Uwe Egly
Hans Tompits

On Different Proof-Search Strategies for Orthologic

Abstract. In this paper, we consider three different search strategies for a cut-free sequent system formalizing orthologic, and estimate the respective search spaces. Applying backward search, there are classes of formulae for which both the minimal proof length and the search space are exponential. In a combined forward and backward approach, all proofs are polynomial, but the potential search space remains exponential. Using a forward strategy, the potential search space becomes polynomial yielding a polynomial decision procedure for orthologic and the word problem for free ortholattices.

Keywords: orthologic, minimal quantum logic, proof theory, Gentzen system, proof complexity, polynomial decision procedure.

1. Introduction

Comparing the efficiency of calculi and proof-search strategies is an important topic in automated deduction. Usually, measurements and comparisons are performed on the basis of test sets like the well-known TPTP library. Only in few publications (like, e.g., [10]), the size of the search space is analytically analyzed.

In this paper, we compare three different proof-search strategies for a cut-free propositional sequent calculus, called GOL, characterizing a sub-classical logic known as *orthologic*. One of the interesting applications of this calculus is that it facilitates a decision procedure for the word problem for free ortholattices [5, 12]. Moreover, orthologic provides important relationships to other non-classical logics like Sambin's basic logic [3]. In general, orthologic emanated from algebraic investigations about the logical structure of orthomodular lattices, strongly connected to the Hilbert-space formalism of quantum mechanics [1]. Accordingly, the logic of orthomodular lattices is also termed *quantum logic*, whilst orthologic itself is (historically) referred to as *minimal quantum logic*.

The distinguishing feature of GOL is the stipulation that sequents contain at most two formula occurrences. As a consequence, in contrast to classical propositional logic, not all inferences can be permuted over each other. Moreover, the usual normal forms like conjunctive or disjunctive normal form are not possible because distributivity does not hold in general.

The considered strategies for proof search are (i) backward search (from the end sequent towards the axioms), (ii) a combination of backward and forward search, and (iii) forward search (from the axioms to the end sequent). In the last approach, the subformula property is used in order to distinguish the relevant axiom and inference schemata.

Our results can be summarized as follows. Using backward search, there is an infinite class $(F_n)_{n\geq 0}$ of formulae such that each proof of F_n is exponential in the size of F_n. Hence, the corresponding search space is trivially exponential. Using the mixed strategy and a sequence notation for the resulting proofs, *all* proofs are polynomial, but the potential search space remains exponential. However, using forward search, *all proofs together with the corresponding search space become polynomial.* Consequently, forward search in GOL implements a polynomial decision procedure for orthologic (and thus for the word problem for free ortholattices). The interesting point to note here is that the polynomial decision procedure is established by an analysis of permutation properties of GOL and the potential search spaces for different search strategies.

All the mentioned search strategies have been implemented. The first two strategies are implemented using PROLOG, whereas the last strategy is implemented in C. We briefly compare running times of our forward search implementation with running times of examples reported by McCune [8].

The paper is organized as follows. Section 2 gives background terminology and notation. Section 3 explores permutation properties of GOL, and Section 4 contains the main results. Section 5 concludes with a short discussion.

2. Background

Throughout this paper, we use a propositional language with $\neg, \wedge, \vee$ as the only connectives. Formulae are defined according to the usual formation rules and are denoted by lower-case letters; sets are denoted by uppercase letters. In order to save parentheses, we assume that $\neg$ binds stronger than $\wedge$, which in turn binds stronger than $\vee$.

DEFINITION 1. *The* logical complexity *of a formula f, denoted by* $\mathsf{lcomp}(f)$, *is defined as the number of occurrences of logical connectives $\neg, \wedge, \vee$ in f. The* weight *of f, denoted by* $\mathsf{weight}(f)$, *is* $\mathsf{lcomp}(f)$ *plus the number of occurrences of atomic formulae in f.*

DEFINITION 2. *An n–*restricted *sequent is an ordered pair of the form $M \vdash N$, where M and N are sets of formulae and $|M| + |N| \leq n$. M is the* antecedent, *N is the* succedent *of the sequent, and $|X|$ denotes the cardinality of the set X.*

$$\frac{M \vdash a, N}{M \vdash a \vee b, N}\,R1 \qquad \frac{M \vdash b, N}{M \vdash a \vee b, N}\,R2 \qquad \frac{M, a \vdash N \quad M, b \vdash N}{M, a \vee b \vdash N}\,R3$$

$$\frac{M, a \vdash N}{M, a \wedge b \vdash N}\,R4 \qquad \frac{M, b \vdash N}{M, a \wedge b \vdash N}\,R5 \qquad \frac{M \vdash a, N \quad M \vdash b, N}{M \vdash a \wedge b, N}\,R6$$

$$\frac{M, a \vdash N}{M \vdash \neg a, N}\,R7 \qquad \frac{M \vdash a, N}{M, \neg a \vdash N}\,R8$$

$$\frac{M \vdash N}{M, a \vdash N}\,R9 \qquad \frac{M \vdash N}{M \vdash a, N}\,R10$$

Figure 1. The inference rules of GOL.

For a singleton set $M = \{f\}$, we often use M to denote the formula f.

Since sequents consists of two sets (rather than two multisets), implicit contractions take place if both sequent formulae are identical and occur at the same side of the sequent.

DEFINITION 3. *The* degree *of a formula f, denoted by* $\mathsf{deg}(f)$*, is defined as the weight of f. For a 2–restricted sequent S, we define:*

$$\mathsf{deg}(S) = \begin{cases} 0 & \text{if } S = \vdash\,; \\ \mathsf{deg}(m) \cdot \mathsf{deg}(n) & \text{if } S \text{ is } (m \vdash n),\ (m, n \vdash) \text{ or } (\vdash m, n); \\ \mathsf{deg}(m)^2 & \text{if } S \text{ is } (m \vdash) \text{ or } (\vdash m). \end{cases}$$

The last clause in the above definition represents the fact that a sequent with one formula occurrence coincides with a sequent with two "copies" of the same formula at the same side.

We use the following sequent calculus, GOL, adapted from the sequent calculus OCL+ in [12] (see also [5]). Formal objects of GOL are 2–restricted sequents; (logical) axioms of GOL are of the form $a \vdash a$ for some formula a, and the inference rules are depicted in Figure 1.

Principal formulae, side formulae etc. are defined in the usual way. Applications of inference rules in backward proof search are called *reductions*. Rules R1, R2, R4, and R5 are called *α–rules*; rules R7 and R8 are called *negation rules*; rules R3 and R6 are called *β–rules*; and rules R9 and R10 are called *weakening rules*. An *α–formula* is the principal formula of an α–rule, and a *β–formula* is the principal formula of a β–rule. Rules containing a single premise are called *unary*, and rules with two premises are called *binary*. The cut rule, given by

$$\frac{M_1 \vdash c, N_1 \quad M_2, c \vdash N_2}{M_1, M_2 \vdash N_1, N_2,}\,Rcut$$

is admissible. The system $\mathsf{GOL}+\mathsf{Rcut}$ is the calculus GOL extended by the cut rule.

