_i = x'_i$ for all $i \neq h$, then $E_c = C_{q,x_h} \times X$ and $E_{c'} = C_{q',x'_h} \times X$ for a certain set X, up to a permutation of coordinates (the same one for both). Since the sets $C_{q,x}$ are pairwise disjoint, we must have $q = q'$ and $x_h = x'_h$, i.e., $c = c'$.

(vii): Assume that the component is not included in B. There exists a sequence $P_0, \dots, P_r$ of polyhedrons such that $P_0 = F_w$, each P_i for $i > 0$ is H_c or E_c, $P_i \cap P_{i+1} \neq \varnothing$, and $P_r \nsubseteq B$. By (vi), $P_r = H_c$ for some accepting configuration c. By (ii), $P_1 = H_{0,0,w}$. By (v), we may assume that no two E_c are adjacent in the sequence. By (iv), this implies that E_c can only be adjacent to H_c and $H_{\sigma(c)}$. By (iii), no two H_c are adjacent. Summing up, there exists a sequence $c_0, \dots, c_p$ of pairwise distinct configurations such that $c_0 = \langle 0, 0, w \rangle$ is the initial configuration, c_p is an accepting configuration, and for each $i < p$, $c_{i+1} = \sigma(c_i)$ or $c_i = \sigma(c_{i+1})$. However, if $c_i = \sigma(c_{i+1})$ and $c_{i+2} = \sigma(c_{i+1})$, then $c_i = c_{i+2}$, and we can delete c_{i+1} and c_{i+2} from the sequence. Thus, we can assume that there exists $j \leq p$ such that $c_{i+1} = \sigma(c_i)$ for all $i < j$, and $c_i = \sigma(c_{i+1})$ for all $i \geq j$. Since c_p is an accepting configuration and successors of accepting configurations are again accepting, c_j is also an accepting configuration, hence M accepts w.

Conversely, if $c_0, \dots, c_p$ is the sequence of configurations of M during an accepting computation on w, then the sequence $F_w, H_{c_0}, E_{c_0}, H_{c_1}, \dots, E_{c_{p-1}}, H_{c_p}$ witnesses that F_w is in A_w connected to the complement of B. □ (Claim 2)

We now express A_w and B by propositional formulas (using variables $t, u_0, \dots, u_{n-1}, v_0, \dots, v_{n-1}$ in the same fashion as in (1)). Let $\iota_0, \iota_1, \iota_*, \iota'_0, \iota'_1, \bar{\iota}_*, \zeta_*, \zeta'_0, \bar{\zeta}_1$ be formulas in one variable whose truth sets are $I_0, I_1, I_*, I'_0, I'_1, [0,1] \smallsetminus \mathrm{int}(I_*), J_*, J'_0, [0,1] \smallsetminus \mathrm{int}(J_1)$, respectively, and for any $q \le s$ and $x \in \{0,1\}$, let $\gamma_{q,x}$ be a formula in four variables whose truth set is $C_{q,x}$. Notice that these formulas only depend on M and not on n or w, hence they are fixed constant-size formulas. Then we put

$$\begin{aligned}
\eta_n &= \zeta_*(t) \wedge \bigvee_{h<n} \bigwedge_{i<n} \iota_{\delta_{h,i}}(u_i) \wedge \bigwedge_{i<n} \iota_*(v_i),\\
\varepsilon_n &= \bigvee_{q,x,h,d} \Big(\gamma_{q,x}(t, u_h, u_{h+d}, v_h) \wedge \bigwedge_{i \ne h, h+d} \iota_0(u_i) \wedge \bigwedge_{i \ne h} \iota_*(v_i)\Big),\\
\varphi_w &= \zeta'_0(t) \wedge \iota'_1(u_0) \wedge \bigwedge_{i=1}^{n-1} \iota'_0(u_i) \wedge \bigwedge_{i<n} \iota'_{w_i}(v_i),\\
\alpha_w &= \eta_n \vee \varepsilon_n \vee \varphi_w,\\
\beta_n &= \bar{\zeta}_1(t) \vee \bigvee_{i<n} (\bar{\iota}_*(u_i) \vee \bar{\iota}_*(v_i)),
\end{aligned}$$

where the disjunction in ε_n is taken over all $q \le s$, $x \in \{0,1\}$, $d \in \{1,-1\}$, and $h < n$ such that $T(q,x) = \langle r, y, d \rangle$ and $0 \le h + d < n$. It follows from (2) that $t(\eta_n) = H$, $t(\varepsilon_n) = E$, $t(\varphi_w) = F_w$, $t(\alpha_w) = A_w$, and $t(\beta_n) = B$, hence using Claim 2 and Theorem 3.1,

$$\alpha_w \not\mid\!\sim_{\mathbf{Ł}} \beta_n \quad \text{iff} \quad M \text{ accepts } w.$$

We have $|\alpha_w| = O(n^2)$ and $|\beta_n| = O(n)$, and it is easy to see that α_w and β_n are polynomial-time (or even log-space) computable given w, hence

$$x \in L \quad \text{iff} \quad \alpha_{f(x)} \not\mid\!\sim_{\mathbf{Ł}} \beta_{|f(x)|}$$

provides a polynomial-time reduction of L to $\not\mid\!\sim_{\mathbf{Ł}}$. □

Remark 4.3 It follows from Theorem 4.2 that the quasi-equational theory of free MV-algebras is PSPACE-hard. Since the universal theory of free MV-algebras was shown to be in PSPACE in [13], both these theories are PSPACE-complete.

