No complete linear term rewriting system for propositional logic

Anupam Das and Lutz Straßburger

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 Dummy classification

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

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Linear inferences, as defined in [7] and also known as "balanced" tautologies (e.g. in [17]) or linear rules (e.g. in deep inference [2], [3]), 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 to consider is essentially due to the observation that *all* boolean tautologies can be written in this form, by means of a polynomial-time translation [17]. Consequently, it has been asked (e.g. in [7] [17]) 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 [7] [17], 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 a sound and complete "proof system" [2] for CPL with no meaningful duplication, creation or

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

[2] Recall that proof systems are usually required to be efficiently (i.e. polynomial-time) checkable, e.g. as defined in [13].

Conference title on which this volume is based on.
Editors: Billy Editor, Bill Editors; pp. 1–16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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[4] 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 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 [5]).[5] They are closely related to positional games (first mentioned in [11]) and have been used in amplification of approximation circuits, (first in [19], more generally in [8]) 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 whose nodes are the variables of a formula, which we call "relation webs" in this work, following [9] [16].

In this paper we work in both these settings, as well as that of term rewriting, and present novel interplays between them. In particular, the proof of our main result, Thm. 32, 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. 14 and Thm. 20, 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, e.g. 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* [18]. In this way we suggest that our main result, a natural statement in the language of rewriting theory, is of independent interest to the rewriting community.

The organisation of this paper is as follows. In Sects. 2 and 3 we present the basics on term rewriting in CPL and their 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 to prove polynomial-time weak normalisation in Sect. 6.

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, finally , in Sect. 8 we make some concluding remarks.

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

[4] Assumed to contain certain core rules.

[5] 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 [10] characterising these functions. The book we reference presents an excellent and comprehensive introduction to the area.

2 Preliminaries on rewriting theory

We generally work in the first-order term rewriting setting defined in the Terese textbook, *Term Rewriting Systems* [18]. 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.

Importantly we point out that two conventions we adopt differ from their 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.

The motivation for (2) is that we fundamentally care about weak normalisation, e.g. Cor. 33, but it will be useful to make statements resembling strong normalisation, under this notion of rewriting modulo, e.g. Thm. 32. All the equivalence relations we will work with are polynomial-time decidable, and so this convention is consistent with the previous one. The same notion of rewriting modulo was also used in previous work [7].

Propositional logic in the term rewriting setting

Our language is built from the connectives $\top, \bot, \neg, \wedge, \vee$ and a set Var of propositional variables, typically denoted $x, y, z, \dots$. The set Ter of formulae, or *terms* is built freely from this signature in the usual way, typically denoted $s, t, u, \dots$. 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 there are no occurrences of $\top$ and $\bot$ in t.

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 $x \in \mathit{Var}(t)$, 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.

▶ **Definition 1** (Substitutions). A *substitution* is a mapping $\sigma \colon \mathit{Var} \to \mathit{Ter}$ 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 $\mathit{Ter} \to \mathit{Ter}$, in the usual way.

▶ **Definition 2** (Rewrite rules). A *rewrite rule* is an expression $l \to r$, where l and r are terms. We write $\rho : l \to_\rho r$ 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* (*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 $s \to_\rho t$ to express that $s \to t$ is a *reduction step* of ρ, i.e. that $s = \sigma(l)$ and $t = \sigma(r)$ for some substitution σ.

▶ **Definition 3** (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 $\to_R$, where $s \to_R t$ just if $s \to_\rho t$ for some $\rho \in R$.

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

▶ **Definition 4** (Derivations)**.** A *derivation* under a binary relation $\to_R$ on *Ter* is a sequence $\pi : t_0 \to_R t_1 \to_R \cdots \to_R t_l$. In this case we say that π has *length* l.

We also write $\to_R^*$ to denote the reflexive transitive closure of $\to_R$.

▶ **Definition 5** (Rewriting modulo)**.** For an equivalence relation $\sim$ on *Ter* and a TRS R, we define the relation $\to_{R/\sim}$ by $s \to_{R/\sim} t$ if there are s', t' such that $s \sim s' \to_R t' \sim t$ such that $s' \nsim t'$.