An occurrence Ψ of a formula g in a formula f is *positive* (*negative*) if Ψ is preceeded by an even (odd) number of negation signs. In such a case, we also say that g occurs in positive (negative) *polarity* in f.

Let $S\colon M \vdash N$ be a sequent. Following Smullyan [14], we represent S by a set of *signed formulae* of the form $\{\, \mathsf{T}M, \mathsf{F}N \,\}$, where the sign T is chosen for antecedent formulae and the sign F is chosen for succedent formulae. The choice of T for the antecedent formulae and F for the succedent formulae is for historical reasons where, instead of a proof of a formula, a refutation of the negated formula is considered. Therefore, positive formula occurrences of the end sequent occur in the succedent; the sign of such a formula is F. Likewise, negative formula occurrences of the end sequent occur in the antecedent; the sign of such a formula is T.

We use K and L as meta-variables for signs. For a sign K, $\overline{\mathsf{K}} = \mathsf{T}$ if $\mathsf{K} = \mathsf{F}$, and $\overline{\mathsf{K}} = \mathsf{F}$ if $\mathsf{K} = \mathsf{T}$.

An important feature of cut-free proofs is the *subformula property*, which states that each proof α with end sequent $\vdash f$ contains only subformulae of f. This property holds in a strict sense for the calculus GOL: for each sequent $M \vdash N$ occurring in proof α of $\vdash f$, elements of M are negative subformula occurrences of f, and elements of N are positive subformula occurrences of f.

We consider two different types of proofs in GOL. Proofs of the first type are usual *tree* proofs, and proofs of the second type are *sequence* proofs. In both cases, if we speak about proofs, we usually mean *cut-free* proofs. In addition to the axioms of GOL, we allow non-logical axioms in intermediate proofs, but we eventually replace non-logical axioms S by a GOL-proof with end sequent S.

DEFINITION 4. *A* sequence proof *of a sequent S from a set of non-logical axioms A in* GOL *is a sequence $S_1, \ldots, S_n$ of sequents such that $S = S_n$ and, for $1 \le m \le n$ and $k, l < m$,*

1. *S_m is a logical or non-logical axiom, or*
2. *S_m is the conclusion of a unary inference with premise S_k, or*
3. *S_m is the conclusion of a binary inference with premises S_k and S_l.*

We denote proofs by lower-case Greek letters. The number of sequents in a (tree or sequence) proof φ is denoted by $\#seq(\varphi)$.

For a tree proof φ of a sequent S in GOL, there are many corresponding sequence proofs of the same end sequent. We can speak about *the* sequence proof corresponding to φ, if we choose a deterministic translation.

Definition 5. *Let T be the set of tree proofs in* GOL*, and S the set of sequence proofs in* GOL*. We define the map $\theta\colon T \to S$ as follows:*

$$\theta(\varphi) = \begin{cases} U & \text{if } \varphi \text{ is an axiom } U \text{ (logical or non-logical);} \\ \theta(\varphi_1)\, U & \text{if } \varphi \text{ is of the form } \frac{\varphi_1}{U}; \\ \theta(\varphi_1)\,\theta(\varphi_2)\, U & \text{if } \varphi \text{ is of the form } \frac{\varphi_1 \ \varphi_2}{U}. \end{cases}$$

It is easily shown that θ is well defined and that $\#seq(\varphi) = \#seq(\theta(\varphi))$.

3. Proof Normal Forms in GOL

In this section, we study normal forms of proofs in GOL. We use some permutation properties of inferences in GOL, but, due to lack of space, we cannot discuss all of them. In contrast to full propositional logic, where all inferences can be permuted over each other, there are so-called *non-permutabilities* of inferences in GOL.[1]

In the following, we consider permutation schemata (R, R') where R, R' are one of α (for α–rules), β (for β–rules), w (for weakening rules) or n (for negation rules).

Definition 6. *Rule R' is* permutable over *rule R (towards the axioms), if for all applications r of R and r' of R' such that*

1. *r occurs immediately above r',*
2. *the principal formula f_r of r is not a side formula of r', and*
3. *A is the set of all premises of r, S the conclusion of r, r' takes premises from $B \cup \{S\}$ ($|B| = 0$ if r' is unary, and $|B| = 1$ otherwise) and yields the conclusion S',*

there is a proof of S' from $A \cup B'$ in which an application of R', preceded by zero or one weakening, occurs immediately above an application of R, which is followed by zero or one weakening. B' is obtained as follows. If $B = \emptyset$ then $B' = \emptyset$. If $B = \{ Q \}$ then reduce f_r in Q by applying r. This results either in one premise $B' = \{ Q_1 \}$ or in two premises $B' = \{ Q_1, Q_2 \}$.

[1] See [6, 15, 2] for a discussion of permutation properties in calculi for classical and intuitionistic first-order logic.

The construction of B' is required for the permutation schemata (β, β) and (n, β) in a full discussion of all permutation properties. The weakenings above an application of R and below an application of R' (after permutation) are necessary if implicit contractions occur in the inference figures. Implicit contractions occur (in a backward search) when a side formula of the applied inference already occurs in the conclusion.

In the following, we investigate some permutation properties of GOL, which can be used in the search for proofs in order to restrict the potential search space.

THEOREM 7. *In* GOL*-proofs, rule* R' *cannot be permuted over* R (*towards the axioms*) *in the following cases:*

$$\begin{array}{r||c|c|c|}
R & \alpha & \alpha & w \\
R' & \beta & w & \beta
\end{array}$$

Moreover, weakening is permutable over β*–rules and negation rules.*

PROOF. Sequents justifying the non-permutabilities (R, R') are as follows:

$$\begin{array}{ll}
(\alpha,\, \beta): & a \vee b \;\vdash\; b \vee a\,; \\
(\alpha,\, \mathrm{w}): & a \;\vdash\; b \vee \neg b\,; \\
(\mathrm{w},\, \beta): & b \wedge \neg b \vee a \;\vdash\; a.
\end{array}$$

The permutation schemata for $(\beta$, w) and $(n$, w) are as follows (recall that K and L denote signs).

