UNIVERSITY OF BIRMINGHAM
University of Birmingham Research at Birmingham

New minimal linear inferences in boolean logic independent of switch and medial

Das, Anupam; Rice, Alex A.

DOI:
10.4230/LIPIcs.FSCD.2021.14

License:
Creative Commons: Attribution (CC BY)

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Das, A & Rice, AA 2021, New minimal linear inferences in boolean logic independent of switch and medial. in N Kobayashi (ed.), *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*., 14, Leibniz International Proceedings in Informatics, LIPIcs, vol. 195, Schloss Dagstuhl, 6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, Virtual, Buenos Aires, Argentina, 17/07/21. https://doi.org/10.4230/LIPIcs.FSCD.2021.14

Link to publication on Research at Birmingham portal

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 24. Apr. 2024

New Minimal Linear Inferences in Boolean Logic Independent of Switch and Medial

Anupam Das
University of Birmingham, UK

Alex A. Rice
University of Cambridge, UK

Abstract

A *linear inference* is a valid inequality of Boolean algebra in which each variable occurs at most once on each side. Equivalently, it is a linear rewrite rule on Boolean terms that constitutes a valid implication. Linear inferences have played a significant role in structural proof theory, in particular in models of *substructural logics* and in normalisation arguments for *deep inference* proof systems.

Systems of linear logic and, later, deep inference are founded upon two particular linear inferences, *switch* : $x \wedge (y \vee z) \to (x \wedge y) \vee z$, and *medial* : $(w \wedge x) \vee (y \wedge z) \to (w \vee y) \wedge (x \vee z)$. It is well-known that these two are not enough to derive all linear inferences (even modulo all valid linear equations), but beyond this little more is known about the structure of linear inferences in general. In particular despite recurring attention in the literature, the smallest linear inference not derivable under switch and medial ("switch-medial-independent") was not previously known.

In this work we leverage recently developed *graphical* representations of linear formulae to build an implementation that is capable of more efficiently searching for switch-medial-independent inferences. We use it to find two "minimal" 8-variable independent inferences and also prove that no smaller ones exist; in contrast, a previous approach based directly on formulae reached computational limits already at 7 variables. One of these new inferences derives some previously found independent linear inferences. The other exhibits structure seemingly beyond the scope of previous approaches we are aware of; in particular, its existence contradicts a conjecture of Das and Strassburger.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting; Theory of computation → Proof theory; Theory of computation → Linear logic

Keywords and phrases rewriting, linear inference, proof theory, linear logic, implementation

Digital Object Identifier 10.4230/LIPIcs.FSCD.2021.14

Supplementary Material An associated implementation can be found here:
Software (Source Code): `https://github.com/alexarice/lin_inf` [25]
archived at `swh:1:dir:47d6487bfda3d3848d7289f3b5cfae1824e2ae78`

Funding This work was supported by a UKRI Future Leaders Fellowship, *Structure vs. Invariants in Proofs*, project reference MR/S035540/1. Alex Rice acknowledges funding from the Royal Society.

Acknowledgements The authors would like to thank Lutz Strassburger, Ross Horne and Matteo Acclavio for several interesting discussions surrounding this work. We are also grateful to the anonymous reviewers for their valuable feedback and suggestions.

1 Introduction

A *linear inference* is a valid implication $\varphi \to \psi$ of Boolean logic, where φ and ψ are *linear*, i.e. each variable occurs at most once in each of φ and ψ. Such implications have played a crucial role in many areas of structural proof theory. For instance the inference *switch*,

$$\mathsf{s} \; : \; x \wedge (y \vee z) \;\to\; (x \wedge y) \vee z$$

governs the logical behaviour of the *multiplicative* connectives $\parr$ and $\otimes$ of linear logic [16], and similarly the inference *medial*,

© Anupam Das and Alex A. Rice;
licensed under Creative Commons License CC-BY 4.0

$$\mathsf{m} \;:\; (w \wedge x) \vee (y \wedge z) \;\rightarrow\; (w \vee y) \wedge (x \vee z)$$

together with the structural rules *weakening* and *contraction*, governs the logical behaviour of the *additive* connectives $\oplus$ and $\&$ [26, 27]. Both of these inferences are fundamental to *deep inference* proof theory, in particular allowing weakening and contraction to be reduced to atomic form [6, 5], thereby admitting elegant "geometric" proof normalisation procedures based on *atomic flows* [18, 19]. One particular feature of these normalisation procedures is that they are robust under the addition of further linear inferences to the system, thanks to the atomisation of structural steps.

On the other hand the set of *all* linear inferences L plays an essential role in certain models of linear logic and related substructural logics. In particular, the multiplicative fragment of Blass' *game semantics* model of linear logic validates *just* the linear inferences (there called "binary tautologies") [3], and this coincides too with the multiplicative fragment of Japaridze's *computability logic*, cf., e.g., [20]. From a complexity theoretic point of view, the set L is sufficiently rich to encode all of Boolean logic: it is **coNP**-complete [30, 15].

It was recently shown by one of the authors, together with Strassburger, that, despite its significance, L admits no feasible[1] axiomatisation by linear inferences unless $\mathbf{coNP} = \mathbf{NP}$ [14, 15], resolving a long-standing open problem of Blass and Japaridze for their respective logics (see, e.g., [21]). From a Boolean algebra point of view, this means that the class of linear Boolean inequalities has no feasible basis (unless $\mathbf{coNP} = \mathbf{NP}$). From a proof theoretic point of view this means that any propositional proof system (in the Cook-Reckhow sense [8, 9], see also [22]) must necessarily admit some "structural" behaviour, even when restricted to proving only linear inferences (unless $\mathbf{coNP} = \mathbf{NP}$).

An immediate consequence of this result is that s and m above do not suffice to generate all linear inferences (unless $\mathbf{coNP} = \mathbf{NP}$), even modulo all valid linear equations.[2] In fact, this was known before the aforementioned result, due to the identification of an explicit 36 variable inference in [30].[3] Already in that work the question was posed whether such an inference was minimal, and since then the identification of a minimal $\{\mathsf{s}, \mathsf{m}\}$-independent linear inference has been a recurring theme in the literature of this area.

It has been verified in [11] that a minimal $\{\mathsf{s}, \mathsf{m}\}$-independent linear inference must be "non-trivial", as long as we admit all true linear equations. Intuitively, "non-triviality" rules out pathological inferences such as $x \wedge y \rightarrow x \vee y$ or $x \wedge (y \vee z) \rightarrow x \vee (y \wedge z)$. For these inferences the variable, say, y is, in a sense, redundant; it turns out that they may be derived in $\{\mathsf{s}, \mathsf{m}\}$, modulo linear equations, from a smaller non-trivial "core". We recall these arguments in Section 2.

Furthermore [11] identified a 10 variable linear inference that is not derivable by switch and medial (even under linear equations), which Strassburger conjectured was minimal [29]. Around the same time Šipraga attempted a computational approach, searching for independent linear inferences by brute force [31]. However, computational limits were reached already at 7 variables. In particular, every linear inference of up to 6 variables is already derivable by switch and medial, modulo linear equations; due to the aforementioned 10 variable inference, any minimal independent linear inference must have size 7,8,9, or 10.

---

[1] By "feasible", in this work, we always mean polynomial-time computable. This is a natural condition arising from proof theory [8, 9], and is also required for the result to be meaningful: it prevents us just taking the entire set L as an axiomatisation.

[2] The valid linear equations are just associativity, commutativity, and unit laws, cf. [14, 15].

[3] Strassburger refers to the inference as a "balanced tautology", but like the "binary tautologies" of Blass and Japaridze, these are equivalent to linear inferences. In particular we recast Strassburger's example as a bona fide linear inference in Section 3.1.

Since 2013 there have been significant advances in the area, in particular through the proliferation of *graph-theoretic* tools. Indeed, the interplay between formulae and graphs was heavily exploited for the aforementioned result of [14, 15]. Since then, multiple works have emerged in the world of linear proof theory that treat these graphs as "first class citizens", comprising a now active area of research [23, 2, 1, 7].

Contribution

In this work we revisit the question of minimal $\{\mathsf{s}, \mathsf{m}\}$-independent linear inferences by exploiting the aforementioned recent graph theoretic techniques. Such an approach vastly reduces the computational resources necessary and, in particular, we are able to provide a conclusive result: the smallest $\{\mathsf{s}, \mathsf{m}\}$-independent linear inference has size 8. In fact there are two minimal such ones:[4]

$$\begin{aligned} & (z \vee (w \wedge w')) \wedge ((x \wedge x') \vee ((y \vee y') \wedge z')) \\ \to \quad & (z \wedge (x \vee y)) \vee ((w \vee y') \wedge ((w' \wedge x') \vee z')) \end{aligned} \tag{1}$$

$$\begin{aligned} & ((w \wedge w') \vee (x \wedge x')) \wedge ((y \wedge y') \vee (z \wedge z')) \\ \to \quad & (w \wedge y) \vee ((x \vee (w' \wedge z')) \wedge ((x' \wedge y') \vee z)) \end{aligned} \tag{2}$$

We dedicate some discussion to each of these separately in Section 3.2, and include a manual verification of their soundness and $\{\mathsf{s}, \mathsf{m}\}$-independence in Appendix A, as a sanity check.

