No complete linear term rewriting system for propositional logic

Anupam Das, Lutz Straßburger

No complete linear term rewriting system for propositional logic

Anupam Das[1] **and Lutz Straßburger**[2]

1 **ENS de Lyon and Inria, France,** anupam.das@ens-lyon.fr
2 **INRIA Saclay – Île-de-France, France, and Laboratoire d'Informatique (LIX), Palaiseau, France,** lutz@lix.polytechnique.fr

Abstract

Recently it has been observed that the set of all sound linear inference rules in propositional logic is already **coNP**-complete, i.e. that every Boolean tautology can be written as a (left- and right-) linear rewrite rule. This raises the question of whether there is a rewriting system on linear terms of propositional logic that is sound and complete for the set of all such rewrite rules. We show in this paper that, as long as reduction steps are polynomial-time decidable, such a rewriting system does not exist unless **coNP** = **NP**.

We draw tools and concepts from term rewriting, Boolean function theory and graph theory in order to access the required intermediate results. At the same time we make several connections between these areas that, to our knowledge, have not yet been presented and constitute a rich theoretical framework for reasoning about linear TRSs for propositional logic.

1998 ACM Subject Classification F.4 Mathematical Logic and Formal Languages

Keywords and phrases Linear rules, Term rewriting, Propositional logic, Proof theory, Deep inference

Digital Object Identifier 10.4230/LIPIcs.RTA.2015.127

1 Introduction

Linear inferences, as defined in [9] and also known as "balanced" tautologies (e.g. in [24]) or linear rules (e.g. in deep inference [2], [3] [12], [13]), are sound implications in classical propositional logic (CPL), each of whose variables occur exactly once in both the premiss and the conclusion. From the point of view of term rewriting they are rewrite rules that are non-erasing, left- and right-linear, and such that the Boolean function computed by the left hand side logically implies that computed by the right hand side.[1]

The reason why this is an interesting set of rewrite rules is due to the observation that *all* Boolean tautologies can be written in this form, by means of a polynomial-time translation [24]. In this work we ask whether one can derive all of CPL *internally* to this fragment; i.e. is there a set of linear inferences (satisfying certain conditions) that is complete, under term rewriting, for the set of all linear inferences (denoted L henceforth)?

It was previously shown that such a set could not be finite [9, 24], via an encoding of instances of the pigeonhole principle as linear inferences. However in this work we consider any system whose reduction steps can be checked efficiently, i.e. form a polynomial-time decidable set. The motivation behind this generality is that such a set would constitute

[1] For generality and ease of presentation, we later drop the "non-erasing" criterion for linear inferences in this work.

26th International Conference on Rewriting Techniques and Applications (RTA'15).
Editor: Maribel Fernández; pp. 127–142

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

a sound and complete *proof system*[2] for CPL with no meaningful duplication, creation or destruction of formulae,[3] in stark contrast to the traditional approach of *structural* proof theory, based on rules exhibiting precisely such behaviour.

In this work we show that no such linear system exists, unless **coNP** = **NP**. In a little more detail, we show that any such system would admit a derivation of each valid linear inference of polynomial length (and so polynomial-size, by linearity). This would imply that **coNP** is contained in **NP** as follows:

1. There is an **NP**-algorithm for L: simply guess the correct derivation in some sound and complete linear system.
2. Since $TAUT$ is polynomial-time reducible to L there is also an **NP**-algorithm for $TAUT$.
3. By the Cook-Levin theorem that SAT is **NP**-complete [5, 20], we have that $TAUT$ is **coNP**-complete, and so there is a **NP**-algorithm for **coNP**.

Functions computed by linear terms of CPL have been studied in Boolean function theory, and more specifically circuit complexity, for decades, where they are called "read-once functions" (e.g. in [7]).[4] They are closely related to positional games (first mentioned in [15]) and have been used in amplification of approximation circuits, (first in [26], more generally in [11]) as well amongst other areas. Their equivalence classes under associativity and commutativity of $\wedge$ and $\vee$ can also be represented as the set of "cographs", or "P_4-free" graphs, essentially what we call "relation webs" in this work, following [13] and [23].

In this paper we work in both the Boolean function theoretic and graph theoretic settings, as well as that of term rewriting, presenting novel interplays between them. In particular, the proof of our main result, Thm. 30, crucially uses concepts from all three settings, which we hope is clear from the exposition.

We develop connections and applications of concepts about read-once functions, e.g., Prop. 13 and Thm. 19, that seem to be novel, as results on such concepts have appeared before only in the setting of isolated Boolean functions, rather than in a logical setting where we care futhermore about logical relations between functions, in particular, when one function implies another.

From the point of view of rewriting theory, logic has always been a motivational domain of applications. For example, "tautology checking" is used as one of the three motivating examples in the Terese book, *Term Rewriting Systems* [25]. Rewriting systems for propositional logic can be recovered from axiom systems for Boolean algebras and Boolean rings, e.g. as in [10] and [18]. While this area has been well studied, our 'deep inference' style approach is more general in scope due to our handling of negation: by dealing with terms in negation normal form we can reason about systems that are not purely equational, but consisting of arbitrary sound rules, due to the absence of negative contexts. Notice that complete equational theories for CPL cannot possibly be linear, e.g. due to Thm. 9, and so such a question is only pertinent in our more general setting.

The organisation of this paper is as follows. In Sects. 2 and 3 we present the basics on term rewriting in CPL and usual Boolean interpretations. In Sect. 4 we define relation webs and give graph-theoretic versions of various logical concepts. In Sect. 5 we present a normal form of linear derivations, which we ultimately use in Sect. 6 to prove our main

[2] Recall that proof systems are usually required to be efficiently (i.e. polynomial-time) checkable [6].

[3] The only duplication would occur in the reduction from $TAUT$ to L where its complexity is bounded by some fixed polynomial.

[4] These have been studied in various forms and under different names. The first appearance we are aware of is in [4], and also the seminal paper of [14] characterising these functions. The book we reference presents an excellent and comprehensive introduction to the area.

result, polynomial-time weak normalisation. In Sect. 7 we apply previous results to deduce and conjecture forms of *canonicity* of certain linear rules prominent in deep inference proof theory, and in Sect. 8 we make some concluding remarks.

2 Preliminaries on rewriting theory

We generally work in the first-order term rewriting setting defined in the Terese textbook, *Term Rewriting Systems* [25]. We will, in fact, use the same notation for all symbols except the connectives, for which we use more standard notation from proof theory. In particular we will use $\bot$ and $\top$ for the truth constants, reserving 0 and 1 for the inputs and outputs of Boolean functions, introduced later.