$(\beta$, w):

$$\dfrac{\dfrac{\mathsf{K}a \quad \mathsf{K}b}{\mathsf{K}a \circ b}\,\beta}{\mathsf{K}a \circ b\,\mathsf{L}c}\,w \qquad \dfrac{\dfrac{\mathsf{K}a}{\mathsf{K}a\,\mathsf{L}c}\,w \quad \dfrac{\mathsf{K}b}{\mathsf{K}b\,\mathsf{L}c}\,w}{\mathsf{K}a \circ b\,\mathsf{L}c}\,\beta$$

$(n$, w):

$$\dfrac{\dfrac{\overline{\mathsf{K}}a}{\mathsf{K}\neg a}\,n}{\mathsf{K}\neg a\,\mathsf{L}b}\,w \qquad \dfrac{\dfrac{\overline{\mathsf{K}}a}{\overline{\mathsf{K}}a\,\mathsf{L}b}\,w}{\mathsf{K}\neg a\,\mathsf{L}b}\,n$$

■

By using the permutations of inferences from Theorem 7, we show in Lemma 9 that we can restrict our attention to proofs where weakenings occur directly below α–rules. Proofs having this property are said to be in *weakening normal form.*

DEFINITION 8. *Let* φ *be a tree proof of* $\vdash f$ *in* GOL. φ *is in* weakening normal form (*WNF*) *if any application of R9 or R10 occurs immediately below an application of an* α*–rule.* φ *is called* simple *if it is weakening-free.*

LEMMA 9. *Let* ψ *be a tree proof of* $\vdash f$ *in* GOL. *Then, there is a tree proof* ψ' *of* $\vdash f$ *in WNF, and the number of branches in* ψ' *is not greater than the number of branches in* ψ.

PROOF. Let ψ be a tree proof of $\vdash\ f$. We first replace inferences stemming from implicit contractions (i.e., a premise consists of one formula although the conclusion consists of two formulae and the inference rule is different from weakening) by weakenings. We then remove weakenings, where the weakening formula appears already in the premise. Finally, we eliminate weakenings below applications of β–rules and negation rules.

Let b be an α–formula with immediate subformulae b_1 and b_2. Consider the α–rule

$$\frac{\mathsf{K}b_1}{\mathsf{K}b\,\mathsf{K}b_1}\ \alpha$$

where b is the principal formula of this inference. Replace this α–rule by the weakening rule

$$\frac{\mathsf{K}b_1}{\mathsf{K}b\,\mathsf{K}b_1.}\ w$$

Proceed similarly for negations and β–formulae and use the following two pairs of inference schemata:

$$\frac{\mathsf{K}b}{\mathsf{K}b\,\overline{\mathsf{K}}\neg b}\ n \qquad \frac{\mathsf{K}b}{\mathsf{K}b\,\overline{\mathsf{K}}\neg b}\ w \qquad\qquad \frac{\mathsf{K}b_1 \quad \mathsf{K}b_1\,\mathsf{K}b_2}{\mathsf{K}b\,\mathsf{K}b_1}\ \beta \qquad \frac{\mathsf{K}b_1}{\mathsf{K}b\,\mathsf{K}b_1}\ w$$

Let ψ_1 be the result of these transformations. Obviously, the number of branches in ψ_1 is not greater than the number of branches in ψ.

Next, those applications of weakening are removed from ψ_1 where the weakening formula appears already in the premise. The situation is as follows ($M \cup N \neq \emptyset$):

$$\frac{M\ \vdash\ N}{M\ \vdash\ N}\ w \tag{$*$}$$

The newly introduced (occurrence of the) weakening formula is removed by an implicit contraction due to the idempotence property of sets. Let φ be the proof obtained from ψ_1 by omitting such weakenings.

Let us now turn our attention to applications of weakening directly below applications of β–rules or negation rules. Let $w(\varphi)$ be the number of applications of weakening immediately below an application of these rules, and let $W_1(\varphi), \ldots, W_{w(\varphi)}(\varphi)$ denote these applications. Moreover, let $m_i(\varphi)$ ($1 \leq i \leq w(\varphi)$) be the maximal number of applications of β–rules or negation rules between the current weakening $W_i(\varphi)$ and the first application of an α–rule above $W_i(\varphi)$. Observe that, on any branch from $W_i(\varphi)$ towards the axioms, there must be (at least) one application of an α–rule. This follows immediately from the removal of all weakenings of the form $(*)$, and therefore the premises of all remaining weakenings must consist of exactly one formula. Since φ is a proof, each of these premises must be provable. Consequently, there must be an application of an α–rule on every branch from a premise to an axiom. Moreover, there is *no* application of weakening between $W_i(\varphi)$ and the first application of an α–rule above $W_i(\varphi)$. Otherwise, the unprovable empty sequent $\vdash$ would occur in a proof.

Let $S_{w(\varphi)} = \sum_{i=1}^{w(\varphi)} 3^{m_i(\varphi)}$ for $w(\varphi) > 0$ and $S_{w(\varphi)} = 0$ otherwise. Since $m_i(\varphi) \geq 1$ $(1 \leq i \leq w(\varphi))$, $S_{w(\varphi)} \geq w(\varphi) \geq 0$. We show that φ can be transformed into WNF without changing the number of branches. The proof is by induction on $k(\varphi) = w(\varphi) + S_{w(\varphi)}$.

Induction Basis. Assume $k(\varphi) = 0$. Then $w(\varphi) = 0$ and φ is already in WNF.

Induction Step. Assume $k(\varphi) = n > 0$, and suppose that, for all tree proofs φ' with $k(\varphi') < n$, there is a tree proof π (of the same end sequent) in WNF with the same number of branches as in φ'.

Let us arbitrarily choose an application $W_j(\varphi)$ $(1 \leq j \leq w(\varphi))$ such that no other $W_l(\varphi)$ $(l \neq j)$ occurs in the tree proof of the premise of $W_j(\varphi)$. Without loss of generality, we assume that $W_j(\varphi)$ is an application of $R9$. We distinguish two cases:

Case 1. The rule immediately above the considered weakening rule is a β–rule (this corresponds to the case (β, w) in the proof of Theorem 7). Then we have the situation

$$\cfrac{\cfrac{\overset{\varphi_1}{M_1 \vdash N_1} \qquad \overset{\varphi_2}{M_2 \vdash N_2}}{M \vdash N}\,\beta}{M,\, a \vdash N}\,R9$$

(the value of M, N, M_1, M_2, N_1, N_2 depends on the particular β–rule at hand). The indicated weakening rule $W_j(\varphi)$ is permuted over the β–rule immediately above, *without* changing the number of branches. The resulting tree proof, φ', is

$$\cfrac{\cfrac{\overset{\varphi_1}{M_1 \vdash N_1}}{M_1,\, a \vdash N_1}\,R9 \qquad \cfrac{\overset{\varphi_2}{M_2 \vdash N_2}}{M_2,\, a \vdash N_2}\,R9}{M,\, a \vdash N.}\,\beta$$

We distinguish two cases depending on the value of $m_j(\varphi)$.