Our main contribution is an implementation that checks inference for $\{\mathsf{s}, \mathsf{m}\}$-derivability, which was able to confirm that all 7 variable linear inference are derivable from switch and medial. In fact we found (1) independently of the implementation presented in this paper.[5] Ultimately, we improved the implementation to run on inferences of size 8 too, and our inference (1) was duly found, as well as (2) above and its dual. One highlight of this find is that it exhibits a peculiar structural property that *refutes* Conjecture 7.9 from [15], as we explain in Section 3.2.2.

Our implementation [25] is split into a *library* and an *executable*, where the executable implements our search algorithm described in Section 5.2, and the library contains foundations for working with linear inferences using the graph theoretic techniques presented in Section 4. These are written in Rust and designed to be relatively fast while maintaining readability. Our intention is that this could form a reusable base for future investigations in the area, both for linear formulae and for the recent linear graph theoretic settings of [23, 2, 1, 7].

2 Preliminaries

Throughout this paper we shall work with a countably infinite set of **variables**, written x, y, z etc. A **linear formula** on a (finite) set of variables $\mathcal{V}$ is defined recursively as follows:

- $\top$ and $\bot$ are linear formulae on $\varnothing$, the empty set of variables (called **units** or **constants**).
- x and $\neg x$ are linear formulae on $\{x\}$, for each variable x.[6]
- If φ is a linear formula on $\mathcal{V}_1$ and ψ is a linear formula on $\mathcal{V}_2$, with $\mathcal{V}_1 \cap \mathcal{V}_2 = \varnothing$, then $\varphi \vee \psi$ and $\varphi \wedge \psi$ are linear formulae on $\mathcal{V}_1 \cup \mathcal{V}_2$.

[4] Minimal with respect to inter-derivability; unique up to associativity, commutativity, renaming of variables and De Morgan duality.

[5] These two developments were respectively communicated via blog posts [24] and [13].

[6] Note that the restriction of negation to only variables does not compromise expressivity, since the De Morgan laws preserve linearity on a set of variables.

A linear formula that does not contain $\top$ or $\bot$ is **constant-free**. A linear formula with no negated variables (i.e. formulas of form $\neg x$) is **negation-free**. Later in the paper, we will be able to restrict our search to inferences between constant-free negation-free formulae.

In what follows, we shall omit explicit consideration of variable sets, assuming that they are disjoint whenever required by the notation being used.

A relation $\sim$ on linear formulae is **closed under contexts** if for all φ, ψ, χ, we have:

$$\varphi \sim \psi \implies \varphi \wedge \chi \sim \psi \wedge \chi \qquad \varphi \sim \psi \implies \varphi \vee \chi \sim \psi \vee \chi$$
$$\varphi \sim \psi \implies \chi \wedge \varphi \sim \chi \wedge \psi \qquad \varphi \sim \psi \implies \chi \vee \varphi \sim \chi \vee \psi$$

An equivalence relation (on linear formulae) that is closed under contexts is called a (linear) **congruence**.

▶ **Definition 1** (Linear equations). *Let $\sim_{\mathsf{ac}}$ be the smallest congruence satisfying,*

$$\varphi \vee \psi \sim_{\mathsf{ac}} \psi \vee \varphi \qquad \varphi \wedge (\psi \wedge \chi) \sim_{\mathsf{ac}} (\varphi \wedge \psi) \wedge \chi$$
$$\varphi \wedge \psi \sim_{\mathsf{ac}} \psi \wedge \varphi \qquad \varphi \vee (\psi \vee \chi) \sim_{\mathsf{ac}} (\varphi \vee \psi) \vee \chi$$

$\sim_{\mathsf{u}}$ is the smallest congruence satisfying:

$$\begin{array}{llll} \varphi \wedge \top \sim_{\mathsf{u}} \varphi & \varphi \vee \bot \sim_{\mathsf{u}} \varphi & \top \wedge \varphi \sim_{\mathsf{u}} \varphi & \bot \vee \varphi \sim_{\mathsf{u}} \varphi \\ \varphi \wedge \bot \sim_{\mathsf{u}} \bot & \varphi \vee \top \sim_{\mathsf{u}} \top & \bot \wedge \varphi \sim_{\mathsf{u}} \bot & \top \vee \varphi \sim_{\mathsf{u}} \top \end{array} \tag{3}$$

$\sim_{\mathsf{acu}}$ is the smallest congruence containing both $\sim_{\mathsf{ac}}$ and $\sim_{\mathsf{u}}$.

Note that we can have $\varphi \sim_{\mathsf{u}} \psi$ even when φ and ψ have different sets of variables. Moreover, $\sim_{\mathsf{u}}$ generates a unique normal form of linear formulae by maximally eliminating constants:

▶ **Proposition 2** (Folklore, e.g. [10]). *Every formula is $\sim_{\mathsf{u}}$-equivalent to a unique constant-free formula, or is equivalent to $\bot$ or $\top$.*

▶ Remark 3 (On logical equivalence). Clearly, if $\varphi \sim_{\mathsf{acu}} \psi$ then φ and ψ are logically equivalent. In fact, for linear formulae, we also have a converse: two linear formulae φ and ψ are logically equivalent if and only if $\varphi \sim_{\mathsf{acu}} \psi$ [14, 15]. This property follows from Proposition 2 above, the results of Section 2.2, and the graphical representation of linear formulae and their semantics in Section 4.

2.1 Linear inferences

A **linear inference** is just a valid implication $\varphi \to \psi$ (with respect to usual Boolean semantics) where φ and ψ are linear formulae. The left-hand side (LHS) and right-hand side (RHS) of a linear inference, generally speaking, need not be linear formulae on the same variables. Nonetheless we shall occasionally refer to linear inferences "on $\mathcal{V}$" or "on n variables", assuming that the LHS and RHS are both linear formulae on some fixed $\mathcal{V}$ with $|\mathcal{V}| = n$.

There are two linear inferences we shall particularly focus on, due to their prevalence in structural proof theory. **Switch** is the following inference on 3 variables,

$$\mathsf{s} : x \wedge (y \vee z) \to (x \wedge y) \vee z \tag{4}$$

and **medial** is the following inference on 4 variables:

$$\mathsf{m} : (w \wedge x) \vee (y \wedge z) \to (w \vee y) \wedge (x \vee z) \tag{5}$$

We may compose switch and medial (and more generally an arbitrary set of linear inferences) to form new linear inferences by construing them as *term rewriting rules*. More generally, we will consider rewriting derivations modulo the equivalence relations $\sim_{\mathsf{ac}}$ and $\sim_{\mathsf{acu}}$ we introduced earlier. In the latter case, as previously mentioned, the underlying set of variables may change during a derivation, though Proposition 2 will later allow us to work with some fixed set of variables throughout $\{\mathsf{s}, \mathsf{m}\}$ derivations.

▶ **Definition 4** (Rewriting). *We write $\to_{\mathsf{s}}$ and $\to_{\mathsf{m}}$ for the term rewrite systems generated by* (4) *and* (5) *respectively. I.e. $\to_{\mathsf{s}}$ and $\to_{\mathsf{m}}$ are the smallest relations satisfying* (4) *and* (5)*, respectively, closed under substitution and contexts. Write $\varphi \rightsquigarrow_{\mathsf{m}} \psi$ if there are φ', ψ' s.t. $\varphi \sim_{\mathsf{ac}} \varphi' \to_{\mathsf{m}} \psi' \sim_{\mathsf{ac}} \psi$, and $\varphi \rightsquigarrow_{\mathsf{mu}} \psi$ for the same with $\sim_{\mathsf{ac}}$ replaced by $\sim_{\mathsf{acu}}$. Define $\rightsquigarrow_{\mathsf{s}}$, $\rightsquigarrow_{\mathsf{su}}$, $\rightsquigarrow_{\mathsf{ms}}$, $\rightsquigarrow_{\mathsf{msu}}$ similarly (in particular, $\rightsquigarrow_{\mathsf{ms}} = \rightsquigarrow_{\mathsf{m}} \cup \rightsquigarrow_{\mathsf{s}}$).*

We write $\overset{*}{\rightsquigarrow}_{\mathsf{ms}}$ for the reflexive transitive closure of $\rightsquigarrow_{\mathsf{ms}}$, and say $\varphi \to \psi$ is ***$\{\mathsf{s}, \mathsf{m}\}$-derivable*** *if $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \psi$. We may similarly write $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{s}} \psi$ (or $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{m}} \psi$), saying $\varphi \to \psi$ is $\{\mathsf{s}\}$-derivable (resp., $\{\mathsf{m}\}$-derivable), and similarly for other sets of linear inferences.*

Finally, we also write $\overset{*}{\rightsquigarrow}_{\mathsf{msu}}$ for the reflexive transitive closure of $\rightsquigarrow_{\mathsf{msu}}$, and say that $\varphi \to \psi$ is ***$\{\mathsf{s}, \mathsf{m}\}$-derivable with units*** *if $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \psi$. Similarly for $\overset{*}{\rightsquigarrow}_{\mathsf{su}}$, $\overset{*}{\rightsquigarrow}_{\mathsf{mu}}$ and other sets of linear inferences.*

Clearly, s and m are *valid*, so any derivation $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \psi$ comprises a linear inference.