We adopt two particular conventions which differ from usual definitions in the literature:

1. A TRS is usually defined as an arbitrary set of rewrite rules. Here we insist that the set of instances of these rules, or reduction steps, is polynomial-time decidable.
2. Rewriting modulo an equivalence relation usually places no restriction on the source and target of a reduction step. Here we insist that they must be *distinct* modulo the equivalence relation.

The motivation for (1) is that we wish to be as general as possible without admitting trivial results. If we allowed all sets then a complete system could be specified quite easily indeed. Furthermore, that an inference rule is easily or feasibly checkable is a usual requirement in proof theory, and in proof complexity this is formalised by the same condition (1) on inference rules, essentially due to the fact that *TAUT* is **coNP**-complete. Perhaps it would be better to call these 'polynomial' TRSs, however we drop this prefix for presentation reasons throughout this article.

The motivation for (2) is that we fundamentally care about weak normalisation, e.g. Cor. 31, but it will be useful to make statements resembling strong normalisation under this notion of rewriting modulo, e.g. Thm. 30. All the equivalence relations we will work with are polynomial-time decidable, and so this convention is consistent with (1). The same notion of rewriting modulo was also used in previous work [9].

Propositional logic in the term rewriting setting

Our language is built from the connectives $\bot, \top, \wedge, \vee$ and a set *Var* of propositional variables, typically denoted $x, y, z, \ldots$. The set *Var* is equipped with an involution (i.e. self-inverse function) $\bar{\cdot} : \mathit{Var} \to \mathit{Var}$. We call $\bar{x}$ the *dual* of x and, for each pair of dual variables, we arbitrarily choose one to be *positive* and the other to be *negative*.

The set *Ter* of formulae, or *terms*, is built freely from this signature in the usual way. Terms are typically denoted by $s, t, u, \ldots$, and term and variable symbols may occur with superscripts and subscripts if required.

In this setting $\top$ and $\bot$ are considered the constant symbols of our language. We say that a term t is *constant-free* if $\top$ and $\bot$ do not occur in t.

We do not include a symbol for negation in our language. This is due to the fact that soundness of a rewrite step is only preserved under *positive* contexts. Instead we simply consider terms in negation normal form (NNF), which can be generated for arbitrary terms from positive and negative variables by the De Morgan laws:

$$\bar{\top} = \bot \qquad \bar{\bot} = \top \qquad \bar{\bar{x}} = x \qquad \overline{A \vee B} = \bar{A} \wedge \bar{B} \qquad \overline{A \wedge B} = \bar{A} \vee \bar{B}$$

We say that a term is *negation-free* if it does not contain any negative variables. We write $\mathit{Var}(t)$ to denote the set of variables occurring in t. We say that a term t is *linear* if, for

each $x \in Var(t)$, there is exactly one occurrence of x in t. The *size* of a term t, denoted $|t|$, is the total number of variable and function symbols occurring in t. A *substitution* is a mapping $\sigma\colon Var \to Ter$ from the set of variables to the set of terms such that $\sigma(x) \neq x$ for only finitely many x. The notion of substitution is extended to all terms, i.e. a map $Ter \to Ter$, in the usual way. A (one-hole) *context* is a term with a single 'hole' $\Box$ occurring in place of a subterm. For example consider the following:

$$C_1[\Box] := y \wedge (z \vee \Box) \qquad C_2[\Box] := \Box \vee (w \wedge x) \qquad C_3[\Box] := (w \wedge x) \vee (y \wedge (z \vee \Box))$$

We may write $C_i[t]$ to denote the term obtained by replacing the occurrence of $\Box$ in $C_i[\Box]$ with t. We may also replace holes with other contexts to derive new contexts. For example, notice that $C_3[\Box]$ is equivalent, modulo commutativity of $\vee$, to $C_2[C_1[\Box]]$.

▶ **Definition 1** (Rewrite rules)**.** A *rewrite rule* is an expression $l \to r$, where l and r are terms. We write $\rho : l \to r$ to express that the rule $l \to r$ is called ρ. In this rule we call l the left hand side (LHS) of ρ, and r the right hand side (RHS).

We say that ρ is *left-linear* (resp. *right-linear*) if l (resp. r) is a linear term. We say that ρ is *linear* if it is both left- and right-linear.

We write $s \underset{\rho}{\to} t$ to express that $s \to t$ is a *reduction step* of ρ, i.e. that $s = C[\sigma(l)]$ and $t = C[\sigma(r)]$ for some substitution σ and context $C[\Box]$.

▶ **Definition 2** (Term rewriting systems)**.** A *term rewriting system* (TRS) is a set of rewrite rules whose reduction steps are decidable in polynomial time. The *one-step* reduction relation of a TRS R is $\underset{R}{\to}$, where $s \underset{R}{\to} t$ if $s \underset{\rho}{\to} t$ for some $\rho \in R$.

A *linear* (term rewriting) system is a TRS, all of whose rules are linear.

▶ **Definition 3** (Derivations)**.** A *derivation* under a binary relation $\underset{R}{\to}$ on Ter is a sequence $\pi : t_0 \underset{R}{\to} t_1 \underset{R}{\to} \cdots \underset{R}{\to} t_l$. In this case we say that π has *length* l.

We also write $\overset{*}{\underset{R}{\to}}$ to denote the reflexive transitive closure of $\underset{R}{\to}$.

▶ **Definition 4** (Rewriting modulo)**.** For an equivalence relation $\sim$ on Ter and a TRS R, we define the relation $\underset{R/\sim}{\longrightarrow}$ by $s \underset{R/\sim}{\longrightarrow} t$ if there are s', t' such that $s \sim s' \underset{R}{\to} t' \sim t$ such that $s' \nsim t'$. An $R/\sim$ derivation is also called an R-derivation *modulo* $\sim$.

In this work we consider linear equivalence relations, like associativity and commutivity of $\wedge$ and $\vee$, denoted AC. We also have linear equations for the truth constants, the system U:

$$x \vee \bot = x = \bot \vee x \quad , \quad x \wedge \top = x = \top \wedge x \quad , \quad \top \vee \top = \top \quad , \quad \bot \wedge \bot = \bot$$

We denote by ACU the combined system of AC and U. For certain reasons it will also be useful to consider the system U' that extends U by the following rules:[5]

$$x \vee \top = \top = \top \vee x \quad , \quad x \wedge \bot = \bot = \bot \wedge x$$

We denote by ACU' the combined system of AC and U'. It turns out that this equivalence relation relates precisely those linear terms that compute the same Boolean function, as we discuss in the next section.

▶ Remark (On the use of '$\to$')**.** To avoid possible confusion, notice that we are using the $\to$ symbol both for a formal expression, e.g. the rewrite rule $s \to t$, and with annotations to express a relation between two terms, e.g. the reduction step $s \underset{\rho}{\to} t$.