An $R/\sim$-derivation is also called an R-derivation *modulo* $\sim$.

In this work will consider linear equivalence relations, for example 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 nonlinear system U' that extends U by the following rules:

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

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 subscripts to express a relation between two terms, e.g. the reduction step $s \to_\rho t$. This distinction will become significant in the next section.

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 set of variables $X \subseteq Var$ is a map $\{0,1\}^X \to \{0,1\}$.[6] We associate $\{0,1\}^X$ with $\mathcal{P}(X)$, the powerset of X, i.e. we specify an argument of a boolean functions by the subset of its variables assigned to 1.

Formally, a function $\nu : X \to \{0,1\}$ is specified as a set X_ν where $x \in X_\nu$ 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.

[6] In this work we will insist that this X is always a finite set.

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 show 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 6** (Soundness). We say that a rewrite rule $s \to t$ is *sound* if s, t compute boolean functions f, g respectively such that $f \leq g$. We say that a TRS is sound if all its rules are. A *linear inference* is a sound linear rewrite rule. The set of all linear inferences is denoted L.

▶ Notation 7. 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 [7]. 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[7] of variables in a left- and right- linear rule would constitute what we call a "triviality" in Section 5 where we also elaborate on and address this issue.

Finally, we give a known result, essentially from [17], that was one of the key motivations for this work:

▶ **Proposition 8.** 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 this translation is not very complicated, however we point out that it does induce an at most quadratic blowup in size from a tautology to a linear inference.

Proof of Proposition 8. The proof can essentially be found in [17]. Since the setting there is slightly different, we repeat it here for the sake of completeness.

That L is in **coNP** is trivial since checking soundness of the rewrite rule $s \to t$ is checking validity of the formula $\neg s \vee t$. To prove **coNP**-hardness, we can reduce validity of general tautologies negation normal form to soundness of linear rewrite rules. For this, let t be a formula in negation normal form (i.e., negation $\neg$ occurs only in front of variables). We let t' be the formula obtained from t by doing the following replacement for every variable x occurring in t: Let n be the number of occurrences of x in positive form in t, and let m

[7] We point out that in many settings, indeed in [18], 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.

be the number of occurrences of $\neg x$ in t. If $n \geq 1$ and $m \geq 1$, then introduce $2 \cdot n \cdot m$ fresh variables $x'_{i,j}, x''_{i,j}$ for $1 \leq i \leq n$ and $1 \leq j \leq m$. Now replace for every $1 \leq i \leq n$ the ith occurrence of x by $x'_{i,1} \vee \ldots \vee x'_{i,m}$, and replace for every $1 \leq j \leq m$ the jth occurrence of $\neg x$ by $x''_{1,j} \vee \ldots \vee x''_{n,j}$. If $n = 0$, then introduce $2m$ fresh variables $x'_1, x''_1, \ldots, x'_m, x''_m$ and replace the jth $\neg x$ by $x'_j \wedge x''_j$. If $m = 0$, then introduce $2n$ fresh variables $x'_1, x''_1, \ldots, x'_n, x''_n$ and replace the ith x by $x'_i \wedge x''_i$.

Now t' is a linear term (without negation), and its size is quadratic in the size of t. Now let s' be the conjunction of all pairs $x' \vee x''$ of variables introduced in the construction of t'. Then $\mathit{Var}(s') = \mathit{Var}(t')$ and s' is also a linear term and has 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 $\neg x'$. 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 9** (Read-once functions). A boolean function is *read-once* if it is computed by some linear term (of propositional logic).

The following result first appeared in [10], and was later generalised to read-once "threshold" formulae in [12].

▶ **Theorem 10.** *If constant-free negation-free linear terms compute the same (read-once) boolean function then they are equivalent modulo AC.*

We point out that a proof of this can be easily derived from results in the next section, by the presentation of equivalence classes modulo AC and graph-theoretic definition of soundness.

The following consequences of the above result appeared first in [7], where detailed proofs may be found.