▶ **Example 5** ("Mix"). Units can help us derive even constant-free linear inferences. For instance, **mix**: $\varphi \wedge \psi \to \varphi \vee \psi$ is $\{\mathsf{s}, \mathsf{m}\}$-derivable with units:

$$\varphi \wedge \psi \sim_{\mathsf{acu}} \varphi \wedge (\bot \vee \psi) \to_{\mathsf{s}} (\varphi \wedge \bot) \vee \psi \sim_{\mathsf{acu}} \psi \wedge (\top \vee \varphi) \to_{\mathsf{s}} (\psi \wedge \top) \vee \varphi \sim_{\mathsf{acu}} \varphi \vee \psi$$

Note that mix is not derivable without using instances of $\sim_{\mathsf{u}}$.

▶ **Example 6** (Weakening and duality). By setting $\varphi = \top$ and $\psi = \bot$ in Example 5, we have:

$$\bot \sim_{\mathsf{acu}} \top \wedge \bot \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \top \vee \bot \sim_{\mathsf{acu}} \top$$

Using this we may $\{\mathsf{s}, \mathsf{m}\}$-derive **weakening**, $\varphi \to \varphi \vee \chi$, with units as follows:

$$\varphi \sim_{\mathsf{acu}} \varphi \vee (\bot \wedge \chi) \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \varphi \vee (\top \wedge \chi) \sim_{\mathsf{acu}} \varphi \vee \chi$$

Notice that $\overset{*}{\rightsquigarrow}_{\mathsf{msu}}$ is closed under De Morgan duality: If $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \chi$ and $\bar{\varphi}$ and $\bar{\chi}$ are obtained from φ and χ, respectively, by flipping each $\vee$ to a $\wedge$ and vice versa, then $\bar{\chi} \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \bar{\varphi}$. This follows by direct inspection of s, m and each clause of $\sim_{\mathsf{acu}}$; indeed the same property holds for $\overset{*}{\rightsquigarrow}_{\mathsf{ms}}$ by the same reasoning. As a result, we also have that **coweakening**, $\varphi \wedge \chi \to \varphi$, is $\{\mathsf{s}, \mathsf{m}\}$-derivable with units.

We are now able to state the main theorem of this paper:

▶ **Theorem 7.** *Suppose φ is a linear formula over $\mathcal{V}_1$ and ψ is a linear formula over $\mathcal{V}_2$ and $\mathsf{r} : \varphi \to \psi$ is a linear inference. Then if $|\mathcal{V}_1 \cap \mathcal{V}_2| \leq 7$ we have that $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{msu}} \psi$.*

Furthermore, there is a valid linear inference $\varphi \to \psi$ on 8 variables with $\varphi \not\overset{*}{\rightsquigarrow}_{\mathsf{msu}} \psi$, so 7 is maximal with the property above.

2.2 Trivial inferences

In order to state Theorem 7 above in its most general form, we have allowed linear formulae to include constants and negation, and linear inferences to be between formulae with different variable sets. However it turns out that we may proceed to prove Theorem 7, without loss of

generality, by working with constant-free, negation-free formulae on some fixed set of variables, as was already shown in [11]. This is done by defining the notion of a *trivial* inference, whose $\{\mathsf{s},\mathsf{m}\}$-derivability, with units, may be reduced to that of a smaller non-trivial inference.

▶ **Definition 8.** *An inference $\varphi \to \psi$ is **trivial at a variable** x if $\varphi[\top/x] \to \psi[\bot/x]$ is again a valid inference. An inference is **trivial** if it is trivial at one of its variables.*

▶ **Example 9.** The mix inference from Example 5, $x \wedge y \to x \vee y$, is trivial at x and trivial at y. Note, however, that it is not trivial at x and y "at the same time", in the sense that the simultaneous substition of $\bot$ for x and y in the LHS and $\top$ for x and y in the RHS does not result in a valid implication. In contrast, the linear inference $w \wedge (x \vee y) \to w \vee (x \wedge y)$ from [11] is, indeed, trivial at x and y "at the same time".

Neither switch nor medial are trivial.

▶ Remark 10 (Global vs local triviality). Note that triviality is closed under composition by linear inferences: if $\varphi \to \psi$ is trivial at x and $\psi \to \chi$ is valid, then $\varphi \to \chi$ is trivial at x. Similarly for $\chi \to \psi$ if $\chi \to \varphi$ is valid. One pertinent feature is that the converse does not hold: there are "globally" trivial derivations that are nowhere "locally" trivial. For instance consider the following derivation (from [15, Remark 5.6]):

$$w \wedge x \wedge (y \vee z) \rightsquigarrow_{\mathsf{s}} w \wedge ((x \wedge y) \vee z) \rightsquigarrow_{\mathsf{s}} (w \wedge z) \vee (x \wedge y) \rightsquigarrow_{\mathsf{m}} (w \vee x) \wedge (y \vee z)$$

The derived inference is just an instance of mix, from Example 5, on the redex $w \wedge x$, which is trivial. However, no local step is trivial.

To prove (the first half of) Theorem 7, in Section 5 we will actually prove the following apparent weakening of that statement:

▶ **Theorem 11.** *Let $n < 8$. Let φ and ψ be constant-free negation-free linear formulae on n variables and suppose $\varphi \to \psi$ is a non-trivial linear inference. Then $\varphi \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \psi$.*

In fact this statement is no weaker at all, and we will now see how the consideration of triviality allows us to only deal with such special cases without loss of generality.

▶ **Proposition 12** ([11, Theorem 34])**.** *Let φ and ψ be linear formulae on $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively, and let $\mathsf{r} : \varphi \to \psi$ be a linear inference. There is a non-trivial linear inference $\mathsf{r}' : \varphi' \to \psi'$ on some $\mathcal{V}' \subseteq \mathcal{V}_1 \cap \mathcal{V}_2$ such that $\mathsf{r} : \varphi \to \psi$ is $\{\mathsf{s},\mathsf{m},\mathsf{r}'\}$-derivable with units.*

Note in particular that, in the statement above, if r' is $\{\mathsf{s},\mathsf{m}\}$-derivable with units, then so is r. This is also the case for the next result.

▶ **Proposition 13.** *Let $\mathsf{r} : \varphi \to \psi$ be a non-trivial linear inference among variables $\mathcal{V} \neq \varnothing$. Then there is a constant-free negation-free non-trivial linear inference $\mathsf{r}' : \varphi' \to \psi'$ on $\mathcal{V}$ s.t. $\mathsf{r} : \varphi \to \psi$ is $\{\mathsf{s},\mathsf{m},\mathsf{r}'\}$-derivable with units.*

Proof. First, note that both φ and ψ must be linear formulae on $\mathcal{V}$, since $\varphi \to \psi$ is non-trivial. For the same reason, no variable can occur positively in φ and negatively in ψ or vice-versa, since $\varphi \to \psi$ is non-trivial, and so any negated variable may be safely replaced by its positive counterpart. From here, we simply set φ' and ψ' to be the constant-free formulae (uniquely) obtained from Proposition 2 by $\sim_{\mathsf{u}}$. Non-triviality of r' follows from that of r by logical equivalence. ◀

▶ **Corollary 14.** *The statement of Theorem 11 implies (the first half of) the statement of Theorem 7.*

Proof. Let r be as in Theorem 7. Let r' be the non-trivial linear inference obtained by Proposition 12 above, and let r'' be the non-trivial constant-free negation-free linear inference thence obtained by Proposition 13. By Theorem 11, r'' is $\{\mathsf{s}, \mathsf{m}\}$-derivable and so, by Proposition 12 and Proposition 13, r is also $\{\mathsf{s}, \mathsf{m}\}$-derivable with units. ◀

It is clear that if an inference is derivable with switch and medial then it is also derivable with switch, medial, and units. The following proposition, while not necessary for the proof of Corollary 14, allows the the converse in some cases, and is the reason why our search algorithm in Section 5 will only check for $\{\mathsf{s}, \mathsf{m}\}$-derivability.