[5] Notice that these are not linear in the sense of [9], but are considered linear in our more general setting.

3 Preliminaries on Boolean functions

In this section we introduce the usual Boolean function models for terms of propositional logic.

A *Boolean function* on a (finite) set of variables $X \subseteq Var$ is a map $f\colon \{0,1\}^X \to \{0,1\}$. We identify $\{0,1\}^X$ with $\mathcal{P}(X)$, the powerset of X, i.e. we may specify an argument of a Boolean function by the subset of its variables assigned to 1.

A little more formally, a function $\nu : X \to \{0,1\}$ is specified by the set X_ν it indicates, i.e. $x \in X_\nu$ just if $\nu(x) = 1$. For this reason we may quantify over the arguments of a Boolean function by writing $Y \subseteq X$ rather than $\nu \in \{0,1\}^X$, i.e., we write $f(Y)$ to denote the value of f if the input is 1 for the variables in Y and 0 for the variables in $X \setminus Y$. Similarly, we write $f(\overline{Y})$ for the value of f when the variables in Y are 0 and the variables in $X \setminus Y$ are 1.

3.1 Boolean semantics of terms

A term t computes a Boolean function $\{0,1\}^{Var(t)} \to \{0,1\}$ in the usual way.

For Boolean functions $f, g : \{0,1\}^X \to \{0,1\}$ we write $f \leq g$ if $\forall Y \subseteq X$ we have that $f(Y) \leq g(Y)$. Notice that the following can easily be shown to be equivalent:

1. $f \leq g$.
2. $f(Y) = 1 \Rightarrow g(Y) = 1$.
3. $g(Y) = 0 \Rightarrow f(Y) = 0$.

We also write $f < g$ if $f \leq g$ but $f(Y) \neq g(Y)$ for some $Y \subseteq X$.

▶ **Definition 5** (Soundness)**.** We say that a rewrite rule $s \to t$ is *sound* if s and t compute Boolean functions f and g, respectively, such that $f \leq g$. We say that a TRS is sound if all its rules are sound. A *linear inference* is a sound linear rewrite rule. The set of all linear inferences is denoted by L.

▶ Notation 6. To switch conveniently between the settings of terms and Boolean functions, we freely interchange notations, e.g. writing $s \leq t$ to denote that $s \to t$ is sound, and saying $f \to g$ is sound when $f \leq g$.

▶ Remark. We point out that, here, our definition of "linear inference" differs slightly from that occurring in [9]. Namely, we insist only that the LHS and RHS are linear, but not necessarily that they have the same variable set. We choose this more general definition since it seems more natural in the setting of term rewriting. Furthermore, since it is indeed a more general definition, the same result carries over for the previous notion too. In fact, in later sections, we will restrict our attention to the former notion of linear inference due to the fact that any erasure or introduction[6] of variables in a linear rule would constitute what we call a "triviality" in Section 5, where we also elaborate on and address this issue.

Finally we give one of the key motivations for this work, essentially from [24]:

▶ **Proposition 7.** L *is* **coNP**-*complete.*

This result is the reason, from the point of proof theory, why one might restrict attention to only linear inferences at all: every Boolean tautology can be written as a linear inference. As we can see from the proof that follows, the translation is not very complicated. However,

[6] We point out that in many settings, indeed in [25], a rewrite rule is not allowed to introduce new variables. I.e. all variables occurring on the RHS must also occur in the LHS. In our setting it seems more natural and symmetric to allow such behaviour and, again, this yields a more general result.

it does induce an at most quadratic blowup in size from an input tautology to a linear inference.

We include a proof below, for completeness, and since the statement here differs slightly from that in [24].

Proof of Proposition 7. That L is in **coNP** is due to the fact that checking soundness of a rewrite rule $s \to t$ can be reduced to checking validity of the formula $\bar{s} \vee t$. To prove **coNP**-hardness, we can reduce validity of general tautologies to soundness of linear rewrite rules. We let t' be the term obtained from t (which is assumed to be in NNF) by doing the following for each positive variable x: let n be the number of occurrences of x in t, and let m be the number of occurrences of $\bar{x}$ in t. If $n = 0$ replace every occurence of $\bar{x}$ by $\bot$, and if $m = 0$ replace every occurrence of x by $\bot$. Otherwise, introduce $2mn$ fresh (positive) variables $x'_{i,j}, x''_{i,j}$ for $1 \leq i \leq n$ and $1 \leq j \leq m$. Now, for $1 \leq i \leq n$, replace the i^{th} occurrence of x by $x'_{i,1} \vee \ldots \vee x'_{i,m}$ and, for $1 \leq j \leq m$, replace the j^{th} occurrence of $\bar{x}$ by $x''_{1,j} \vee \ldots \vee x''_{n,j}$.

Now t' is a linear term (without negation), and its size is quadratic in the size of t. Let s' be the conjunction of all pairs $x' \vee x''$ of variables introduced in the construction of t'. Clearly $Var(s') = Var(t')$ and s' is also a linear term of the same size as t'. Furthermore, t is a tautology if and only if $s' \to t'$ is sound. To see this, let s'' and t'' be obtained from s' and t', respectively, by replacing each x'' by $\bar{x}'$. Then s'' always evaluates to 1, and t'' is a tautology if and only if t is a tautology. ◀

3.2 Read-once functions and linear terms

Linear terms compute what are known as "read-once" Boolean functions, and we survey some of their theory in this section.

▶ **Definition 8** (Read-once functions)**.** A Boolean function is *read-once* if it is computed by some linear term (of propositional logic).

It is not exactly clear when the following result first appeared, although we refer to a discussion in [7] where it is stated that results directly implying this were first mentioned in [19]. The result also occurs in [14], and is generalised to certain other bases in [16] and [17].

▶ **Theorem 9.** *Constant-free negation-free linear terms compute the same (read-once) Boolean function if and only if they are equivalent modulo AC.*

A proof of this can easily be derived from results in Sect. 4, by the presentation of equivalence classes modulo AC as relation webs and the graph-theoretic definition of soundness.

The following consequences of Thm. 9 appear in [9], where detailed proofs may be found.

▶ **Corollary 10.** *Negation-free linear terms compute the same (read-once) Boolean function if and only if they are equivalent modulo ACU'.*

Proof idea. The result essentially follows from the observation that every negation-free term is ACU'-equivalent to $\bot$, $\top$ or a unique constant-free term [8]. ◀

▶ **Corollary 11.** *Any sound negation-free linear TRS, modulo ACU', is terminating in exponential-time.*

Proof. The result follows by Boolean semantics and the preceding corollary: each consequent term must compute a distinct Boolean function that is strictly bigger, under $\leq$, and the graph of $\leq$ has length 2^n, where n is the number of variables in the input term. ◀

3.3 Minterms and maxterms

In this section we restrict our attention to *monotone* Boolean functions, i.e., those functions $f\colon \{0,1\}^X \to \{0,1\}$ such that $Y \subseteq Y' \subseteq X$ implies $f(Y) \leq f(Y')$. We point out the observation that negation-free terms compute monotone Boolean functions.