5 Conclusion

We have settled the computational complexity of admissibility in **Ł** by showing its PSPACE-completeness. One consequence is that the algorithm for admissibility given in [13] cannot be significantly improved. Moreover, it confirms the intuition suggested by the criterion from [13] that admissibility in **Ł** is best viewed in terms of undirected reachability in the anchored polytope graph, at least in the sense that it leads to the right complexity estimate of the problem. It is also worth mentioning that similarly to the case of natural transitive modal logic and intuitionistic logic, the admissibility problem in **Ł** turns out to be more complex than the derivability problem (assuming NP $\neq$ PSPACE).

Our result resolves Problem 5.2 from [13]. We remark that Problem 5.1 is also essentially solved: Marra and Spada [16] proved the unification type of **Ł** to be nullary, which also shows that some formulas cannot have projective approximations, despite that all formulas have admissibly saturated approximations by [14]. The description of projective formulas in **Ł** remains an intriguing open problem (some results in this direction have been obtained by Cabrer and Mundici [3]), nevertheless, in view of the nonexistence of projective approximations, it is not directly relevant to admissibility; a question more to the point is a characterization of admissibly saturated formulas, which is satisfactorily resolved by [14, 2]. Leaving admissibility aside, an interesting related problem is to get a better understanding of unification in **Ł**. For instance, despite its nullary type, it is conceivable that one can describe (infinite) complete sets of unifiers in some transparent algorithmic way.

Funding

This work was supported by Institutional Research Plan AV0Z10190503, grant IAA100190902 of GA AV ČR, project 1M0545 of MŠMT ČR, and a grant from the John Templeton Foundation.

Acknowledgements

I would like to thank the referee for useful suggestions.

References

[1] Sanjeev Arora and Boaz Barak, *Computational complexity: A modern approach*, Cambridge University Press, 2009.

[2] Leonardo M. Cabrer, *Simplicial geometry of unital lattice-ordered abelian groups*, submitted, arXiv:1202.5947 [math.CO].

[3] Leonardo M. Cabrer and Daniele Mundici, *Rational polyhedra and projective lattice-ordered abelian groups with order unit*, Communications in Contemporary Mathematics, to appear, arXiv:0907.3064 [math.AT].

[4] Petr Cintula and George Metcalfe, *Admissible rules in the implication-negation fragment of intuitionistic logic*, Annals of Pure and Applied Logic 162 (2010), no. 2, pp. 162–171.

[5] Silvio Ghilardi, *Unification in intuitionistic logic*, Journal of Symbolic Logic 64 (1999), no. 2, pp. 859–880.

[6] ———, *Best solving modal equations*, Annals of Pure and Applied Logic 102 (2000), no. 3, pp. 183–198.

[7] Rosalie Iemhoff, *On the admissible rules of intuitionistic propositional logic*, Journal of Symbolic Logic 66 (2001), no. 1, pp. 281–294.

[8] __________, *Intermediate logics and Visser's rules*, Notre Dame Journal of Formal Logic 46 (2005), no. 1, pp. 65–81.

[9] __________, *On the rules of intermediate logics*, Archive for Mathematical Logic 45 (2006), no. 5, pp. 581–599.

[10] Emil Jeřábek, *Admissible rules of modal logics*, Journal of Logic and Computation 15 (2005), no. 4, pp. 411–431.

[11] __________, *Complexity of admissible rules*, Archive for Mathematical Logic 46 (2007), no. 2, pp. 73–92.

[12] __________, *Independent bases of admissible rules*, Logic Journal of the IGPL 16 (2008), no. 3, pp. 249–267.

[13] __________, *Admissible rules of Łukasiewicz logic*, Journal of Logic and Computation 20 (2010), no. 2, pp. 425–447.

[14] __________, *Bases of admissible rules of Łukasiewicz logic*, Journal of Logic and Computation 20 (2010), no. 6, pp. 1149–1163.

[15] Paul Lorenzen, *Einführung in die operative Logik und Mathematik*, Grundlehren der mathematischen Wissenschaften vol. 78, Springer, 1955 (in German).

[16] Vincenzo Marra and Luca Spada, *Duality, projectivity, and unification in Łukasiewicz logic and MV-algebras*, preprint, 2011.

[17] Robert McNaughton, *A theorem about infinite-valued sentential logic*, Journal of Symbolic Logic 16 (1951), no. 1, pp. 1–13.

[18] Daniele Mundici, *Satisfiability in many-valued sentential logic is NP-complete*, Theoretical Computer Science 52 (1987), no. 1–2, pp. 145–153.

[19] Omer Reingold, *Undirected connectivity in log-space*, Journal of the Association for Computing Machinery 55 (2008), no. 4, article no. 17.

[20] Vladimir V. Rybakov, *Admissibility of logical inference rules*, Studies in Logic and the Foundations of Mathematics vol. 136, Elsevier, 1997.

[21] __________, *Logical consecutions in discrete linear temporal logic*, Journal of Symbolic Logic 70 (2005), no. 4, pp. 1137–1149.

[22] __________, *Linear temporal logic with Until and Before on integer numbers, deciding algorithms*, in: Computer Science – Theory and Applications (D. Grigoriev, J. Harrison, and E. A. Hirsch, eds.), Lecture Notes in Computer Science vol. 3967, Springer, 2006, pp. 322–333.

[23] Frank Wolter and Michael Zakharyaschev, *Undecidability of the unification and admissibility problems for modal and description logics*, ACM Transactions on Computational Logic 9 (2008), no. 4, article no. 25.