▶ **Proposition 15** (Follows from [11], Lemma 28)**.** *Suppose $\varphi \to \psi$ is a non-trivial constant-free negation-free linear inference that is $\{\mathsf{s}, \mathsf{m}\}$-derivable with units. Then $\varphi \to \psi$ is also $\{\mathsf{s}, \mathsf{m}\}$-derivable (without units).*

The idea here is to systematically rewrite a derivation with units to one without, line by line under Proposition 13. Crucially, the invariant of non-triviality constrains the contexts in which constants may occur, ensuring that the constant-elimination procedure preserves instances of s or m.

2.3 Minimality of inferences

Let us take a moment to explain the various notions of "inference minimality" that we shall mention in this work.

Size minimality refers simply to the number of variables the inference contains. E.g. when we say that the 8-variable inferences in the next section are size minimal (or "smallest") non-$\{\mathsf{s}, \mathsf{m}\}$-derivable with units (or **$\{\mathsf{s}, \mathsf{m}\}$-independent** linear inferences, we mean that there are no $\{\mathsf{s}, \mathsf{m}\}$-independent linear inferences with fewer variables.

A linear inference $\varphi \to \psi$ is **logically minimal** if there is no $\sim_{\mathsf{acu}}$-distinct interpolating linear formula. I.e. if $\varphi \to \chi$ and $\chi \to \psi$ are linear inferences, then χ is $\sim_{\mathsf{acu}}$-equivalent to φ or ψ (and so, by Remark 3, is logically equivalent to φ or ψ).

Finally, a linear inference $\varphi \to \psi$ is **$\{\mathsf{s}, \mathsf{m}\}$-minimal** if there is no formula χ s.t. $\varphi \rightsquigarrow_{\mathsf{ms}} \chi$ or $\chi \rightsquigarrow_{\mathsf{ms}} \psi$ and $\chi \to \psi$ or $\varphi \to \chi$, respectively, is a valid linear inference which is not a logical equivalence.

It is clear from the definitions that any logically minimal inference is also $\{\mathsf{s}, \mathsf{m}\}$-minimal, though the converse may not be true. The reason for considering $\{\mathsf{s}, \mathsf{m}\}$-minimality is that it is easier to systematically check by hand. In fact, the implementation we give later in Section 5 further verifies that our new 8-variable inferences are logically minimal.

Logical minimality also serves an important purpose for our proof of Theorem 11, as it allows the following reduction, greatly reducing the search space for our implementation, in fact to nearly 1% of its original size for 8 variable inferences:[7]

▶ **Lemma 16.** *Suppose the statement of Theorem 11 holds whenever $\varphi \to \psi$ is logically minimal. Then the statement of Theorem 11 holds (even when $\varphi \to \psi$ is not logically minimal).*

Proof. Suppose we have a non-trivial inference between constant-free negation-free linear inferences $\varphi \to \psi$. Then $\varphi \to \psi$ can be refined into a chain of logically minimal linear inferences $\varphi \to \chi_0 \to \cdots \to \chi_n \to \psi$. All of these must be non-trivial, as triviality of any of them would imply triviality of $\varphi \to \psi$, cf. Remark 10. Therefore if all such inferences are derivable from switch and medial (with units) then so is $\varphi \to \psi$, by transitivity. ◀

[7] $5364/514486 \approx 1.04\%$

3 New 8-variable {s, m}-independent linear inferences

In this section we shall present the new 8-variable linear inferences of this work ((1) and (2) from the introduction), and give self-contained arguments for their $\{\mathsf{s},\mathsf{m}\}$-independence and $\{\mathsf{s},\mathsf{m}\}$-minimality, as a sort of sanity check for the implementation described in the next section. We shall also briefly discuss some of their structural properties, in reference to previous works in the area. Thanks to the results of the previous section, in particular Proposition 13 and Remark 10, we shall only consider non-trivial constant-free negation-free linear inferences with the same variables in the LHS and RHS. Furthermore, by Proposition 15 we shall only consider $\{\mathsf{s},\mathsf{m}\}$-derivability (i.e., without units).

3.1 Previous linear inferences

In [30] Strassburger presented a 36-variable inference that is $\{\mathsf{s},\mathsf{m}\}$-independent, by an encoding of the pigeonhole principle with 4 pigeons and 3 holes. He there referred to it as a "balanced" tautology, but in our setting it is a linear inference that can be written as follows:[8]

$$\bigwedge_{i=1}^{3}\bigwedge_{j=1}^{i}(x_{ij}\vee x'_{ij})\wedge\bigwedge_{i=1}^{3}\bigwedge_{j=1}^{i}(y_{ij}\vee y'_{ij})\wedge\bigwedge_{i=1}^{3}\bigwedge_{j=1}^{i}(z_{ij}\vee z'_{ij})$$

$$\rightarrow\quad\left[\begin{array}{lllllll}&((x_{11}\vee x_{21}\vee x_{31})&\wedge&(y_{11}\vee y_{21}\vee y_{31})&\wedge&(z_{11}\vee z_{21}\vee z_{31}))\\\vee&((x'_{11}\vee x_{22}\vee x_{32})&\wedge&(y'_{11}\vee y_{22}\vee y_{32})&\wedge&(z'_{11}\vee z_{22}\vee z_{32}))\\\vee&((x'_{21}\vee x'_{22}\vee x_{33})&\wedge&(y'_{21}\vee y'_{22}\vee y_{33})&\wedge&(z'_{21}\vee z'_{22}\vee z_{33}))\\\vee&((x'_{31}\vee x'_{32}\vee x'_{33})&\wedge&(y'_{31}\vee y'_{32}\vee y'_{33})&\wedge&(z'_{31}\vee z'_{32}\vee z'_{33}))\end{array}\right]$$

In [11] Das noticed that a more succinct encoding of the pigeonhole principle could be carried out, with only 3 pigeons and 2 holes, resulting in a 10-variable $\{\mathsf{s},\mathsf{m}\}$-independent linear inference. A variation of that, e.g. as used in [12], is the following:

$$\begin{array}{ll}&(z\vee(w\wedge w'))\wedge(y\vee y')\wedge(u\vee u')\wedge((x\wedge x')\vee z')\\\rightarrow\quad&(z\wedge(x\vee y))\vee(u\wedge x')\vee(w'\wedge u')\vee((w\vee y')\wedge z')\end{array}\tag{6}$$

In fact this is not a $\{\mathsf{s},\mathsf{m}\}$-minimal inference, but we write this one here for comparison to one of the new 8-variable inferences in the next subsection. It can be checked valid and non-trivial by simply checking all cases, or by use of a solver. We do not give an argument for $\{\mathsf{s},\mathsf{m}\}$-independence here, but such an argument is similar to the one we give for an 8-variable inference Equation (7), which is given the next subsection.

3.2 The two minimal 8 variable {s, m}-independent linear inferences

Pre-empting Section 5.2, let us explicitly give the two minimal linear inferences found by our algorithm and justify their $\{\mathsf{s},\mathsf{m}\}$-independence and $\{\mathsf{s},\mathsf{m}\}$-minimality, as a sort of sanity check for our implementation later. As we will see, they both turn out to be significant in their own right, which is why we take the time to consider them separately.

[8] We write Strassburger's inference by encoding each q_{i1j} as x_{ij}, each q_{i2j} as y_{ij}, each q_{i3j} as z_{ij}, and using "primed" variables instead of duals, with the LHS of the inference being the appropriate instances of excluded middle.

3.2.1 A refinement of the 3-2-pigeonhole-principle

First let us consider the 8 variable linear inference that may be used to derive Equation (6), cf. Appendix A.1 (identical to (1) from the introduction):

$$\begin{array}{ll} & (z \vee (w \wedge w')) \wedge ((x \wedge x') \vee ((y \vee y') \wedge z')) \\ \rightarrow & (z \wedge (x \vee y)) \vee ((w \vee y') \wedge ((w' \wedge x') \vee z')) \end{array} \tag{7}$$

Recalling the notion of "duality" from Example 6, let us formally define the **dual** of a linear inference $\varphi \rightarrow \chi$ to be the linear inference $\bar{\chi} \rightarrow \bar{\varphi}$, where $\bar{\varphi}$ and $\bar{\chi}$ are obtained from φ and χ, respectively, by flipping all $\vee$s to $\wedge$s and vice-versa. Considering linear inferences up to renaming of variables, we have:

▶ **Observation 17.** (7) *is self-dual.*

Indeed, the formula structure of the RHS is clearly the dual of that of the LHS, and the mapping from a variable in the LHS to the variable at the same position in the RHS is, in fact, an involution. I.e., u is mapped to itself; v is mapped to y which in turn is mapped to v; v' is mapped to w which is in turn mapped to v'; x is mapped to y' which in turn is mapped to x; and z is mapped to itself. Validity may be routinely checked by any solver, but we give a case analyis of assignments in Appendix A.2.