Minterms and maxterms correspond to minimal DNF and CNF representations, respectively, of a monotone Boolean function. We refer the reader to [7] for an introduction to their theory. In this work we use them in a somewhat different way to Boolean function theory, in that we devise definitions of logical concepts, such as soundness and, later in Sect. 5, what we call "triviality". The reason for this is to take advantage of the purely function-theoretic results stated in this section (e.g. Gurvich's Thm. 14 below) to derive our main results.

▶ **Definition 12.** Let f be a monotone Boolean function on a variable set X. A set $Y \subseteq X$ is a *minterm* (resp. *maxterm*) for f if it is a minimal set such that $f(Y) = 1$ (resp. $f(\overline{Y}) = 0$). The set of all minterms (resp. maxterms) of f is denoted $MIN(f)$ (resp. $MAX(f)$).

Using these notions, we can now give an alternative definition of soundness.

▶ **Proposition 13** (Soundness via minterms or maxterms)**.** *For monotone Boolean functions f, g on the same variable set, the following are equivalent:*

1. $f \leq g$.
2. $\forall S \in MIN(f).\ \exists S' \in MIN(g).\ S' \subseteq S$.
3. $\forall T \in MAX(g).\ \exists T' \in MAX(f).\ T' \subseteq T$.

Proof. 1 $\implies$ 2. Let $f \leq g$ and suppose there is an $S \in MIN(f)$ such that there is no $S' \in MIN(g)$ with $S' \subseteq S$. Then $f(S) = 1$ and $g(S) = 0$, contradicting $f \leq g$.

2 $\implies$ 1. Let Y be such that $f(Y) = 1$. Then there is a minterm $S \in MIN(f)$ with $S \subseteq Y$. By 2, there is a minterm $S' \in MIN(g)$ with $S' \subseteq S$, and therefore $S' \subseteq Y$. Therefore $g(Y) = 1$, by monotonicity, and so $f \leq g$.

1 $\implies$ 3 and 3 $\implies$ 1 are proved similarly. ◀

The following classical result is due to Gurvich in [14], but has appeared in various presentations. In particular, the proof appearing in [7] uses the notion of *cooccurrence* graph, to which our "relation webs" in the next section essentially amounts.[7]

▶ **Theorem 14** (Gurvich)**.** *A monotone Boolean function f is read-once if and only if*

$$\forall S \in MIN(f).\ \forall T \in MAX(f).\ |S \cap T| = 1 \quad .$$

4 Relation webs

In this section we restrict our attention to negation-free constant-free linear terms. It will be useful for us to consider not only the Boolean semantics of terms but also their syntactic structure, in the form of *relation webs* [13, 23]. It turns out that many of the same concepts that we have seen in the previous sections can be defined in this setting and the interplay between the two settings is something that we will take advantage of in later results.

[7] Indeed, by the end of Sect. 4 we will have developed enough technology to give a self-contained proof of this result, but that is beyond the scope of this work.

4.1 Preliminary material

We make use of *labelled graphs* with their standard terminology. For a graph G we denote its *vertex set* or set of *nodes* as $V(G)$, and the set of its *labelled edges* as $E(G)$.

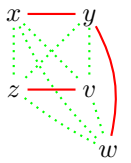
For graphs G and H such that $V(G) \subseteq V(H)$, we say "G in H" to assert that G is an (induced) subgraph[8] of H. In particular we say "$x \overset{\star}{\text{——}} y$ in G" to express that the edge $\{x, y\}$ is labelled $\star$ in the graph G.

We say that a set $X \subseteq V(G)$ is a *$\star$-clique* if every pair $x, y \in X$ has a $\star$-labelled edge between them. A *maximal* $\star$-clique is a $\star$-clique that is not contained in any larger $\star$-clique.

Analysing the term tree of a negation-free constant-free linear term, notice that for each pair of variables x, y, there is a unique connective $\star \in \{\wedge, \vee\}$ at the root of the smallest subtree containing the (unique) occurrences of x and y. Let us call this the *first common connective* of x and y in t.

▶ **Definition 15** (Relation webs)**.** The *(relation) web* $\mathcal{W}(t)$ of a constant-free negation-free linear term t is the complete graph whose vertex set is $Var(t)$, such that the edge between two variables x and y is labelled by their first common connective in t.

As a convention we will write x —— y if the edge $\{x, y\}$ is labelled by $\wedge$, and we write x ····· y if it is labelled by $\vee$.

▶ **Example 16.** The term $([x \vee w] \wedge y) \vee (z \wedge v)$ has the relation web 

.

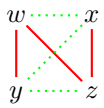
▶ Remark (Labels). We point out that, instead of using labelled complete graphs, we could have also used unlabelled arbitrary graphs, since we have only two connectives ($\wedge$ and $\vee$) and so one could be specified by the lack of an edge. This is indeed done in some settings, e.g. the cooccurrence graphs of [7]. However, we use the current formulation in order to maintain consistency with the previous literature, e.g. [13] and [23], and since it helps write certain arguments, e.g. in Sect. 7, where we need to draw graphs with incomplete information.

One of the reasons for considering relation webs is the following proposition, which allows to reason about equivalence classes modulo AC easily. It follows immediately from the definition and that AC preserves first common connectives.

▶ **Proposition 17.** *Constant-free negation-free linear terms are equivalent modulo AC if and only if they have the same web.*

An important property of webs is that they have no minimal paths of length > 2. More precisely, we have the following proposition:

▶ **Proposition 18.** *A complete $\{\wedge, \vee\}$-labelled graph on X is the web of some negation-free constant-free linear term on X if and only if it contains no induced subgraphs of the form:*



(1)

A proof of this property can be found, for example, in [21], [22], [1], or [13]. It is called *P_4-freeness* or *Z-freeness* or *N-freeness*, depending on the viewpoint. We will make crucial use of it when later reasoning with webs.

[8] In fact, since all graphs we deal with are complete, all subgraphs are implicitly induced.