Subcase 1. $m_j(\varphi) = 1$. By the indicated permutation of inference rules, $w(\varphi') = w(\varphi) - 1$ because the two indicated weakenings are now directly below applications of an α–rule. For φ', we estimate:

$$\begin{aligned} k(\varphi') &= w(\varphi') + \sum_{i=1}^{w(\varphi')} 3^{m_i(\varphi')} \\ &= w(\varphi) - 1 + \sum_{i=1}^{j-1} 3^{m_i(\varphi)} + \sum_{i=j+1}^{w(\varphi)} 3^{m_i(\varphi)} \ < \ k(\varphi). \end{aligned}$$

Subcase 2. $m_j(\varphi) > 1$. By the indicated permutation of inference rules, $w(\varphi') = w(\varphi) + 1$, but $m_j(\varphi') = m_j(\varphi) - 1$. For φ', we estimate:

$$\begin{aligned} k(\varphi') &= w(\varphi') + \sum_{i=1}^{w(\varphi')} 3^{m_i(\varphi')} \\ &= w(\varphi) + 1 + \sum_{i=1}^{j-1} 3^{m_i(\varphi)} + \sum_{i=j+1}^{w(\varphi)} 3^{m_i(\varphi)} + 2 \cdot 3^{m_j(\varphi)-1} \\ &= w(\varphi) + 1 + \sum_{i=1}^{j-1} 3^{m_i(\varphi)} + \sum_{i=j+1}^{w(\varphi)} 3^{m_i(\varphi)} + 3 \cdot 3^{m_j(\varphi)-1} - 3^{m_j(\varphi)-1} \\ &= w(\varphi) + 1 + S_{w(\varphi)} - 3^{m_j(\varphi)-1} \quad < \quad k(\varphi). \end{aligned}$$

In both subcases, the induction hypothesis yields a tree proof ψ in WNF with exactly the same number of branches as in φ.

Case 2. The rule immediately above the considered weakening rule is a negation rule (this corresponds to the case $(\neg, \mathrm{w})$ in the proof of Theorem 7). Then we have the situation

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ M_1 \vdash N_1\end{array}}{M \vdash N}\, n}{M,\, a \vdash N}\, R9$$

(the value of M, N, M_1, M_2, N_1, N_2 depends on the particular negation rule at hand). The indicated weakening rule $W_j(\varphi)$ is permuted over the negation rule immediately above *without* changing the number of branches. The resulting proof, φ', is

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ M_1 \vdash N_1\end{array}}{M_1,\, a \vdash N_1}\, R9}{M,\, a \vdash N.}\, n$$

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{a_n \vdash a_n}{\vdash a_n, \neg a_n}\,R7 \qquad \dfrac{\begin{array}{c}\alpha_{n-1}\\ \vdash f_{n-1}\end{array}}{\vdash a_n, f_{n-1}}\,R10}{\vdash a_n, \neg a_n \wedge f_{n-1}}\,R6 \\ \vdots \\ \vdash a_n, f_n\end{array} \qquad \begin{array}{c}\beta_n\\ \vdash b_n, f_n\end{array}}{\vdash a_n \wedge b_n, f_n}\,R6 \qquad \begin{array}{c}\alpha'_n\\ \vdash a_n \wedge c_n, f_n\end{array}}{\vdash (a_n \wedge b_n) \wedge (a_n \wedge c_n), f_n}\,R6}{\vdash (a_n \wedge b_n) \wedge (a_n \wedge c_n) \vee (((\neg a_n \wedge f_{n-1}) \vee \neg b_n) \vee \neg c_n)}\,R1
$$

Figure 2. Duplication of subproofs.

By the indicated permutation of inference rules, either $w(\varphi') = w(\varphi) - 1$ or $w(\varphi') = w(\varphi)$. In the latter case, $m_j(\varphi') = m_j(\varphi) - 1$. In both cases, $k(\varphi') < k(\varphi)$, and the induction hypothesis yields a proof ψ in WNF with exactly the same number of branches as in φ. ∎

It is immediately apparent that, in a proof in WNF, the premise of an application of a weakening rule consists of *exactly one* formula.

4. On Proof Length and Search Space in GOL

In this section, the length of proofs and the potential search spaces are analyzed for different search strategies. We first show that backward search can result in minimal tree proofs whose number of sequents is exponential in the weight of the input formula. The corresponding search space is also exponential. Then we discuss a combination of forward and backward search, for which all proofs are polynomial but the search space remains exponential in the worst case. Finally, we show that forward search implements a polynomial decision procedure.

4.1. A Class of Hard Formulae for Backward Search

Let a_0, a_i, b_i, c_i $(1 \leq i \leq n)$ be atoms and let f_0 be the formula $a_0 \vee \neg a_0$. For $n > 0$, we define:

$$f_n\colon (a_n \wedge b_n) \wedge (a_n \wedge c_n) \vee (((\neg a_n \wedge f_{n-1}) \vee \neg b_n) \vee \neg c_n).$$

Obviously, $\mathsf{weight}(f_0) = 4$ and $\mathsf{weight}(f_n) = 17n + 4$ for $n > 0$.

LEMMA 10. *Any tree proof of* $\vdash f_n$ *in* GOL *has more than* 2^n *sequents.*

PROOF SKETCH. For $n = 0$, any proof of f_n obviously has more than $2^0 = 1$ sequents. For $n > 0$, the reduction of the formula $a_n \wedge b_n$ must be prior to the reduction of $(\neg a_n \wedge f_{n-1}) \vee \neg b_n$, since otherwise we get a classically unprovable sequent. Similarly, the reduction of the formula $a_n \wedge c_n$ must be prior to the reduction of $((\neg a_n \wedge f_{n-1}) \vee \neg b_n) \vee \neg c_n$, for the same reason. But then, f_{n-1} has to be proven in two independent branches. The inference figure in Figure 2 illustrates this fact, where α'_n is similar to the left gray-shaded subproof with b_n replaced by c_n and some α–rules being deleted. Iterating this procedure eventually yields the desired minimal proof length. ■

4.2. A Mixed Proof Search Strategy

We have seen in Lemma 9 that we can restrict our attention to proofs in weakening normal form, where applications of weakening occur only below α–rules. We now discuss the *mixed* proof strategy where backward search is combined with forward search.

The basic idea underlying our procedure, `prove-cf`, is as follows. Since proofs without weakenings are short, we rely on backward proof search in the restricted weakening-free calculus. The proof procedure is called `prove-cfs`. An important property of this proof procedure is its ability to allow additional non-logical axioms. But which sequents should be considered as non-logical axioms? The key observation is to (potentially) allow sequents S of the form $b \vdash a$ or $a \vdash b$ where a is a proper α–subformula of f and b is a subformula of f. Consider the following (partial) proof φ in weakening normal form:

$$\dfrac{\dfrac{\begin{array}{c}\phi_1\\ S\end{array}}{\vdash a}\,\alpha}{b \vdash a.}\,w$$

In order to avoid the application of the weakening rule, we allow the sequent $b \vdash a$ to be a non-logical axiom. Moreover, to ensure a correct procedure, it must be guaranteed that $b \vdash a$ is provable in GOL. Let A be the set of non-logical axioms which is initially empty. A is constructed iteratively as follows.

1. We check for each proper α–subformula of f (in order of increasing weight) whether either $\vdash a$ or $a \vdash$ has a weakening-free tree proof from A (at most one of the above sequents can be provable).