We may also establish $\{\mathsf{s}, \mathsf{m}\}$-independence and $\{\mathsf{s}, \mathsf{m}\}$-minimality by checking all applications of s or m to the LHS (note that we do not need to check the RHS, by Observation 17 above). This analysis is given explicitly in Appendix A.4.

3.2.2 A counterexample to a conjecture of Das and Strassburger

Finally, our search algorithm found a completely new linear inference (identical to (2) from the introduction):

$$\begin{array}{ll} & ((w \wedge w') \vee (x \wedge x')) \wedge ((y \wedge y') \vee (z \wedge z')) \\ \rightarrow & (w \wedge y) \vee ((x \vee (w' \wedge z')) \wedge ((x' \wedge y') \vee z)) \end{array} \tag{8}$$

Again, validity is routine, but a case analysis is given in Appendix A.3. We may establish $\{\mathsf{s}, \mathsf{m}\}$-independence and $\{\mathsf{s}, \mathsf{m}\}$-minimality again by checking all possible rule applications. This analysis is given in Appendix A.5.

This new inference exhibits a rather interesting property, which we shall frame in terms of the following notion, since it will be used in the next section:

▶ **Definition 18.** *Let φ be a linear formula on a variable set $\mathcal{V}$. For distinct $x, y \in \mathcal{V}$, the* ***least common connective*** *(lcc) of x and y in φ is the connective $\vee$ or $\wedge$ at the root of the smallest subformula of φ containing both x and y.*

Note that, in the inference (8) above, the lcc of w' and x' changes from $\vee$ to $\wedge$, but the lcc of y and y' changes from $\wedge$ to $\vee$. No such example of a minimal linear inference exhibiting both of these properties was known before; switch, medial and all of the linear inferences of this section either preserve $\vee$-lccs or preserve $\wedge$-lccs. In fact, Das and Strassburger showed that any valid linear inference preserving $\wedge$-lccs is already derivable by medial [15, Theorem 7.5], and further conjectured that there was no minimal inference that preserves neither $\wedge$-lccs nor $\vee$-lccs. Naturally, our new inference is a counterexample to that:

▶ **Theorem 19.** *Conjecture 7.9 from [15] is false.*

4 A graph-theoretic presentation of linear inferences

A significant cause of algorithmic complexity when searching for linear inferences is the multitude of formulae equivalent modulo associativity and commutativity ($\sim_{\mathsf{ac}}$). For example, for 7 variables, there are 42577920 formulae (ignoring units), yet only 78416 equivalence classes. Under Remark 3 it would be ideal if we could deal with $\sim_{\mathsf{ac}}$-equivalence classes directly, realising logical and syntactic notions on them in a natural way. This is precisely what is accomplished by the graph-theoretic notion of a *relation web*, cf. [17, 28, 14, 15].

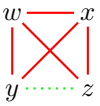
Throughout this section we work only with constant-free negation-free linear formulae, cf. Theorem 11. Recall the notion of *least common connective* (lcc) from Definition 18.

▶ **Definition 20.** *Let φ be a linear formula on a variable set $\mathcal{V}$. The **relation web** (or simply **web**) of φ, written $\mathcal{W}(\varphi)$, is a simple undirected graph with:*

- *The set of nodes of $\mathcal{W}(\varphi)$ is just $\mathcal{V}$, i.e. the variables occurring in φ.*
- *For $x, y \in \mathcal{V}$, there is an edge between x and y in $\mathcal{W}(\varphi)$ if the lcc of x and y in φ is $\wedge$.*

When we draw graphs, we will draw a solid red line x —— y if there is an edge between x and y, and a green dotted line x ····· y otherwise.

▶ **Example 21.** Let φ be the linear formula $w \wedge (x \wedge (y \vee z))$. $\mathcal{W}(\varphi)$ is the following graph:



Note that linear formulae equivalent up to associativity and commutativity have the same relation web, since $\sim_{\mathsf{ac}}$ does not affect the lccs. For instance, if $\psi = (w \wedge x) \wedge (z \vee y)$, then $\mathcal{W}(\psi)$ is still just the relation web above. In fact, we also have the converse:

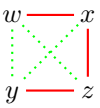
▶ **Proposition 22** (E.g., [15], Proposition 3.5)**.** *Given linear formulae φ and ψ, $\varphi \sim_{\mathsf{ac}} \psi$ if and only if $\mathcal{W}(\varphi) = \mathcal{W}(\psi)$.*

Thus relation webs are natural representations of equivalence classes of linear formulae modulo associativity and commutativity.

It is easy to see that the image of $\mathcal{W}$ is just the *cographs*. A **cograph** is either a single node, or has the form $\mathcal{R}$ —— $\mathcal{S}$ or $\mathcal{R}$ ····· $\mathcal{S}$ for cographs $\mathcal{R}$ and $\mathcal{S}$.[9] A **cograph decomposition** of a cograph $\mathcal{R}$ is just a definition tree according to these construction rules (its "cotree"), from which we may easily extract a linear formula with web $\mathcal{R}$. Note from Example 21 that the cograph decomposition of a relation web need not be unique, since formulae equivalent modulo associativity and commutativity have the same relation web.

Cographs admit an elegant *local* characterisation by means of forbidden subgraphs:

▶ **Definition 23.** *P_4 is the following graph:*



*A graph G is P_4-**free** if none of its induced[10] subgraphs are isomorphic to P_4.*

---

[9] Formally, $\mathcal{R}$ —— $\mathcal{S}$ has as nodes the disjoint union of the nodes of $\mathcal{R}$ and the nodes of $\mathcal{S}$; edges within the $\mathcal{R}$ component are inherited from $\mathcal{R}$ and similarly for $\mathcal{S}$; there is also an edge between every node in $\mathcal{R}$ and every node in $\mathcal{S}$. $\mathcal{R}$ ····· $\mathcal{S}$ is defined similarly, but without the last clause.

[10] An induced subgraph is one whose edges are just those of G restricted to a subset of the nodes.

▶ **Proposition 24** (E.g. [17, 28]). *A graph is a cograph if and only if it is P_4-free. Thus, relation webs are just the P_4-free graphs whose nodes are variables.*

Note, in particular, that this characterisation gives us an easy way to check whether a graph is the web of some formula: just check every 4-tuple of nodes for a P_4. What is more, we may also verify several semantic properties of linear inferences, such as validity and triviality, directly at the level of relation webs:

▶ **Proposition 25** (Follows from Proposition 4.4 and Theorem 4.6 in [15]). *Let φ and ψ be linear formulae on the same set of variables. $\varphi \to \psi$ is a valid linear inference if and only if for every maximal clique P of $\mathcal{W}(\varphi)$, there is some $Q \subseteq P$ such that Q is a maximal clique of $\mathcal{W}(\psi)$.*

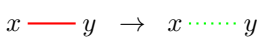
▶ **Proposition 26** (E.g., [15], Proposition 5.7). *Let φ and ψ be linear formulae on the same variables. $\varphi \to \psi$ is a linear inference that is trivial at x if and only if for every maximal clique P of $\mathcal{W}(\varphi)$, there is some $Q \subseteq P \setminus \{x\}$ such that Q is a maximal clique of $\mathcal{W}(\psi)$.*

Note that the criterion for triviality is a strict strengthening of that for validity, as we would expect. For both of the results above, there is a *dual* characterisation in terms of *maximal stable sets* instead of maximal cliques. For instance, the characterisation of validity morally states "whenever φ evaluates to 1, then ψ evaluates to 1". The dual characterisation is that for every maximal stable set Q of $\mathcal{W}(\psi)$ there is a maximal stable set P of $\mathcal{W}(\varphi)$ with $P \subseteq Q$, which morally states "whenever ψ evaluates to 0, then φ evaluates to 0". We will not make use of these dual characterisations in this work.

▶ **Example 27** (Validity of switch and medial, triviality of mix). The switch and medial inferences can be construed as the following "graph rewrite" rules on relation webs, respectively:

It is easy to see that the validity criterion of Proposition 25 holds for each of these rules. For s, the maximal cliques $\{x, y\}$ and $\{x, z\}$ in the LHS are mapped to $\{x, y\}$ and $\{z\}$ in the RHS respectively. For m, the maximal cliques $\{w, x\}$ and $\{y, z\}$ in the LHS are mapped to themselves in the RHS.

Now consider the trivial inference $x \wedge y \to x \vee y$, construed as the graph rewrite rule:



We can easily verify the criterion for triviality at x from Proposition 26 since the only maximal clique on the LHS, $\{x, y\}$ has $\{y\} \subseteq \{x, y\} \setminus \{x\}$ as a maximal clique on the RHS.