4.2 Relationships to minterms and maxterms

Essentially one can think of relation webs as a graph-theoretic formulation of minterms and maxterms, as opposed to the set-theoretic formulation earlier, in light of the following result:

▶ **Theorem 19.** *A set of variables is a minterm (resp. maxterm) of a negation-free constant-free linear term t if and only if it is a maximal $\wedge$-clique (resp. maximal $\vee$-clique) in $\mathcal{W}(t)$.*

The proof of this follows easily from the following alternative definition of minterms and maxterms, based on structural induction on a term:

▶ **Proposition 20** (Inductive definition of minterms and maxterms)**.** *Let t be a linear term. A set $S \subseteq Var(t)$ is a minterm of t if and only if:*
- *$t = x$ and $S = \{x\}$.*
- *$t = t_1 \vee t_2$ and S is a minterm of t_1 or of t_2.*
- *$t = t_1 \wedge t_2$ and $S = S_1 \cup S_2$ where each S_i is a minterm of t_i.*

Dually, a set $T \subseteq Var(t)$ is a maxterm of t if and only if:
- *$t = x$ and $T = \{x\}$.*
- *$t = t_1 \vee t_2$ and $T = T_1 \cup T_2$ where each T_i is a maxterm of t_i.*
- *$t = t_1 \wedge t_2$ and T is a maxterm of t_1 or of t_2.*

5 Dealing with constants, negation, erasure and trivialities

In this section we show that we need not deal with linear rules that contain constants or negation when looking for a complete linear system, or linear rules all of whose variables do not occur on both sides. The fundamental concept here is that of "triviality", first introduced in [9] as "semantic triviality". This turns out also to be precisely the concept which allows us to polynomially restrict the length of linear derivations for our main result in Sect. 6.

Many of the following results appeared in [9], so we present only brief arguments here.

5.1 Triviality

The idea behind triviality of a variable in some linear inference is that the inference is "independent" of the behaviour of that variable.

▶ **Definition 21** (Triviality)**.** Let f and g be Boolean functions on a set of variables X, and let $x \in X$. We say $f \to g$ is *trivial* at x if for all $Y \subseteq X$, we have $f(Y \cup \{x\}) \leq g(Y \setminus \{x\})$. We say simply that f is 'trivial' if it is trivial at one of its variables.

▶ Remark (Hereditariness of triviality). Notice that the triviality relation is somehow hereditary: if a sound sequence $f_0 \to f_1 \to \ldots \to f_l$ of Boolean functions is trivial at some point $f_i \to f_{i+1}$ for $0 \leq i < l$ then $f_1 \to f_n$ is trivial. However the converse does not hold: if the first and last function of a sound sequence constitutes a trivial pair it may be that there is no local triviality in the sequence. E.g. the endpoints of the derivation,

$$(w \wedge x) \vee (y \wedge z) \to [w \vee y] \wedge [x \vee z] \to w \vee x \vee (y \wedge z)$$

form a pair that is trivial at w (or trivial at x), but no local step witnesses this. In these cases we call the sequence *globally* trivial. This notion is fundamental later in Lemma 33, on which our main result crucially relies.

In a similar way as we could express soundness with minterms or maxterms in Prop. 13, we can also define triviality with minterms or maxterms.

▶ **Proposition 22.** *The following are equivalent:*
1. $f \to g$ *is trivial at* x.
2. $\forall S \in MIN(f).\ \exists S' \in MIN(g).\ S' \subseteq S \setminus \{x\}$.
3. $\forall T \in MAX(g).\ \exists T' \in MAX(f).\ T' \subseteq T \setminus \{x\}$.

Proof. We first show that $1 \implies 2$. Assume $f \to g$ is trivial at x, and let $S \in MIN(f)$. We have $f(S) = 1$, and hence also $f(S \cup \{x\}) = 1$. By way of contradiction assume there is no $S' \in MIN(g)$ with $S' \subseteq S \setminus \{x\}$. Therefore $g(S \setminus \{x\}) = 0$, contradicting triviality at x. Next, we show $2 \implies 1$. For this, let Y be such that $f(Y \cup \{x\}) = 1$. Then there is a minterm $S \in MIN(f)$ with $S \subseteq Y \cup \{x\}$. By 2, there is a minterm $S' \in MIN(g)$ with $S' \subseteq S \setminus \{x\}$. Hence $S' \subseteq Y \setminus \{x\}$. Therefore $g(Y \setminus \{x\}) = 1$, and thus $f \to g$ is trivial at x. To show $1 \implies 3$ and $3 \implies 1$ we proceed analogously. ◀

We now present a series of results illustrating that we need not consider trivial derivations in any linear system containing certain rules. These results are then used to show that constants and negation are similarly unimportant.

▶ **Definition 23.** We define the following rules:

$$\mathsf{s} : x \wedge [y \vee z] \to (x \wedge y) \vee z \quad , \quad \mathsf{m} : (w \wedge x) \vee (y \wedge z) \to [w \vee y] \wedge [x \vee z]$$

We call the former *switch* and the latter *medial* [2].

In what follows we implicitly assume that rewriting is conducted modulo ACU.

▶ **Lemma 24.** *If* s, t *are negation-free linear terms on* $x_1, \ldots, x_n$ *and* $s \leq t$*, then there are terms* s', t', u *such that:*
1. *There are derivations* $s \xrightarrow[\mathsf{s},\mathsf{m}]{*} s' \vee u$ *and* $t' \vee u \xrightarrow[\mathsf{s},\mathsf{m}]{*} t$ *of length* $O(n^2)$.
2. $s' \to t'$ *is sound and nontrivial.*

Proof. See [9]. Briefly, the idea is that u is obtained by repeatedly 'moving aside' trivial variables, using s, m and ACU, until there are no trivialities remaining in $s' \to t'$. ◀

▶ **Theorem 25.** *Let* R *be a complete linear system. If* $s \xrightarrow[R]{*} t$ *then there is an* R*-derivation from* s *to* t *with only* $O(|s|^2)$*-many steps whose redex and contractum constitute a triviality.*

Proof. Apply the lemma above to generate terms s', t', u as above. Since R is complete there must be a derivation of $s' \to t'$, and this cannot contain any trivialities by the hereditariness property (cf. Rmk. 5.1) and the fact that $s' \to t'$ is nontrivial.

Therefore the only steps whose redex and contractum form a trivial pair are those generated by 1 in Lemma 24 above, whence we know that the number of such steps is quadratic in the number of variables. ◀

5.2 Erasing and introducing rules

A left- and right-linear rewrite rule may still erase or introduce variables, i.e. there may be variables on one side that do not occur on the other. However, notice that any such situation must constitute a triviality at such a variable, since the soundness of the step is not dependent on the value of that variable.

▶ **Proposition 26.** *Suppose* $\rho : l \to r$ *is linear, and there is some variable* x *occurring in only one of* l *and* r*. Then* ρ *is trivial at* x.

5.3 Negation

If a (positive) variable x occurs negatively on both sides of a linear rule then $\bar{x}$ can be replaced soundly by x on both sides. Otherwise, if x occurs positively on one side and negatively on the other, it must be that we have a triviality at x.