2. If one of the above sequents (say $\vdash a$) is provable, we extend the set of non-logical axioms by the set:

$$\{ b \vdash a \mid b \text{ is subformula of } f \} \cup \{ \vdash b, a \mid b \text{ is subformula of } f \}$$

which consists of all sequents derivable from $\vdash a$ by weakening.

If A is actually extended, we try to prove the initial sequent $\vdash f$ from the new set of non-logical axioms using `prove-cfs`. If the result is `fail`, the generation of non-logical axioms is continued; otherwise, we are done. In the success case, we can construct, out of the different tree proofs for the appropriate α–subformulae of f, a sequence proof φ of $\vdash f$ which has no non-logical axioms and such that the number of sequents in φ is polynomial in $\mathsf{deg}(f)$. In the failure case, i.e., whenever there is no set A such that $\vdash f$ is provable from A, it is guaranteed that $\vdash f$ is not provable in GOL.

4.2.1. The Search for Weakening-free Proofs

In this subsection, we establish that weakening-free proofs have, in contrast to unrestricted tree proofs, a pleasant feature: for a sequent of the form $\vdash f$, all proofs in tree form have at most $\mathsf{deg}(f)^2$ branches. The underlying restricted calculus, i.e., GOL without the weakening rules, is obviously correct but incomplete. Later, we define a proof procedure based on this restricted calculus which is complete and which generates only short (polynomial) proofs (not necessarily in tree form).

THEOREM 11. *Let S be a 2-restricted sequent of the form $M \vdash N$. Let φ be a simple tree proof of S from a set A of (non-atomic) non-logical axioms. Then,* $\mathsf{deg}(S)$ *is greater than or equal to the number of branches in φ.*

PROOF. The proof is by induction on $l = \#seq(\varphi)$.

INDUCTION BASIS. Assume $l = 1$. Then S is an axiom or $S \in A$. Therefore, the only sequent in φ is S. Clearly, φ has exactly one branch and $\mathsf{deg}(S) \geq 1$ is greater than or equal to the number of branches in φ.

INDUCTION STEP. Assume $l > 1$, and suppose that, for all simple tree proofs φ' with $\#seq(\varphi') < l$,

$(*)$ $\mathsf{deg}(T_{\varphi'})$ is not less than the number of branches in φ',

where $T_{\varphi'}$ is the end sequent of φ'.

We distinguish the different possibilities for the last rule I in φ.

Case 1. $I = R7$ ($I = R8$). Then S is of the form $M \vdash \neg a, N$ ($M, \neg a \vdash N$). By the induction hypothesis, we have a simple tree proof ψ of $T\colon M, a \vdash N$ ($T\colon M \vdash a, N$) satisfying $(*)$. An additional application of R7 (R8) yields a simple tree proof of S with the same number of branches as in ψ.

Case 2. $I = R1$ ($I = R2$). Then S is of the form $M \vdash a \vee b, N$. By the induction hypothesis, we have a simple tree proof ψ of $T\colon M \vdash a, N$ or of $T\colon M \vdash b, N$ satisfying $(*)$. An additional application of R1 or R2 yields a simple tree proof of S with the same number of branches as in ψ.

Case 3. $I = R4$ ($I = R5$). Symmetric to Case 2 with a final application of R4 or R5.

Case 4. $I = R3$. Then S is of the form $M, a \vee b \vdash N$. By the induction hypothesis, we have two simple tree proofs ψ_1 of $T_1\colon M, a \vdash N$ and ψ_2 of $T_2\colon M, b \vdash N$ satisfying $(*)$. An additional application of R3 yields a proof ψ of S.

For calculating the number of branches of ψ, we assume without loss of generality that $M = \emptyset$ (recall the restriction that a sequent consists of at most two formula occurrences). Then we have:

$$
\begin{aligned}
\mathsf{deg}(T_1) &= \begin{cases} \mathsf{deg}(a) \cdot \mathsf{deg}(N) & \text{if } N \neq \emptyset \wedge N \neq \{a\}; \\ \mathsf{deg}(a)^2 & \text{else.} \end{cases} \\
\mathsf{deg}(T_2) &= \begin{cases} \mathsf{deg}(b) \cdot \mathsf{deg}(N) & \text{if } N \neq \emptyset \wedge N \neq \{b\}; \\ \mathsf{deg}(b)^2 & \text{else.} \end{cases} \\
\mathsf{deg}(S) &= \begin{cases} \mathsf{deg}(a \vee b) \cdot \mathsf{deg}(N) & \text{if } N \neq \emptyset \wedge N \neq \{a \vee b\}; \\ \mathsf{deg}(a \vee b)^2 & \text{else.} \end{cases}
\end{aligned}
$$

Since $\mathsf{deg}(a \vee b) = \mathsf{deg}(a) + \mathsf{deg}(b) + 1$, we have $\mathsf{deg}(S) > \mathsf{deg}(T_1) + \mathsf{deg}(T_2)$ in both cases. Furthermore, since $\mathsf{deg}(T_i)$ ($i = 1, 2$) is not less than the number of branches in ψ_i, we get that $\mathsf{deg}(S)$ is not less than the sum of branches of ψ_1 and ψ_2, which in turn gives us that $\mathsf{deg}(S)$ is not less than the number of branches in ψ.

Case 5. $I = R6$. Symmetric to Case 4 with a final application of R6. ■

Backward proof search in GOL for *simple* tree proofs terminates independently from the used strategy. Observe that, for proofs *with* weakenings, termination of backward proof search depends on the chosen strategy. Often, a loop check is required in order to guarantee termination. There are formulae for which there is no simple proof in tree form, i.e., weakening is essential for completeness.

4.2.2. The Search for Arbitrary Proofs

Let us now turn our attention to the general case. In order to get a complete procedure for GOL, we extend the incomplete proof search for weakening-free proofs by the generation of lemmata which can eventually be used to construct a short sequence proof with weakenings.

LEMMA 12. *Let f be a formula. If there is* no *proper α–subformula g of f such that $\vdash g$ or $g \vdash$ has a simple tree proof, then $\vdash f$ has a tree proof iff $\vdash f$ has a simple tree proof.*

PROOF. ($\Leftarrow$) Trivial.

($\Rightarrow$) Assume that φ is a tree proof of $\vdash f$ with weakenings. Due to Lemma 9, it is sufficient to restrict φ to be in WNF. There must be at least one application of R9 or R10 and the premise sequent of all applications of R9 or R10 must consist of *exactly one* formula. Due to the subformula property of cut-free proofs, this single formula must be a subformula of f.