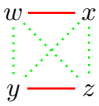
▶ Remark 28. With the results in this section, the notation for inferences between formulae can be equally used for relation webs. For example, for webs $\mathcal{R}$ and $\mathcal{S}$, we can write $\mathcal{R} \rightsquigarrow_{\mathsf{ms}} \mathcal{S}$ is valid to mean that the inference between (any choice of) the underlying linear formulae is an instance of $\rightsquigarrow_{\mathsf{ms}}$, and $\mathcal{R} \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \mathcal{S}$ to mean the there is a derivation from switch and medial between the underlying formulae. Since these relations are invariant under associativity and commutativity, they are independent of the particular cograph decomposition chosen.

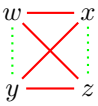
Furthermore, to prove Theorem 11, it is sufficient to show that for all webs $\mathcal{R}$ and $\mathcal{S}$ with size less than 8, if $\mathcal{R} \to \mathcal{S}$ is valid and non-trivial then $\mathcal{R} \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \mathcal{S}$.

The final component needed to be able to work fully with webs is a way to check if a given inference is an instance of switch or medial. Such characterisations exist:

▶ **Proposition 29** ([28, Theorem 5]). *Let $\mathcal{R} \to \mathcal{S}$ represent a constant-free negation-free non-trivial linear inference. Then $\mathcal{R} \to \mathcal{S}$ is derivable from medial if and only if:*

- *Whenever x —— y in $\mathcal{R}$, also x —— y in $\mathcal{S}$.*
- *Whenever x ····· y in $\mathcal{R}$ but x —— y in $\mathcal{S}$ there exists w and z such that*



is an induced subgraph of $\mathcal{R}$ and


is an induced subgraph of $\mathcal{S}$.

The second condition can, in fact, be replaced by simply requiring that $\mathcal{R} \to \mathcal{S}$ is valid [15, Theorem 7.5]. A relation web characterisation for switch derivability can also be found in [28, Theorem 6.2], however we do not use it in our implementation.

5 Implementation

As stated in previous sections, Theorem 11 is proved using a computational search. In this section we describe the algorithm used to search for $\{\mathsf{s}, \mathsf{m}\}$-independent inferences, as well as some of the optimisations we employ so that this search finishes in a reasonable time. Many of these optimisations may be of self-contained theoretical interest.

The implementation is written in Rust,[11] which offers a combination of good performance (both in terms of speed and memory management) but also provides a variety of high level abstractions such as algebraic data types. Furthermore, it has built-in support for iterators, allowing the code to be written in a more functional style, and has a built-in testing framework, meaning that sanity checks can be built into the code base. The code is available at [25] and has been split into two parts: a *library* containing types for undirected linear graphs and formulae and some operations on them, and an *executable* which implements the search algorithm using this library as a base.

5.1 Library

The library portion of the implementation defines methods for working with relation webs, as well as the ability to convert formulae to relation webs and vice versa. The majority of the library consists of the `LinGraph` trait, which is an interface for types that can be treated as undirected graphs. This allows us to query the edges between variables as well as perform more involved operations such as checking whether a graph is P_4-free. We may also ask whether a pair of relation webs forms a valid linear inference and check whether the inference is trivial using Propositions 25 and 26.

Storing graphs and relation webs. The library was designed with the intention of storing graphs as compactly as possible. Therefore there are implementations of `LinGraph` which pack the data (a series of bits for whether there exists an edge between each pair of nodes) into various integer types. The implementation is given for unsigned 8 bit, 16 bit, 32 bit (which can store up to 8 variable graphs), 64 bit, and 128 bit integers. Furthermore there is an implementation using vectors (variable length arrays) of Boolean values, which is less memory efficient but can store relation webs of arbitrary size. A further improvement could be to use an external library implementing bit arrays to make a memory efficient, yet infinitely scalable implementation.

[11] `https://www.rust-lang.org/`

Checking an inference between graphs. In order to implement linear inference checking, we use a data type representing maximal cliques of a relation web, which we represent as the trait `MClique`. It is possible to use Rust's inbuilt `HashSet`[12] to do this but, as above, a more memory efficient solution is provided where we store the data in a single integer, with each bit determining whether a node is contained in the clique. For example a maximal clique on an 8 node graph can be encoded into a single byte. While checking for linear inferences and triviality, the main operation on maximal cliques is asking whether one is a subset of the other. This operation can be carried out very quickly using bitwise operations. Lastly we also need a way to generate the maximal cliques of a relation web. This is done using the Bron-Kerbosch algorithm [4], which is fast enough for our purposes (as we are only finding the maximal cliques of relatively small graphs).

Working with isomorphism. There is also code for working with isomorphisms of graphs, which is used in the search algorithm to shrink the search space further. This is implemented as a module where permutations and operations on these permutations are defined, as well as having the ability to apply a permutation to the nodes of a graph, to get a new but isomorphic graph.

Generating all P_4-free graphs. The library also has a function that allows all P_4-free graphs of a certain size to be generated. The naive algorithm for doing this which simply generates all graphs and checks each one for being P_4-free is computationally infeasible for graphs with more than a few variables, as the number of graphs scales superexponentially with the number of variables (for instance there are 2^{21} 7-variable relation webs). Instead, we use a recursive algorithm that generates all P_4-free graphs of size n by first generating the P_4-free graphs of size $n-1$ and then checking all possible extensions of these graphs to see if they are P_4-free. Correctness of this procedure is due to the fact that induced subgraphs of a P_4-free are themselves P_4-free. In fact, a further optimisation is also added: when we check whether the extensions are P_4-free, it is sufficient to only check if subsets of the nodes containing the added node are not isomorphic to P_4, instead of checking every subset.

Sanity checks. Finally, the library also contains some automated tests used as sanity checks on the code, which may be used to check various implementations against each other.

5.2 Search algorithm

The main part of the implementation is a search algorithm to find logically minimal non-trivial inferences between relation webs that are not derivable from switch and medial. The search algorithm functions in multiple phases. After each phase the results are serialised and saved to disk so that the algorithm can be restarted from this point.

Phase 1: generating P_4-free graphs on n nodes. Suppose we are searching for $\{\mathsf{s}, \mathsf{m}\}$-independent linear inferences between webs on n variables. The first phase, as described in the previous section, is to gather all P_4-free graphs with n nodes.

[12] `https://doc.rust-lang.org/std/collections/struct.HashSet.html`

Phase 2: identifying isomorphism classes and canonical representatives. To describe the second phase we need to introduce some new notions. Without loss of generality, we will assume henceforth that the variable set is given by $\mathcal{V} = \{0, \ldots, n-1\}$.

▶ **Definition 30.** *Note that the function* $\iota : \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid x < y\} \to \mathbb{N}$ *given by* $\iota(x, y) = x + \sum_{i<y} i$ *is a bijection. Define the* ***numerical representation*** *of a linear graph* $\mathcal{R}$*, written* $N(\mathcal{R})$*, to be the natural number whose* $\iota(x, y)^{th}$ *least significant bit is* 1 *if and only if* $(x, y) \in \mathcal{R}$*.*

This is the encoding used to store graph in integers as described in the previous section.

▶ **Definition 31.** *Given a bijection* $\rho : \mathcal{V} \to \mathcal{V}$*, we write* $\rho(\mathcal{R})$ *for the graph on* $\mathcal{V}$ *with edges* $(\rho(x), \rho(y))$ *for each edge* $(x, y) \in \mathcal{R}$*.* $\mathcal{R}$ *and* $\mathcal{S}$ *are* ***isomorphic*** *if* $\mathcal{S} = \rho(\mathcal{R})$*, for some bijection* $\rho : \mathcal{V} \to \mathcal{V}$*, in which case* ρ *is called an* ***isomorphism*** *from* $\mathcal{R}$ *to* $\mathcal{S}$*.*

As isomorphism is an equivalence relation, we can partition the set of P_4-free graphs into isomorphism classes. It can readily be checked that N (from Definition 30) is injective and can therefore be used to induce a strict total ordering on graphs. Say that a relation web is **least** if it is the smallest element in its isomorphism class (with respect to this ordering induced from N).

The second phase of the algorithm is to identify these least relation webs, as well as identify the isomorphism between every relation web and its isomorphic least relation web. It will become clear why this data is needed later on in the section. To obtain this, first the relation webs are sorted (by numerical representation) and then, taking each graph $\mathcal{R}$ in turn, applying every possible permutation to its nodes, and seeing if any result in a smaller web (with respect to N). If none do then we record it as a least relation web (with the identity isomorphism). Otherwise suppose it is isomorphic to $\mathcal{R}'$ with isomorphism ρ where $N(\mathcal{R}') < N(\mathcal{R})$. As we are checking graphs in order, we must already know that $\mathcal{R}'$ is isomorphic to least graph $\mathcal{R}''$ with isomorphism π. Then we can record $\mathcal{R}$ as being isomorphic to $\mathcal{R}''$ with isomorphism $\pi \circ \rho$. This allows us to use the following lemma.