▶ **Corollary 11.** *If negation-free linear terms compute the same (read-once) boolean function then 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 [6]. ◀

▶ **Corollary 12.** *Any sound negation-free linear system, 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 let us restrict our attention to *monotone* boolean functions, i.e. those $f : \{0,1\}^X \to \{0,1\}$ 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, also called "prime implicants" and "prime clauses" respectively, correspond to minimal DNF and CNF representations, respectively, of a monotone boolean function. We refer the reader to [5] 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. 15 below) to derive our main results.

▶ **Definition 13.** Let f be a monotone boolean function on a variable set X. A set $Y \subseteq X$ is a *minterm* (*maxterm*) for f if it is a minimal set such that $f(Y) = 1$ (resp. $f(\overline{Y}) = 0$).

The set of all minterms (maxterms) of f is denoted $MIN(f)$ (resp. $MAX(f)$).

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

▶ **Proposition 14** (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. First, we show $1 \implies 2$. Assume $f \leq g$, and by way of contradiction, assume there is an $S \in MIN(f)$ such that there is no $S' \in MIN(g)$ with $S' \subseteq S$. Then we have $f(S) = 1$ and $g(S) = 0$ contradicting $f \leq g$.

Next, we show $2 \implies 1$. For this, 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 $g(Y) = 1$ and so $f \leq g$.

For showing $1 \implies 3$ and $3 \implies 1$ we proceed analogously. ◀

The following classical result characterising the read-once functions over $\wedge$ and $\vee$ is due to Gurvich in [10], but has appeared in various presentations. In particular, the proof appearing in [5] uses the notion of *co-occurrence* graph, to which our "relation webs" in the next section essentially amounts.[8]

▶ **Theorem 15** (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 let us 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* [9] [16]. 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.

4.1 Preliminary material

We make use of *labeled 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 *labeled edges* as $E(G)$.

[8] 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.

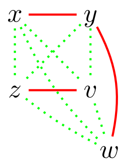
For graphs G, H such that $V(G) \subseteq V(H)$, we say "G in H" to assert that G is an (induced) subgraph[9] of H. In particular we say "$x \overset{\star}{\text{——}} y$ in G" to express that the edge $\{x, y\}$ is labeled $\star$ in the graph G.

We say that a set $X \subseteq V(G)$ is a $\star$-*clique* if every pair $x, y \in X$ has a $\star$-labeled 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 16** (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 labeled by their first common connective in t.

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

▶ **Example 17.** The term $([x \vee w] \wedge y) \vee (z \wedge v)$ has the relation web 

.

▶ Remark (Labels). We point out that, instead of using labeled complete graphs, we could have also used unlabeled 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 [5].

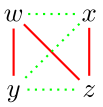
However, we use the current formulation in order to maintain consistency with the previous literature, e.g. [9] and [16], and also since it helps write certain arguments, e.g. in Sect. 7, where we need to draw graphs with incomplete information.

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 18.** *Constant-free negation-free linear terms are equivalent modulo AC iff 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 19.** *A (complete $\{\wedge, \vee\}$-labeled) graph on X is the web of some (negation-free constant-free) linear term on X iff it it does not have any subgraphs of the following configuration (called P_4):*



(1)

A proof of this property can be found, for example, in [14] [15][1][9]. 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.

[9] In fact, since we will deal with only complete graphs and complete subgraphs, all subgraphs will implicitly be 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 20.** *A set of variables is a minterm (maxterm) of a negation-free constant-free linear term t iff it is a maximal $\wedge$-clique (resp. $\vee$-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 21** (Inductive definition of minterms and maxterms). *Let t be a linear term. A set $S \subseteq Var(t)$ is a minterm of t just if:*

- *$t = x$ and $S = \{x\}$.*
- *$t = t_1 \vee t_2$ and S is a minterm of t_1 or t_2.*
- *$t = t_1 \wedge t_2$ and $S = S_1 \sqcup S_2$ where each S_i is a minterm of t_i.*

Dually, a set $T \subseteq Var(t)$ is a maxterm of t just if:

- *$t = x$ and $T = \{x\}$.*
- *$t = t_1 \vee t_2$ and $T = T_1 \sqcup T_2$ 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 t_2.*

5 Dealing with constants, negation, erasure and trivialities

We show in this section that we need not deal with linear rules that contain constants or negation when looking for a complete linear system, or linear rules that whose variables do not all occur on both sides. The fundamental concept here is that of "triviality", which was first introduced in [7] as "semantic triviality". This turns out also to be precisely the concept which allows us to polynomially restrict the length of linear derivations in Sect. 6 for our main result.