Let us arbitrarily choose an application, r, of R9 or R10 such that no other application of R9, R10 occurs in the tree proof of the premise of r. Let the premise sequent of r be of the form $\vdash g$ or $g \vdash$, where g is a proper α–subformula of f. There must be a simple tree proof ψ of $\vdash g$ or $g \vdash$ which occurs as a subproof in φ, because we have chosen r in such a way that no other application of R9 or R10 occurs in ψ. The existence of a simple tree proof ψ contradicts our global assumption that there is no proper α–subformula g of f such that $\vdash g$ or $g \vdash$ has a simple tree proof. Consequently, φ must be simple. ■

In the procedure `prove-cf` depicted in Figure 3, `prove-cfs`, `car`, and `cdr` are used. The last two items are usual LISP functions: `car` returns the first element of a nonempty list, or `nil` otherwise; `cdr` returns the list without the first element for a list with more than one element, or `nil` otherwise. The procedure `prove-cfs(T, A)` searches backward for a *simple* tree proof of the input sequent T with non-logical axioms from A. It returns the proof or `fail`. Recall that `prove-cfs` is correct but incomplete.

The procedure `prove-cf(f)` "computes" a cut-free sequence proof of $\vdash f$ or returns `fail`, if no such proof exists. The following properties of the algorithm are proved in the sequel:

- The algorithm terminates with less than $3 \cdot \mathsf{deg}(f)$ calls of the procedure `prove-cfs`. Moreover, the degrees of the arguments are polynomially bounded. This is shown in Lemma 13.

```
φ := prove-cfs( ⊢ f, ∅);
if φ ≠ fail then return φ;
A := ∅;
S : list of unprocessed proper α-subformulae of f,
   sorted by increasing weight;
while S ≠ nil ∧ φ = fail do
   h := car(S);
   S := cdr(S);
   φ^p_h := prove-cfs( ⊢ h, A);
   if φ^p_h = fail then
     φ^n_h := prove-cfs(h ⊢ , A);
     if φ^n_h ≠ fail then
       A := A ∪ { (h ⊢ m), (h,m ⊢ ) | m is a subformula of f};
       φ := prove-cfs( ⊢ f, A);
     fi
   else
     A := A ∪ { (m ⊢ h), ( ⊢ m,h) | m is a subformula of f};
     φ := prove-cfs( ⊢ f, A);
   fi
od
return φ;
```

Figure 3. The procedure `prove-cf`.

- Correctness and completeness is shown in Theorem 16 and Theorem 17, respectively.
- If the return value of `prove-cf(f)` is not `fail`, then a polynomial sequence proof φ can be constructed from all simple tree proofs generated by `prove-cfs` (Theorem 17).

In what follows, $n_\alpha(f)$ denotes the number of proper α–subformulae of f.

LEMMA 13. *The procedure* `prove-cf(f)` *terminates with less than* $3 \cdot \mathsf{deg}(f)$ *calls of the subroutine* `prove-cfs` *with arguments* $g_1, \ldots, g_m$ *and* $A_1, \ldots, A_m$ $(m \leq n_\alpha(f) + 1)$ *such that*

$$\mathsf{deg}(g_i) + \mathsf{deg}(A_i) < \mathsf{deg}(f) + 2 \cdot n_\alpha(f) \cdot \mathsf{deg}(f)^3. \tag{1}$$

Proof. First, recall that `prove-cfs` terminates and that the length of the list S before the `while` statement in `prove-cf` is $n_\alpha(f)$. In the body of the `while` statement, the first element of S is selected and this selected item is removed from S. The loop invariant is:

$$\text{length of } S + \# \text{ of repetitions of the } \texttt{while} \text{ statement} = n_\alpha(f). \qquad (2)$$

The `while` statement terminates after at most $n_\alpha(f)$ iterations. In each of these iterations, at most three calls of `prove-cfs` are performed. Since $n_\alpha(f) < \mathsf{deg}(f)$, the total number of `prove-cfs` calls in `prove-cf(f)` is less than $3 \cdot \mathsf{deg}(f)$. The first arguments of all calls of `prove-cfs` are subformulae of f; therefore, $\mathsf{deg}(g_i) \leq \mathsf{deg}(f)$ for $1 \leq i \leq m$, $m \leq n_\alpha(f) + 1$. For each of the at most $2 \cdot n_\alpha(f) \cdot \mathsf{deg}(f)$ non-logical axioms, the degree is less than $\mathsf{deg}(f)^2$. Thus, the degree of the arguments of a call of `prove-cfs` is at most $\mathsf{deg}(f) + 2 \cdot n_\alpha(f) \cdot \mathsf{deg}(f)^3$. ■

Lemma 14. *Let $(h_i)_{1 \leq i \leq l}$ be the sequence of proper α–subformulae of f in increasing weight for which a proof φ_i of either $\vdash h_i$ or $h_i \vdash$ is obtained in a single call of* `prove-cf(f)`*, and let A_i be the corresponding set of non-logical axioms. Then, there exist tree proofs φ'_i of either $\vdash h_i$ or $h_i \vdash$ in* GOL *without non-logical axioms, for $1 \leq i \leq l$.*

Proof Sketch. The proof is by induction on l.

Induction Basis. Assume $l = 1$. Then, $A_1 = \emptyset$ and φ_1 does not have non-logical axioms.

Induction Step. Assume $l > 1$, and suppose that for all $m < l$, either $\vdash h_m$ or $h_m \vdash$ has a proof φ'_m in GOL without non-logical axioms.

Consider h_l and φ_l. This proof can have non-logical axioms either of the form $a \vdash h_m$ or of the form $h_m \vdash a$, for a subformula a of f and $m < l$. Proofs for all non-logical axioms are provided by the induction hypothesis together with weakening. Replacing all non-logical axioms in φ_l by their proofs results in a proof φ'_l without non-logical axioms. ■

Corollary 15. *Let A be the set of all non-logical axioms in* `prove-cf`*. Then, all elements from A are provable in* GOL *without non-logical axioms.*

Theorem 16. *Let f be a formula unprovable in* GOL*. Then, the procedure* `prove-cf(f)` *returns* `fail`*.*

Proof. Obviously, the result `fail` can only be returned in the last line of `prove-cf` because `prove-cfs(` $\vdash$ `f,`$\emptyset$`)` returns `fail` for an unprovable formula f. The proof is by *reductio ad absurdum*.

Assume that $\vdash f$ is not provable in GOL but `prove-cf(f)` returns a value φ different from `fail`. Then a call of `prove-cfs(` $\vdash$ `f, A)` must return φ. Since all non-logical axioms from A are provable in GOL without non-logical axioms, there is a proof of $\vdash f$ in GOL without non-logical axioms. This, however, contradicts our assumption that f is not provable in GOL. ■

THEOREM 17. *Let $(h_i)_{1\leq i\leq m}$ be the sequence of proper α–subformulae of f in increasing weight for which a proof of either $\vdash h_i$ or $h_i \vdash$ is obtained in a single call of* `prove-cf(f)`*. Let $(A_i)_{1\leq i\leq m}$ be the corresponding sets of non-logical axioms and let $(\varphi_i)_{1\leq i\leq m}$ be the corresponding proofs. Moreover, let φ be the last proof generated by* `prove-cf(f)`*. Then, there exists a sequence proof of $\vdash f$ in* GOL without non-logical axioms *and with at most* $2 \cdot \mathsf{deg}(f)^4$ *sequents.*

PROOF. First, observe that, whenever h_i is a proper subformula of h_j, then $i < j$. Furthermore, recall that $\#seq(\theta(\varphi)) = \#seq(\varphi)$ for a tree proof φ. The mapping θ is applied to each subproof obtained in a call of `prove-cfs`, i.e.,

$$\psi\colon \theta(\varphi_1)\ \ldots\ \theta(\varphi_m)\,\theta(\varphi).$$

Obviously, ψ represents a correct sequence proof of $\vdash f$ *without* non-logical axioms.