▶ **Lemma 32.** *The statement of Theorem 11 follows from the following: for any valid non-trivial logically minimal inference* $\mathcal{R} \to \mathcal{S}$ *on* $n < 8$ *variables, where* $\mathcal{R}$ *is least, we have* $\mathcal{R} \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \mathcal{S}$*.*

Proof. To show the statement of Theorem 11, let $\mathcal{R}$ and $\mathcal{S}$ be relation webs on n variables and suppose $\mathcal{R} \to \mathcal{S}$ is a non-trivial linear inference (cf. Remark 28). By Lemma 16, we can further assume that $\mathcal{R} \to \mathcal{S}$ is logically minimal. Then let ρ be an isomorphism from $\mathcal{R}$ to $\mathcal{R}'$ least isomorphic to $\mathcal{R}$, and let $\mathcal{S}' = \rho(\mathcal{S})$. Then $\mathcal{R} \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \mathcal{S}$ if and only if $\mathcal{R}' \overset{*}{\rightsquigarrow}_{\mathsf{ms}} \mathcal{S}'$, as required. ◀

The above lemma allows us to only search inferences from least webs to arbitrary webs. This increases the speed of the search greatly as it turns out there are relatively few least webs. For example, there are 78416 P_4-free graphs with 7 variables with only 180 of them being least (the number of isomorphism classes). Note that we may not similarly restrict the RHS of inferences to least webs. This means we need to know the maximal cliques of every P_4-free graph to determine whether there are inferences between them.

Phase 3: generating all maximal cliques. In phase three we generate all the maximal cliques of the graphs found in phase one and store them so that they do not need to be recomputed every time we check a linear inference. As we can store each maximal clique in a single byte, storing all this data is feasible.

Phase 4: generating "least" linear inferences. With the maximal clique data, phase four of generating a list of all valid linear inferences (from a least web to an arbitrary web) can be easily done by iterating through all possible combinations and checking them using Proposition 25.

Phase 5: checking for non-triviality. Similarly phase five of checking which of these inferences are non-trivial is also simple using Proposition 26. This data is stored in a `HashMap` of sets for quick indexing.

Phase 6: restricting to logically minimal inferences. Phase six is now to restrict our inferences to only those that are logically minimal. Write $\Phi_{\mathcal{R}}$ be the set of webs distinct from $\mathcal{R}$ that $\mathcal{R}$ (non-trivially) implies. We calculate, for a least web $\mathcal{R}$, the set $M_{\mathcal{R}}$ of webs $\mathcal{S}$ with $\mathcal{R} \to \mathcal{S}$ a logically minimal linear inference using the identity:

$$M_{\mathcal{R}} = \Phi_{\mathcal{R}} \setminus \bigcup_{\mathcal{R}' \in \Phi_{\mathcal{R}}} \Phi_{\mathcal{R}'}$$

Note that to calculate this, we need to be able to generate $\Phi_{\mathcal{R}}$ for arbitrary (i.e. not necessarily least) webs. This is where the isomorphism data stored in phase two becomes useful, as if ρ is an isomorphism from $\mathcal{R}$ to $\mathcal{R}'$, with $\mathcal{R}'$ least, we can use,

$$\Phi_{\mathcal{R}} = \{\rho^{-1}(\mathcal{S}) \mid \mathcal{S} \in \Phi_{\mathcal{R}'}\}$$

to generate $\Phi_{\mathcal{R}}$, where we already have $\Phi_{\mathcal{R}'}$. In the implementation, we generate each $\Phi_{\mathcal{R}}$ on the fly (from $\Phi_{\mathcal{R}'}$), though we could have pre-generated all of these, which might provide further speedup for this phase.

Phase 7: checking for switch-medial derivability. The last phase is to check the remaining inferences, of which there are now few enough to feasibly do so. Logically minimal inferences have one further benefit: a logically minimal inference (and in fact any $\{\mathsf{s}, \mathsf{m}\}$-minimal inference) $\varphi \to \psi$ is derivable from switch and medial if and only if it is derivable from a *single* switch or medial step. To check if it is a medial we can use the criterion for medial derivability from Proposition 29.

To check if the inference $\mathcal{R} \to \mathcal{S}$ is a switch, we simply run through all possible cograph decompositions of $\mathcal{R}$ and check if any of the possible switch applications yields $\mathcal{S}$. It would have been possible to use the criterion for switch derivability from [28] (mentioned at the end of Section 4), but running through possible partitions of the nodes of $\mathcal{R}$ was fast enough and easier to implement.

Evaluation and main results. After running all phases on 7 variables, we found that there were 78416 P_4-free graphs of which 180 were least. There were 35110 non-trivial inferences from a least web to an arbitrary web of which 1352 were minimal. Of these minimal inferences, 968 were an instance of switch, 384 were an instance of medial, and there were no other inferences, which completes the proof of Theorem 11.

Furthermore, the algorithm was fast enough to run on 8 variables, where there were 1320064 P_4-free graphs of which 522 were least. There were 514486 non-trivial inferences from a least web to an arbitrary web of which 5364 were minimal, Of these, 3506 were an instance of switch, 1770 were an instance of medial, and there were 88 other inferences. After quotienting out by isomorphism (as restricting to inferences from least graphs does not rule out self isomorphisms on the LHS of the inference), we were left with 3 inferences, of which two were dual to each other leaving the logically minimal $\{\mathsf{s}, \mathsf{m}\}$-independent inferences given in Section 3. These give a proof of Theorem 7, the main theorem of this paper.

6 Conclusions

In this work we undertook a computational approach towards the classification of linear inferences. To this end we succeeded in exhausting the linear inferences up to 8 variables, showing that there are two (distinct) 8 variable linear inferences that are independent of switch and medial. One of these new inferences contradicts a Conjecture 7.9 from [15]. Conversely, all linear inferences on 7 variables or fewer are already derivable using switch and medial.

We point out that it should be possible to adapt our implementation to a variety of logics and, in particular, graph-based systems such as those from [2, 1, 7]. This would be an interesting avenue for future work.

References

1 Matteo Acclavio, Ross Horne, and Lutz Straßburger. An analytic propositional proof system on graphs. *CoRR*, abs/2012.01102, 2020. arXiv:2012.01102.

2 Matteo Acclavio, Ross Horne, and Lutz Straßburger. Logic beyond formulas: A proof system on graphs. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 38–52. ACM, 2020. doi:10.1145/3373718.3394763.

3 Andreas Blass. A game semantics for linear logic. *Ann. Pure Appl. Log.*, 56(1-3):183–220, 1992. doi:10.1016/0168-0072(92)90073-9.

4 Coenraad Bron and Joep Kerbosch. Finding all cliques of an undirected graph (algorithm 457). *Commun. ACM*, 16(9):575–576, 1973.

5 Kai Brünnler. *Deep inference and symmetry in classical proofs*. PhD thesis, Dresden University of Technology, Germany, 2003. URL: http://hsss.slub-dresden.de/hsss/servlet/hsss.urlmapping.MappingServlet?id=1064911987703-3819.

6 Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In Robert Nieuwenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 8th International Conference, LPAR 2001, Havana, Cuba, December 3-7, 2001, Proceedings*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer, 2001. doi:10.1007/3-540-45653-8_24.

7 Cameron Calk, Anupam Das, and Tim Waring. Beyond formulas-as-cographs: an extension of boolean logic to arbitrary graphs. *CoRR*, abs/2004.12941, 2020. arXiv:2004.12941.

8 Stephen A. Cook and Robert A. Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In Robert L. Constable, Robert W. Ritchie, Jack W. Carlyle, and Michael A. Harrison, editors, *Proceedings of the 6th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1974, Seattle, Washington, USA*, pages 135–148. ACM, 1974. doi:10.1145/800119.803893.

9 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. doi:10.2307/2273702.

10 Anupam Das. On the proof complexity of cut-free bounded deep inference. In Kai Brünnler and George Metcalfe, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 20th International Conference, TABLEAUX 2011, Bern, Switzerland, July 4-8, 2011. Proceedings*, volume 6793 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2011. doi:10.1007/978-3-642-22119-4_12.

11 Anupam Das. Rewriting with Linear Inferences in Propositional Logic. In Femke van Raamsdonk, editor, *24th International Conference on Rewriting Techniques and Applications (RTA 2013)*, volume 21 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 158–173, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.RTA.2013.158.

12 Anupam Das. An unavoidable contraction loop in monotone deep inference, 2017. URL: http://cs.bath.ac.uk/ag/das/con-loop.pdf.

13 Anupam Das. A new linear inference of size 8. The Proof Theory Blog, June 2020. URL: https://prooftheory.blog/2020/06/25/new-linear-inference/.

14 Anupam Das and Lutz Straßburger. No complete linear term rewriting system for propositional logic. In Maribel Fernández, editor, *26th International Conference on Rewriting Techniques and Applications, RTA 2015, June 29 to July 1, 2015, Warsaw, Poland*, volume 36 of *LIPIcs*, pages 127–142. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.RTA.2015.127.