▶ **Proposition 27.** *For each linear rule ρ either there is a negation-free linear rule that is equivalent to ρ (i.e. with the same reduction steps), or ρ is trivial.*

5.4 Constants

Let us assume in this subsection that terms are negation-free, in light of Prop. 27 above.

Recall that ACU' preserves the Boolean function computed by a term, and that every linear term is equivalent to $\bot$, $\top$ or a unique constant-free linear term.

▶ **Theorem 28.** *Let R be a complete linear system. Then any constant-free nontrivial linear inference $s \to t$ has a constant-free R/ACU'-derivation.*

Proof. By completeness there is an R-derivation of $s \to t$. Now reduce every line by ACU' to a constant-free term or $\bot$ or $\top$ (e.g. as shown in [9]). If some line reduces to $\bot$ or $\top$ and another does not, then $s \to t$ is trivial, and if every line reduces to $\bot$ or every line reduces to $\top$ then the derivation collapses and is no longer constant-free. ◀

5.5 Putting it together

Combining the various results of this section we obtain the following:

▶ **Theorem 29.** *The following are equivalent:*
1. *There is a sound linear system complete for* L.
2. *There is a sound constant-free negation-free nontrivial linear system, whose rules have the same variables on both sides, complete for the set of such inferences.*

6 Main results

In light of Thm. 29 in the previous section, we assume the following throughout this section:

Terms are constant-free, negation-free and linear on a variable set X of size n.

The following is our main result.

▶ **Main Theorem 30.** *For every sequence of terms $s = t_0 < t_1 < \cdots < t_l = t$ we have that:*
1. *$l = O(n^4)$; or,*
2. *$s \to t$ is trivial.*

Before giving a proof, we show how this implies that there is no sound and complete linear system, modulo hardness assumptions.

▶ **Corollary 31.** *If there is a sound and complete linear system, then there is one that has a $O(n^4)$-length derivation for each linear inference on n variables.*

Proof. This follows from Thm. 30, Lemma 24 and Thm. 29. ◀

▶ **Corollary 32.** *There is no sound linear system complete for* L *unless* **coNP** = **NP**.

Proof. L is **coNP**-complete, by Prop. 7, and so Cor. 31 induces an **NP** decision procedure for L for any such system R: guess a correct sequence of R-steps to derive $s \to t$. ◀

In the next section we give the crucial lemma that allows us to obtain a proof of our main theorem. The argument itself is outlined in the section thereafter.

6.1 Critical minterms and maxterms

For this section, let us fix a sequence $f = f_0 < f_1 < \cdots < f_l = g$ of strictly increasing read-once Boolean functions on a variable set X.

Here we show that, unless $f \to g$ is trivial, for each variable $x \in X$ we must be able to associate a minterm S^x of f such that, for any $S \subseteq S^x$ that is a minterm of some f_i, it must be that $S \ni x$. We simultaneously show the dual property for maxterms.

▶ **Lemma 33** (Subset and intersection lemma). *Suppose $f \to g$ is not trivial. For every variable $x \in X$, there is a minterm S^x of f and a maxterm T^x of g such that:*
1. $\forall S_i \in MIN(f_i).S_i \subseteq S^x \implies x \in S_i$.
2. $\forall T_i \in MAX(g_i).T_i \subseteq T^x \implies x \in T_i$.
3. $\forall S_i \in MIN(f_i), \forall T_i \in MAX(g_i).S_i \subseteq S^x, T_i \subseteq T^x \implies S_i \cap T_i = \{x\}$.

Proof. Suppose that, for some variable x no minterm of f has property 1. In other words, for every minterm S^x of f containing x there is some minterm S_i of some f_i that is a subset of S^x yet does not contain x. Since $f_i \to f_l$ is sound for every i we have that, by Prop. 13, for every minterm S^x of f containing x there is some minterm S_l of $f_l = g$ that is a subset of S^x not containing x. I.e. $f \to g$ is trivial, by Prop. 22, which is a contradiction. Property 2 is proved analogously. Finally, Property 3 is proved by appealing to read-onceness. Any such S_i and T_i must contain x by properties 1 and 2, yet their intersection must be a singleton by Thm. 14 since all f_i are read-once, whence the result follows. ◀

We notice that, since some S_i and T_i must exist for all i, by soundness, we can build a chain[9] of such minterms and maxterms preserving the intersection point. For a given derivation, let us call a choice of such minterms and maxterms *critical*.

6.2 Proof of the main result, Thm. 30

Throughout this section let us fix a sound (negation-free constant-free) linear system R, which we assume to contain s, m,[10] whose reduction relation, modulo AC, is $\underset{R}{\longrightarrow}$.

Recall that $s \underset{R}{\longrightarrow} t$ implies that s, t are distinct modulo AC so compute distinct Boolean functions by Thm. 14 and have distinct relation webs. Let us fix a nontrivial R-derivation,

$$\pi \quad : \qquad s = t_0 \underset{R}{\longrightarrow} t_1 \underset{R}{\longrightarrow} \cdots \underset{R}{\longrightarrow} t_l = t$$

Now, let us fix for each $x \in X$ and $0 \leq i \leq l$ choices S^x_i and T^x_i of critical minterms and maxterms, respectively, of t_i, by Lemma 33. I.e. we have that, for each $x \in X$:

1. $S^x_i \cap T^x_i = \{x\}$ for each $i \leq l$.
2. $S^x_0 \supseteq S^x_1 \supseteq \cdots \supseteq S^x_l$.
3. $T^x_0 \subseteq T^x_1 \subseteq \cdots \subseteq T^x_l$.

[9] More generally we can build lattices of these terms since the properties are universally quantified.
[10] If a linear system is complete, then it must derive s and m with fixed size derivations.

First, we give a definition of the measures we will use to deduce the bound of Thm. 30.

▶ **Definition 34** (Measures). For each term t_i in π we define the following measures:

1. $e_\wedge(t_i)$ (resp. $e_\vee(t_i)$) is the number of $\wedge$- (resp. $\vee$-) labelled edges in $\mathcal{W}(t)$.[11]
2. $\nu^x(t_i)$ (resp. $\mu^x(t_i)$) is the size of the citical minterm (resp. maxterm) of x at t_i, i.e. $|S_i^x|$ (resp. $|T_i^x|$).
3. $\nu(t_i) := \sum_{x \in X} \nu^x(t_i)$ and $\mu(t_i) := \sum_{x \in X} \mu^x(t_i)$.

We point out some simple properties of these measures.

▶ **Proposition 35.** *Let $e := \frac{1}{2}n(n-1)$. We have the following:*

1. *$e_\wedge, e_\vee \leq e$, and $e_\wedge + e_\vee = e$.*
2. *For each $x \in X$ we have that $\nu^x, \mu^x \leq n$, so $\nu, \mu \leq n^2$.*

Proof. 1 follows from the fact that there are only e edges in a web, all of which must be labelled $\wedge$ or $\vee$. For 2, simply observe that a minterm or maxterm has size at most n. ◀

We show that, whenever an $\wedge$-edge becomes labelled $\vee$, some minterm strictly decreases.