Since many of the following notions and results already appeared in [7], we present only brief arguments in this section.

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 22** (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 $f(Y \cup \{x\}) \leq g(Y \setminus \{x\})$.

▶ Remark (Hereditariness of triviality). Notice that the triviality relation is somehow hereditary: if a sound sequence $f_0 \to f_1 \to \dots \to f_l$ of boolean functions is trivial at some point $f_i \to f_{i+1}$ 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 35, on which our main result crucially relies.

In a similar way as we could express soundness with minterms or maxterms in Prop. 14, we can also define triviality with minterms or maxterms.

▶ **Proposition 23.** *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. For showing $1 \implies 3$ 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 24.** 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 (for presentation reasons) assume that rewriting is conducted modulo ACU.

▶ **Lemma 25.** *If* s, t *are negation-free linear terms on* $x_1, \dots, x_n$ *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. u *contains precisely the trivial variables of* $s \to t$*.*
3. $s' \to t'$ *is sound and nontrivial.*

Proof. See [7]. ◀

▶ **Theorem 26.** *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 the lemma, whence we know that the number of such steps is polynomial 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 27.** *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

A variable x occurs either in the same polarity on both sides of a linear inference or positively on one side and negatively on the other side. In the first case, we can soundly eliminate the scope of negation on x (and thus every variable) by De Morgan laws, and finally replacing $\neg x$ for a fresh variable x'. In the second case we have a triviality at x.

▶ **Definition 28** (De Morgan laws)**.** By N let us denote the following equational theory:

$$\neg\neg a = a \quad , \quad \neg(s \wedge t) = \neg s \wedge \neg t \quad , \quad \neg(s \vee t) = \neg s \wedge \neg t$$

▶ **Proposition 29.** *Suppose R is a linear system complete for negation-free linear inferences. Then R/N is complete for* L*.*

5.4 Constants

Let us assume in this subsection that terms are negation-free, in light of that 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 30.** *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 U' to a constant-free term or $\bot$ or $\top$. 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 31.** *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

For presentation reasons, throughout this section we assume the following,

Terms are constant-free, negation-free and linear on a set of variables X.

in light of Thm. 31 in the previous section.

The following is our main result.

▶ **Main Theorem 32.** *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 of this we show how this implies that there is no sound and complete linear system, modulo hardness assumptions.

▶ **Corollary 33.** *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. Follows from the theorem above, Lemma 25 and Thm. 31. ◀

▶ **Corollary 34.** *There is no sound linear system complete for* L *unless* $\mathbf{coNP} = \mathbf{NP}$*.*

Proof. Recall that the set L of linear rules in **coNP**-complete, by Proposition 8. But the above corollary constitutes an **NP**-procedure for L: guess the correct sequence of R-steps to construct a derivation of $s \to t$, which yields the required result. ◀

In the next section we give the crucial lemma that allows us to attain a proof of main theorem. The main argument is then 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 show simultaneously the dual property for maxterms.

▶ **Lemma 35** (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. $\forall S_i \in \mathit{MIN}(f_i).S_i \subseteq S^x \implies x \in S_i$.
2. $\forall T_i \in \mathit{MAX}(g_i).T_i \subseteq T^x \implies x \in T_i$.
3. $\forall S_i \in \mathit{MIN}(f_i), \forall T_i \in \mathit{MAX}(g_i).S_i \subseteq S^x, T_i \subseteq T^x \implies S_i \cap T_i = \{x\}$.

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. 14, 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, 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. 15 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[10] 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 theorem

Throughout this section let us fix a sound (negation-free constant-free) linear system R, which we assume to contain s, m,[11] 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. 15 and have distinct relation webs.

Let us fix an R-derivation,

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

[10] More generally we can actually build lattices of these terms since the properties are universally quantified.