It remains to be shown that $\#seq(\psi) \leq 2 \cdot \mathsf{deg}(f)^4$. Recall that any simple proof φ_i ($i = 1,\ldots,m$) of h_i from A_i has no more than $\mathsf{deg}(h_i)^2$ branches, and the number of sequents of a branch in φ_i is bounded by $2 \cdot \mathsf{deg}(h_i)$. Similarly, any simple proof φ of f from A_m has no more than $\mathsf{deg}(f)^2$ branches, and the number of sequents of a branch in φ is bounded by $2 \cdot \mathsf{deg}(f)$. Therefore, the number of sequents in φ_i and φ is bounded by $2 \cdot \mathsf{deg}(h_i)^3$ and $2 \cdot \mathsf{deg}(f)^3$, respectively. So, we get the following estimate (with $m < \mathsf{deg}(f)$ and $\mathsf{deg}(h_i) < \mathsf{deg}(f)$):

$$\begin{aligned}
\#seq(\psi) &= \#seq(\theta(\varphi_1)) + \cdots + \#seq(\theta(\varphi_m)) + \#seq(\theta(\varphi)) \\
&= \#seq(\varphi_1) + \cdots + \#seq(\varphi_m) + \#seq(\varphi) \\
&< 2 \cdot m \cdot \mathsf{deg}(f)^3 + 2 \cdot \mathsf{deg}(f)^3 \\
&\leq 2 \cdot \mathsf{deg}(f)^4.
\end{aligned}$$

This concludes the proof of the theorem. ■

An immediate consequence is that the decision problem whether $\vdash f$ has a cut-free sequence proof in GOL is in NP, i.e., the problem can be solved by a nondeterministic Turing machine running in polynomial time. We show in the next subsection that this problem is actually in P $\subseteq$ NP, where P is the class of all problems solvable in deterministic polynomial time.

Recall that the procedure `prove-cfs(T, X)` searches backwards for a *simple* tree proof of the input sequent T with non-logical axioms from X. The search space of `prove-cfs` is exponential in the worst case. Let $A(0, p, \circ)$ be p_0, and let $A(n, p, \circ)$ be $p_n \circ A(n-1, p, \circ)$ for $n > 0$ and $\circ \in \{\wedge, \vee\}$. Furthermore, let $S_{q,r}$ denote the sequent $A(q, a, \wedge) \ \vdash \ A(r, b, \vee)$ and consider $S_{n,n}$. Obviously, $\mathsf{weight}(A(n, p, \circ)) = 2n + 1$. Due to the subformula property and the search for a simple proof (in `prove-cfs`), α–rules are the only applicable inferences. Without optimizations, `prove-cfs` has to exhaust all possible combinations of applications of α–rules resulting in an exponential potential search space in the worst case.

THEOREM 18. *For $n > 0$, the number of sequents generated by procedure* `prove-cfs` *in a proof attempt of $S_{n,n}$ is not smaller than 2^{2n-1}.*

PROOF. More generally, we show by induction on $k(q, r) = q + r$ that, for $q, r > 0$,

($*$) the number of sequents generated by `prove-cfs` in a proof attempt of $S_{q,r}$ is not smaller than 2^{q+r-1}.

INDUCTION BASIS. Assume $k(q, r) = 2$. For the sequent $A(1, a, \wedge) \ \vdash \ A(1, b, \vee)$, there are (at least) two sequents, $A(0, a, \wedge) \ \vdash \ A(1, b, \vee)$ and $A(1, a, \wedge) \ \vdash \ A(0, b, \vee)$, which are generated by `prove-cfs`.

INDUCTION STEP. Assume $k(q, r) = l > 2$, and suppose that for all q', r' with $k(q', r') < l$, ($*$) holds. There are at least two sequents which have to be generated by `prove-cfs` in an attempt to prove $T_{q,r}$, namely $T_{q-1,r}$ and $T_{q,r-1}$. The induction hypothesis yields 2^{q+r-2} many sequents for each case. Hence, the number of sequents generated by `prove-cfs` in a proof attempt of $T_{q,r}$ is not smaller than 2^{q+r-1}. The desired result follows. ■

For the sequent $S_{n,n}$, the size of the search space can be decreased if we optimize the search procedure. Since α–rules can be permuted over α–rules, backtracking can be significantly reduced in the course of a proof attempt for $S_{n,n}$, because not all possible sequences of rule applications have to be performed. This optimization fails, however, for sequents for which β–formulae are present as subformulae in the antecedent or the succedent. Consider the following (unprovable) sequent

$$T_{n,n}\colon \quad L_n \ \vdash \ R_n$$

where

$$\begin{aligned} L_n: &\quad (a_0 \vee (b_0 \wedge B_n)) \wedge A_n; \\ R_n: &\quad (b_0 \wedge (a_0 \vee A_n)) \vee B_n; \\ A_n: &\quad a_1 \wedge \ldots \wedge a_n; \\ B_n: &\quad b_1 \vee \ldots \vee b_n. \end{aligned}$$

For $T_{n,n}$, a result similar to Theorem 18 can be proved. Additionally, an exponentially increasing running time (with respect to n) can be empirically observed with an implementation of a simple backward search procedure for weakening-free proofs.

4.3. Forward Proof Search and a Simple Polynomial Decision Procedure

We now discuss the efficiency of forward proof search in GOL. This procedure is sometimes called *inverse (search) method* and is due to Maslov [7] (cf. also [16]). Many researchers consider simple forward search methods as inefficient because they are not goal (or end sequent) oriented. The key feature to get efficient search procedures is to take the (subformulae of the) end sequent into account when instances of axioms and inference rules are generated. Only those instances of inference rules can occur in the search space for a proof of an end sequent which have subformulae of the end sequent (in the right polarity) in their conclusion.