▶ **Proposition 36.** *Suppose, for some $i < l$, we have that x —— y in $\mathcal{W}(t_i)$ and x ····· y in $\mathcal{W}(t_{i+1})$. Then there is a minterm S of t_i, and a minterm S' of t_{i+1} such that $S' \subsetneq S$.*

Proof. Take any maximal $\wedge$-clique in $\mathcal{W}(t_i)$ containing x and y, of which there must be at least one. This must have a $\wedge$-subclique which is maximal in $\mathcal{W}(t_{i+1})$, by Prop. 13 and Thm. 19. This subclique cannot contain both x and y, so the inclusion must be strict. ◀

We show that, if a minterm strictly decreases in size, some critical maxterm must strictly increase in size.

▶ **Proposition 37.** *Suppose for $j > i$ there is some minterm S_i of t_i and some minterm S_j of t_j such that $S_j \subsetneq S_i$. Then, for some variable $x \in X$, we have that $T_i^x \subsetneq T_j^x$.*

Proof. We let x be some variable in $x \in S_i \setminus S_j$, which must be nonempty by hypothesis. By Thm. 14 we have that $|T_i^x \cap S_i| = 1$, so it must be that $T_i^x \cap S_i = \{x\}$ by construction.

On the other hand we also have that $|T_j^x \cap S_j| = 1$, and so there is some (unique) $y \in T_j^x \cap S_j$. Now, since $S_i \supsetneq S_j$ we must have $y \in S_i$. However we cannot have $y \in T_i^x$ since that would imply that $\{x, y\} \subset T_i^x \cap S_i$, contradicting the above.

Finally, by soundness, we have that $T_i^x \supsetneq T_j^x$ as required. ◀

Recall that all $\mathcal{W}(t_i)$ are distinct, so both $e_\wedge$ and $e_\vee$ must change at each step of π.

▶ **Lemma 38** (Increasing measure). *The lexicographical product $\mu \times e_\wedge$ is strictly increasing at each step of π.*

Proof. Notice that, by Lemma 33.2, we have that $T_0^x \subseteq T_1^x \subseteq \cdots \subseteq T_l^x$, i.e. μ is non-decreasing. So let us consider the case that $e_\wedge$ decreases at some step and show that μ must strictly increase. If $e_\wedge(t_i) > e_\wedge(t_{i+1})$ then we must have that some edge is labelled $\wedge$ in $\mathcal{W}(t_i)$ and labelled $\vee$ in $\mathcal{W}(t_{i+1})$. Hence, by Prop. 36 some minterm has strictly decreased in size and so by Prop. 37 some critical maxterm must have strictly increased in size. ◀

From here we can finally give a simple proof of our main result:

[11] Of course, these measures can more generally be defined for any linear term.

Proof of Thm. 30. By Prop. 35 we have that $\mu = O(n^2) = e_\wedge$ and so, since $s \to t$ is nontrivial, it must be that the length l of π is $O(n^4)$, as required. ◀

Notice that, while the various settings exhibit a symmetry between $\wedge$ and $\vee$, it is the property of soundness that induces the necessary asymmetry required to achieve this result.

7 Canonicity

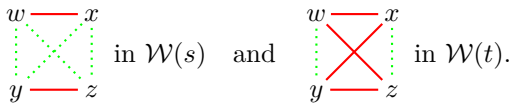
We show that the medial rule is somehow "canonical": it is the *only* linear inference that, on relation webs, preserves $\wedge$-edges (up to reflexive transitive closure modulo AC).

On the other hand, the switch rule is not canonical, in the sense that it is not the only rule that preserves $\vee$-edges, and we give an example of this from previous work. However we conjecture a weaker form of canonicity for the switch rule.

7.1 Canonicity of medial

▶ **Definition 39.** Let s and t be linear terms on a set X of variables. We write $s \blacktriangleleft\blacktriangleright t$ if:

1. Whenever x —— y in $\mathcal{W}(s)$ we have that x —— y in $\mathcal{W}(t)$.
2. Whenever x ····· y in $\mathcal{W}(s)$ and x —— y in $\mathcal{W}(t)$, there are $w, z \in X$ such that,



The following result appeared in [23], where a detailed proof may be found.

▶ **Proposition 40** (Medial criterion). *$s \blacktriangleleft\blacktriangleright t$ if and only if $s \xrightarrow[\mathsf{m}]{*} t$.*

▶ **Definition 41.** If t is a linear term with $x, y, z \in Var(t)$, we say that y *separates x from z* in $\mathcal{W}(t)$ if x —— y in $\mathcal{W}(t)$ and y ····· z in $\mathcal{W}(t)$.

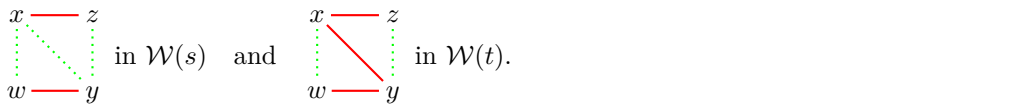
▶ **Theorem 42.** *Let s and t be linear terms on a variable set X. The following are equivalent:*

1. *$s \leq t$ and for all $x, y \in X$ we have x —— y in $\mathcal{W}(s)$ implies x —— y in $\mathcal{W}(t)$.*
2. *$s \blacktriangleleft\blacktriangleright t$.*
3. *$s \xrightarrow[\mathsf{m}]{*} t$.*

Proof. We have that $2 \implies 3$ by Prop. 40 and $3 \implies 1$ by inspection of medial, so it suffices to show $1 \implies 2$. For this, assume 1 and suppose x ····· y in $\mathcal{W}(s)$ and x —— y in $\mathcal{W}(t)$, and let S be a minterm of s containing x. We must have $S \supsetneq \{x\}$ since x —— y in $\mathcal{W}(t)$ and $s \to t$ is sound.[12] Similarly there must be a maxterm T of t containing y such that $T \supsetneq \{y\}$. Now, by 1, it must be that S (resp. T) is also a minterm (resp. maxterm) of t (resp. s),[13] and so, by Thm. 14, there is some (unique) $z \in S \cap T$ which, by definition, separates x from y in both $\mathcal{W}(s)$ and $\mathcal{W}(t)$. By a symmetric argument we obtain a w separating y

[12] Recall that, by Prop. 13 and Thm. 19, there must a subset of S which is a maximal $\wedge$-clique in $\mathcal{W}(t)$.
[13] Since by 1, $\wedge$-edges (resp. $\vee$-edges) are preserved left-to-right (resp. right-to-left) and so $\wedge$-cliques (resp. $\vee$-cliques) must be preserved (resp. reflected). Of course, these must be maximal by soundness.

from x in both $\mathcal{W}(s)$ and $\mathcal{W}(t)$. By construction, w and z must be distinct, so we have the following situation,



whence 2 follows by P_4-freeness. ◀

▶ **Corollary 43.** *The bound in Thm. 30.1 can be improved to* $O(n^3)$.