[11] If a linear system is sound and complete, then so is its extension by s, m.

on a variable set X, with $s \to t$ nontrivial.

Now, let us fix choices S_i^x (T_i^x) of critical minterms (resp. maxterms) of t_i, by Lemma 35. I.e. we have that, for each $x \in X$:

1. $S_i^x \cap T_i^x = \{x\}$ for each $i \leq l$.
2. $S_0^x \supseteq S_1^x \supseteq \cdots \supseteq S_l^x$.
3. $T_0^x \subseteq T_1^x \subseteq \cdots \subseteq T_l^x$.

First, we give a definition of the measures we will use to deduce the bound of Thm. 32.

▶ **Definition 36** (Measures)**.** For each term t_i in π we define the following measures:

1. $\mathsf{r}(t_i)$ (resp. $\mathsf{g}(t_i)$) is the number of $\wedge$- (resp. $\vee$-) labeled edges in $\mathcal{W}(t)$. [12]
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 37.** *Let $e := \frac{1}{2}n(n-1)$. We have the following:*

1. $\mathsf{r}, \mathsf{g} \leq e$*, and* $\mathsf{r} + \mathsf{g} = e$*.*
2. *For each $x \in X$ we have that $\nu^x, \mu^x \leq n$, so $\nu, \mu \leq n^2$.*

Proof. (1) follows since there are only e edges in a web, all of which must be labeled $\wedge$ or $\vee$. (2) follows since each minterm and maxterm has size at most n. ◀

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

▶ **Proposition 38.** *Suppose, for some $i < l$ and some $x, y \in X$, 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 the alternative definition of soundness, Prop. 14. This subclique cannot contain both x or y, so the inclusion must be strict. ◀

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

▶ **Proposition 39.** *Suppose for $j > i$ there is some minterm S_i of t_i and some $S_j \subsetneq S_i$ a minterm of t_j. Then, for some variable $x \in X$, we have that $T_i^x \supsetneq T_j^x$.*

Proof. We let x be some variable in $x \in S_i \setminus S_j$, which must be nonempty by hypothesis. By Thm. 15 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. ◀

Notice that, since each t_i computes a distinct boolean function, we must have that both r and g change at each step.

[12] Of course, these measures are more generally defined for any linear term.

▶ **Lemma 40** (Increasing measure). *The lexicographical product $\mu \times \mathsf{r}$ is strictly increasing at each step of π.*

Proof. Notice that, by Lemma 35.2, 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 r decreases at some step and show that μ must strictly increase.

If $\mathsf{r}(t_i) > \mathsf{r}(t_{i+1})$ then we must have that some edge is labeled $\wedge$ in $\mathcal{W}(t_i)$ and labeled $\vee$ in $\mathcal{W}(t_{i+1})$. Hence, by Prop. 38 some minterm has strictly decreased in size and so by Prop. 39 some critical maxterm must have strictly increased in size. ◀

From here it is simple to give a proof of our main result:

Proof of Thm. 32. By Prop. 37 we have that $\mu = O(n^2) = \mathsf{r}$ and so, since $s \to t$ is nontrivial, it must be that the length l of π is $O(n^4)$, as required. ◀

We point out that, while the various settings seem to exhibit a symmetry between $\wedge$ and $\vee$, it is the criterion of soundness that induces the necessary asymmetry required to achieve our result, as exposited in this section.

7 Canonicity

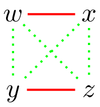
In this section we show that the medial rule is "canonical", in the sense that it is the *only* linear inference (up to reflexive transitive closure modulo AC) that, from the point of view of webs, changes only $\vee$-edges to $\wedge$-edges.

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

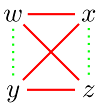
7.1 Canonicity of medial

▶ **Definition 41.** Let s and t be linear terms on a set X of variables. We write $s \lhd\!\!\blacktriangleright t$ 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,



in $\mathcal{W}(s)$ and



in $\mathcal{W}(t)$.

The following result appeared in [16], where a detailed proof may be found.