We show that a polynomial decision procedure for GOL can be obtained if a forward search (with the optimizations mentioned above) is applied. Let $\vdash f$ be the end sequent to be proved, let $U(f) = U^+(f) \cup U^-(f)$ be the set of all subformulae occurring in f, where $U^p(f)$ are the subformulae of f having polarity $p \in \{+, -\}$, and let $c(f)$, $c^+(f)$, and $c^-(f)$ be the cardinalities of $U(f)$, $U^+(f)$, and $U^-(f)$, respectively. Furthermore, let $s(f)$ be the number of possible sequents which can be build from subformulae of f. Then:

$$s(f) = (c^+(f) + 1) \cdot (c^-(f) + 1) + (c^+(f))^2 + (c^-(f))^2 \leq 6(c(f))^2.$$

The product is the number of sequents of the form $a \vdash b$, $a \vdash$, or $\vdash b$ together with $\vdash$, whereas the last two summands are the number of sequents of the form $\vdash a, b$ or $a, b \vdash$, respectively. Rule instances are ordered pairs or triples of the form $T_1 \mid T$ and $T_1, T_2 \mid T$ for unary and binary rules, respectively. A rule instance *for* a formula f is a rule instance with f as its principal or weakening formula. Let $r_1(f)$ be the cardinality of the set of all unary rule instances for subformulae of f, and let $r_2(f)$ be the cardinality of the set of all binary rule instances for subformulae of f. Obviously, $r_1(f) \leq s(f)^2$ and $r_2(f) \leq s(f)^3$. Then, define

$$\begin{aligned} S_0 &= \{\, a \vdash a \mid a \in U^+(f) \cap U^-(f) \,\} \text{ and} \\ S_n &= S_{n-1} \cup \{\, T \mid T \text{ is consequence of a rule instance with} \\ &\qquad\qquad \text{premises in } S_{n-1} \,\}. \end{aligned}$$

Since any rule application introduces at least one connective or atom, n can be bounded by $c(f)$. Moreover, any S_i has no more than $s(f)$ elements. The total number of sequents which can be build in each step is $r_1(f) + r_2(f) \leq s(f)^2 + s(f)^3$. Duplicate sequents are removed immediately because S_i is a *set* of sequents. The total number of generated sequents is bounded by $(s(f)^2 + s(f)^3) \cdot c(f) \leq 252(c(f))^7$.

THEOREM 19. *The problem whether a sequent* $\vdash f$ *is provable in* GOL *is polynomially decidable.*

This result implies that the word problem for free ortholattices is solvable in polynomial time. We remark that a similar consequence is already evident from results given by Skolem in 1920 [13], based on different grounds. More specifically, Skolem showed that the universal first-order theory of lattices is decidable, from which a polynomial decision procedure for finitely presented lattices can be derived. However, as pointed out in [4], this result of Skolem went largely unnoticed by lattice theorists, until it was resurrected by Stan Burries in his studies of the history of logic.

5. Conclusion

We compared three different proof-search strategies for a cut-free sequent calculus for orthologic, namely backward search, a combined forward and backward approach, and forward search. We estimated proof length and search-space size for the backward strategy and the mixed strategy, and showed that forward search yields a polynomial decision procedure for free ortholattices. Our results can be summarized as follows ("polynomial" means that the respective quantity is *always* polynomial in the degree of the formula to be proved, and "exponential" means that there are classes of formulae for which the respective quantity is *always* exponential):

	backward search	mixed strategy	forward search
proof length	exponential	polynomial	polynomial
search space	exponential	exponential	polynomial

Since GOL formalizes orthologic, the question whether a formula is provable in this logic has also a polynomial decision procedure. Furthermore, the technique described in this paper can be used to get similar procedures for other logics as well. For instance, basic logic [3, 11] can be formulated as a cut-free Gentzen calculus based on 3-restricted sequents (if we are interested in proving formulae). Moreover, for other algebraic structures closely related to ortholattices (like, e.g., ascensive grids [12]), polynomial decision procedures for the usual word problems for these structures can be obtained by our approach.

From a more practical point of view, automated proof procedures for (minimal) quantum logic have gained some attention because they can be used as a "calculator" by physicists working in the field of quantum logic. McCune reported in [8] that he was asked (by the authors of [9]) to check whether three rather complicated formulae, E_1, E_2, and E_3, are provable in orthologic. McCune used OTTER and MACE to solve this task, which took 15 minutes for the unprovable formula E_1, 4 seconds for the provable formula E_2, and 22 hours for the provable formula E_3. With our polynomial decision procedure implemented in C, all these formulae together are decided (on comparable hardware) within 1 second.

Acknowledgements

The authors would like to thank G. Sambin for discussions on basic logic, R. Freese for pointing out relevant literature, and the anonymous referees for helpful suggestions.

References

[1] G. Birkhoff and J. v. Neumann. The Logic of Quantum Mechanics. *Ann. Math.*, 37:823–843, 1936.

[2] U. Egly and S. Schmitt. On Intuitionistic Proof Transformations, Their Complexity and Application to Constructive Program Synthesis. *Fundamenta Informaticae*, 39:59–83, 1999.

[3] C. Faggian and G. Sambin. From Basic Logic to Quantum Logics with Cut-Elimination. *International Journal of Theoretical Physics*, 12, 1997.

[4] R. Freese, J. Ježek, and J. B. Nation. *Free Lattices*, volume 42 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., 1995.

[5] G. Kalmbach. *Orthomodular Lattices*. Academic Press, 1983.

[6] S. C. Kleene. Permutability of Inferences in Gentzen's Calculi LK and LJ. *Memoirs of the AMS*, 10:1–26, 1952.

[7] S. Yu. Maslov. An Inverse Method for Establishing Deducibility in the Classical Predicate Calculus. *Doklady Akademii Nauk SSSR*, 159:17–20, 1964.

[8] W. McCune. Automatic Proofs and Counterexamples for some Ortholattice Examples. *Information Processing Letters*, 65:285–291, 1998.

[9] M. Pavičić and N. Megill. Binary Orthologic with Modus Ponens is Either Orthomodular or Distributive. *Helv. Phys. Acta*, 71, 1998.

[10] D. A. Plaisted and Y. Zhu. *The Efficiency of Theorem Proving Strategies*. Computational Intelligence. Vieweg, Braunschweig, 1997.

[11] G. Sambin, G. Battilotti, and C. Faggian. Basic Logic: Reflection, Symmetry, Visibility. *Journal of Symbolic Logic*, 65(3):979–1013, 2000.

[12] J. Schulte Mönting. Cut Elimination and Word Problems for Varieties of Lattices. *Algebra Universalis*, 12:290–321, 1981.

[13] T. Skolem. Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze nebst einem Theoreme über dichte Mengen. *Skrifter utgit av Videnskapsselskapet i Kristiania I, Matematisk-naturvidenskabelig klasse*, 4:1–36, 1920.

[14] R. Smullyan. *First-Order Logic*. Springer Verlag, 1968. Second Printing 1971.

[15] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.

[16] A. Voronkov. Theorem Proving in Non-Standard Logics Based on the Inverse Method. In D. Kapur, editor, *Proc. of the 11th International Conference on Automated Deduction (CADE-11)*, pages 648–662. Springer Verlag, 1992.

Uwe Egly
Institut für Informationssysteme 184/3
Technische Universität Wien
Favoritenstraße 9–11
A-1040 Vienna, Austria
uwe@kr.tuwien.ac.at

Hans Tompits
Institut für Informationssysteme 184/3
Technische Universität Wien
Favoritenstraße 9–11
A-1040 Vienna, Austria
tompits@kr.tuwien.ac.at