For the proof, let us first define $\#_\wedge(t)$ to be the number of $\wedge$ symbols occurring in t.

Proof of Cor. 43. Instead of using $e_\wedge$ in Lemma 38, use $\#_\wedge$, which is linear in the size of the term. If no $\wedge$-edge becomes labelled $\vee$, $\#_\wedge$ must have strictly decreased by Thm. 42. ◀

7.2 Towards canonicity of switch

Switch is not canonical in the same sense, due to the following example appearing in [9]:

$$\begin{array}{rl} & ([z \vee v] \wedge [x \vee (z' \wedge v')]) \vee ([(y \wedge u) \vee w] \wedge [y' \vee u']) \\ \to & [x \vee (y \wedge y')] \wedge [(z \wedge z') \vee (u \wedge u')] \wedge [(v \wedge v') \vee w] \end{array} \tag{2}$$

For this inference, no $\vee$-edge becomes a $\wedge$-edge, but it is not derivable by switch and medial, as shown in [9]. However, we conjecture that a weaker form of canonicity applies.

▶ **Conjecture 44.** *If* $s \to t$ *is sound and nontrivial, every* $\vee$*-edge in* $\mathcal{W}(s)$ *is also labelled* $\vee$ *in* $\mathcal{W}(t)$*,* $s \to t$*, and* $\#_\wedge(s) = \#_\wedge(t)$*, then* $s \overset{*}{\underset{\mathsf{s}}{\to}} t$.

8 Final remarks

Conjecture 44 above is inspired by the observation that the only nontrivial linear inference we know of that preserves $\#_\wedge$ is s. There are known trivial examples (e.g. "supermix" from [9] : $x \wedge (y_1 \vee \cdots \vee y_k) \to x \vee (y_1 \wedge \cdots \wedge y_k)$) that *increase* $\#_\wedge$ but every nontrivial rule we know of, including the rule (2) above, strictly decreases it.

Notice that, the stronger conjecture that s is the only nontrivial rule that preserves $\#_\wedge$ already implies our main result, since $\#_\wedge \times e_\wedge$ would be a strictly decreasing measure.

We point out that this measure is used for the usual proof of termination of $\{\mathsf{s}, \mathsf{m}\}$ (modulo AC), e.g. in [9], and also yields a cubic bound on termination. In this work we have matched that bound for *all* linear derivations in the case of weak normalisation, and in the case of strong normalisation for derivations (modulo ACU) that are not globally trivial.

Finally, some preliminary research has shown that the length-bound for termination of $\{\mathsf{s}, \mathsf{m}\}$ can be improved to a quadratic. We conjecture that such an improvement is also possible in the case of (nontrivial) linear derivations in general.

References

1 Denis Bechet, Philippe de Groote, and Christian Retoré. A complete axiomatisation of the inclusion of series-parallel partial orders. In H. Common, editor, *Rewriting Techniques and Applications, RTA 1997*, volume 1232 of *LNCS*, pages 230–240. Springer, 1997.

2 Kai Brünnler and Alwen F. Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *LNCS*, pages 347–361. Springer, 2001.

3 Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):1–34, 2009. Article 14.

4 Michael Chein. Algorithmes d'écriture de fonctions booléennes croissantes en sommes et produits. *Revue Française d'Informatique et de Recherche Opérationnelle*, 1:97–105, 1967.

5 Stephen Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

6 Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *Proceedings of the 6th annual ACM Symposium on Theory of Computing*, pages 135–148. ACM Press, 1974.

7 Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

8 Anupam Das. On the proof complexity of cut-free bounded deep inference. In K. Brünnler and G. Metcalfe, editors, *Tableaux 2011*, volume 6793 of *LNAI*, pages 134–148, 2011.

9 Anupam Das. Rewriting with linear inferences in propositional logic. In Femke van Raamsdonk, editor, *RTA'13*, volume 21 of *LIPIcs*, pages 158–173, 2013.

10 Nachum Dershowitz and Jieh Hsiang. Rewrite methods for clausal and non-clausal theorem proving. In *Automata, Languages and Programming*, pages 331–346. Springer, 1983.

11 Moshe Dubiner and Uri Zwick. Amplification by read-once formulas. *SIAM Journal on Computing*, 26(1):15–38, 1997.

12 A. Guglielmi and L. Straßburger. Non-commutativity and MELL in the calculus of structures. In L. Fribourg, editor, *CSL 2001*, volume 2142 of *LNCS*, pages 54–68, 2001.

13 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

14 V. A. Gurvich. Repetition-free boolean functions. *Uspekhi Matematicheskikh Nauk*, 32(1):183–184, 1977.

15 V. A. Gurvich. On the normal form of positional games. In *Soviet math. dokl*, volume 25, pages 572–574, 1982.

16 Rafi Heiman, Ilan Newman, and Avi Wigderson. On read-once threshold formulae and their randomized decision tree complexity. In *Theoret. Comp. Science*, pages 78–87, 1994.

17 Lisa Hellerstein and Marek Karpinski. Computational complexity of learning read-once formulas over different bases. Technical report, University of Bonn, 1990.

18 Jieh Hsiang. Refutational theorem proving using term-rewriting systems. *Artificial Intelligence*, 25(3):255–300, 1985.

19 Aleksandr Vasilevich Kuznetsov. Non-repeating contact schemes and non-repeating superpositions of functions of algebra of logic. *Trudy Matematicheskogo Instituta im. VA Steklova*, 51:186–225, 1958.

20 Leonid A. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

21 Rolf H. Möhring. Computationally tractable classes of ordered sets. In I. Rival, editor, *Algorithms and Order*, pages 105–194. Kluwer Acad. Publ., 1989.

22 Christian Retoré. *Réseaux et Séquents Ordonnés*. PhD thesis, Université Paris VII, 1993.

23 Lutz Straßburger. A characterisation of medial as rewriting rule. In Franz Baader, editor, *RTA 2007*, volume 4533 of *LNCS*, pages 344–358. Springer-Verlag, 2007.

24 Lutz Straßburger. Extension without cut. *Ann. Pure Appl. Logic*, 163(12):1995–2007, 2012.

25 Terese. *Term rewriting systems*. Cambridge University Press, 2003.

26 L. G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5(3):363 – 366, 1984.