▶ **Proposition 42** (Medial criterion). *$s \lhd\!\!\blacktriangleright t$ if and only if $s \xrightarrow[\mathsf{m}]{*} t$.*

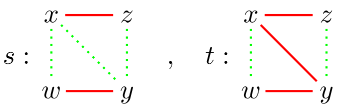
▶ **Definition 43.** If t is a linear term with $x, y, z \in \mathit{Var}(t)$, we say that y *separates* x from z if x —— y and y ······ z .

▶ **Theorem 44.** *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. *$s \lhd\!\!\blacktriangleright t$.*
3. *$s \xrightarrow[\mathsf{m}]{*} t$.*

Proof. We prove $1 \implies 2 \implies 3 \implies 1$.

Assume 1 and suppose x ······ y in s and x —— y in t. Then, by soundness, there must be some z separating x from y in both s and t, and some w separating y from x in both s and t. By construction, z and w must be distinct, so we must have the following situation,



whence 2 follows by P_4-freeness.

Finally, we have that $2 \implies 3$ by Prop. 42 and $3 \implies 1$ by inspection of medial. ◀

▶ **Corollary 45.** *The bound in (1) of Thm. 32 can be improved to $O(n^3)$.*

Proof. Instead of using r in Lemma 40, we can use the number of $\wedge$s occurring in a term, which is now linear in the size of the term. If no $\wedge$-edge becomes labeled $\vee$, the number of $\wedge$s must have strictly decreased by the above result. ◀

7.2 Towards canonicity of switch

Switch is not canonical in the same sense, due to the following example appearing in [7]:

$$\frac{[x \vee (y \wedge y')] \wedge [(z \wedge z') \vee (u \wedge u')] \wedge [(v \wedge v') \vee w]}{([z \vee v] \wedge [x \vee (z' \wedge v')]) \vee ([(y \wedge u) \vee w] \wedge [y' \vee u'])}$$

Notice that, for this inference, no $\vee$-edge becomes a $\wedge$-edge, but it is not derivable by switch and medial, as pointed out in [7].

However, we conjecture that a weaker form of canonicity applies. Let $\#_\wedge(t)$ denote the number of $\wedge$ symbols occurring in t.

▶ **Conjecture 46.** *If $s \to t$ is sound and nontrivial, every $\vee$-edge in $\mathcal{W}(s)$ is also labeled $\vee$ in $\mathcal{W}(t)$, $s \to t$, and $\#_\wedge(s) = \#_\wedge(t)$, then $s \xrightarrow[\mathsf{s}]{*} t$.*

8 Final remarks

The conjecture 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 [7] : $x \wedge (y_1 \vee \cdots \vee y_k) \to x \vee (y_1 \wedge \cdots \wedge y_k)$) that *increase* $\#_\wedge$ but every nontrivial rule we know of, including the one 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 $\#_\wedge \times \mathsf{r}$ would be a strictly decreasing measure.

We point out that this measure is that used for the usual proof of termination of $\{\mathsf{s}, \mathsf{m}\}$ (modulo AC), e.g. in [7], and also yields a cubic bound on termination. In this work we have matched this best known 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, there is ongoing work that the length-bound for termination $\{\mathsf{s}, \mathsf{m}\}$ could be improved to a quadratic. It would be interesting if there was scope for such improvement 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 Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001. `http://www.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf`.

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. `http://cs.bath.ac.uk/ag/p/PrComplDI.pdf`.

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

6 Anupam Das. On the proof complexity of cut-free bounded deep inference. 2011. Tableaux '11.

7 Anupam Das. Rewriting with linear inferences in propositional logic. In Femke van Raamsdonk, editor, *24th International Conference on Rewriting Techniques and Applications (RTA)*, volume 21 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 158–173. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013.

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

9 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007. `http://cs.bath.ac.uk/ag/p/SystIntStr.pdf`.

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

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

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

13 Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory.* Cambridge University Press, New York, NY, USA, 1995.

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

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

16 Lutz Straßburger. A characterisation of medial as rewriting rule. In Franz Baader, editor, *RTA 2007*, volume 4533 of *Lecture Notes in Computer Science*, pages 344–358. Springer-Verlag, 2007. `http://www.lix.polytechnique.fr/~lutz/papers/CharMedial.pdf`.

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

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

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