15 Anupam Das and Lutz Straßburger. On linear rewriting systems for boolean logic and some applications to proof theory. *Log. Methods Comput. Sci.*, 12(4), 2016. doi:10.2168/LMCS-12(4:9)2016.

16 Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987. doi:10.1016/0304-3975(87)90045-4.

17 Alessio Guglielmi. A system of interaction and structure. *ACM Trans. Comput. Log.*, 8(1):1, 2007. doi:10.1145/1182613.1182614.

18 Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Log. Methods Comput. Sci.*, 4(1), 2008. doi:10.2168/LMCS-4(1:9)2008.

19 Tom Gundersen. *A General View of Normalisation through Atomic Flows*. PhD thesis, University of Bath, UK, 2009. URL: https://tel.archives-ouvertes.fr/tel-00441540.

20 Giorgi Japaridze. Introduction to cirquent calculus and abstract resource semantics. *CoRR*, abs/math/0506553, 2005. arXiv:math/0506553.

21 Giorgi Japaridze. Elementary-base cirquent calculus I: parallel and choice connectives. *CoRR*, abs/1707.04823, 2017. arXiv:1707.04823.

22 Jan Krajícek. The cook-reckhow definition. *CoRR*, abs/1909.03691, 2019. arXiv:1909.03691.

23 Lê Thành Dung Nguyên and Thomas Seiller. Coherent interaction graphs. In Thomas Ehrhard, Maribel Fernández, Valeria de Paiva, and Lorenzo Tortora de Falco, editors, *Proceedings Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity-TLLA@FLoC 2018, Oxford, UK, 7-8 July 2018*, volume 292 of *EPTCS*, pages 104–117, 2018. doi:10.4204/EPTCS.292.6.

24 Alex Rice. Linear inferences of size 7. The Proof Theory Blog, October 2020. URL: https://prooftheory.blog/2020/10/01/linear-inferences-of-size-7/.

25 Alex Rice. `lin_inf` rust crate. https://github.com/alexarice/lin_inf, 2021.

26 Lutz Straßburger. A local system for linear logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 9th International Conference, LPAR 2002, Tbilisi, Georgia, October 14-18, 2002, Proceedings*, volume 2514 of *Lecture Notes in Computer Science*, pages 388–402. Springer, 2002. doi:10.1007/3-540-36078-6_26.

27 Lutz Straßburger. *Linear logic and noncommutativity in the calculus of structures*. PhD thesis, Dresden University of Technology, Germany, 2003. URL: http://hsss.slub-dresden.de/hsss/servlet/hsss.urlmapping.MappingServlet?id=1063208959250-7293.

28 Lutz Straßburger. A characterization of medial as rewriting rule. In Franz Baader, editor, *Term Rewriting and Applications, 18th International Conference, RTA 2007, Paris, France, June 26-28, 2007, Proceedings*, volume 4533 of *Lecture Notes in Computer Science*, pages 344–358. Springer, 2007. doi:10.1007/978-3-540-73449-9_26.

29 Lutz Straßburger. Personal communication, 2012.

30 Lutz Straßburger. Extension without cut. *Ann. Pure Appl. Log.*, 163(12):1995–2007, 2012. doi:10.1016/j.apal.2012.07.004.

31 Alvin Šipraga. An automated search of linear inference rules. Summer research project. Supervised by Alessio Guglielmi and Anupam Das, 2012. URL: http://arcturus.su/mimir/autolininf.pdf.

A Further proofs and examples

Proof sketch of Proposition 2. Write $\rightsquigarrow$ for the rewriting relation obtained by orienting every pair of (3) left-to-right. Clearly $\rightsquigarrow$ is terminating since each step decreases formula size. For confluence, note that every critical pair must reduce to the same constant:

$$\bot \vee \bot \rightsquigarrow \bot \quad \bot \vee \top \rightsquigarrow \top \quad \top \vee \bot \rightsquigarrow \top \quad \top \vee \top \rightsquigarrow \top$$
$$\bot \wedge \bot \rightsquigarrow \bot \quad \bot \wedge \top \rightsquigarrow \bot \quad \top \wedge \bot \rightsquigarrow \bot \quad \top \wedge \top \rightsquigarrow \top$$

◀

A.1 Recovering an 8 variable inference

The reason for writing the variation (6) in Section 3.1 instead of the one originally presented in [11] is that it allows us to recover one of the new 8-variable inferences, by a particular reduction first noticed in a blog post [13].

By setting $x' = u' = \neg u$ in (6) and simplifying, we obtain the linear inference:

$$\begin{array}{ll} & (z \vee (w \wedge w')) \wedge (y \vee y') \wedge ((x \wedge x') \vee z') \\ \rightarrow & (z \wedge (x \vee y)) \vee (w' \wedge x') \vee ((w \vee y') \wedge z') \end{array}$$

Again, the inference above is not $\{\mathsf{s}, \mathsf{m}\}$-minimal, since there are two possible applications of switch to the LHS that nonetheless imply the RHS:

$$((z \wedge (y \vee y')) \vee (w \wedge w')) \wedge ((x \wedge x') \vee z') \quad \text{or} \quad (z \vee (w \wedge w')) \wedge ((x \wedge x') \vee ((y \vee y') \wedge z'))$$

Furthermore, are two switch applications leading to the RHS that are nonetheless implied by their respective formulae above:[13]

$$((z \vee (w' \wedge x')) \wedge (x \vee y)) \vee ((w \vee y') \wedge z') \quad \text{or} \quad (z \wedge (x \vee y)) \vee ((w \vee y') \wedge ((w' \wedge x') \vee z'))$$

The two resulting linear inferences are, in fact, isomorphic and indeed $\{\mathsf{s}, \mathsf{m}\}$-minimal, as we shall explain in the next subsection. As we have already mentioned, the fact that this is a *logically minimal* linear inference is shown by means of the implementation presented in Section 5.

A.2 Validity of Equation 7

We consider each assignment that satisfies the LHS and argue that it also satisfies the RHS:

- $\{z, x, x'\}$ satisfies $z \wedge (x \vee y)$.
- $\{z, y, z'\}$ satisfies $z \wedge (x \vee y)$.
- $\{z, y', z'\}$ satisfies $(w \vee y') \wedge ((w' \wedge x') \vee z')$.
- $\{w, w', x, x'\}$ satisfies $(w \vee y') \wedge ((w' \wedge x') \vee z')$.
- $\{w, w', y, z'\}$ and $\{w, w', y', z'\}$ satisfy $(w \vee y') \wedge ((w' \wedge x') \vee z')$.

A.3 Validity of Equation 8

We consider each assignment that satisfies the LHS and argue that it also satisfies the RHS:

- $\{w, w', y, y'\}$ satisfies $w \wedge y$.
- $\{w, w', z, z'\}$ satisfies $w' \wedge z'$ and z.
- $\{x, x', y, y'\}$ satisfies x and $x' \wedge y'$.
- $\{x, x', z, z'\}$ satisfies x and z.

[13] Note that these switch applications were overlooked in the blog post [13].

A.4 {s, m}-independence and {s, m}-minimality of Equation 7

There are two possible medial applications to the subformula $(x \wedge x') \vee ((y \vee y') \wedge z')$ resulting in the following new LHSs:

- $(z \vee (w \wedge w')) \wedge (x \vee y \vee y') \wedge (x' \vee z')$. In this case $\{z, y', x'\}$ is a countermodel.
- $(z \vee (w \wedge w')) \wedge (x \vee z') \wedge (x' \vee y \vee y')$. In this case $\{z, z', x'\}$ is a countermodel.

There are two possible switch applications to the subformula $(y \vee y') \wedge z'$ resulting in the following new LHSs:

- $(z \vee (w \wedge w')) \wedge ((x \wedge x') \vee y \vee (y' \wedge z'))$. In this case $\{w, w', y\}$ is a countermodel.
- $(z \vee (w \wedge w')) \wedge ((x \wedge x') \vee y' \vee (y \wedge z'))$. In this case $\{z, y'\}$ is a countermodel.

Finally any other switch application is on the top-level conjunction, resulting in a formula of the form $z \vee X$, $(w \wedge w') \vee X$, $(x \wedge x') \vee X$ or $((y \vee y') \wedge z') \vee X$, which admits a countermodel $\{z\}$, $\{w, w'\}$, $\{x, x'\}$ or $\{y, z'\}$, respectively.

A.5 {s, m}-independence and {s, m}-minimality of Equation 8

Let us first consider rules applicable to the LHS. There are four possible medial applications, resulting in the following new LHSs:

- $(w \vee x) \wedge (w' \vee x') \wedge ((y \wedge y') \vee (z \wedge z'))$. In this case $\{w, x', y, y'\}$ is a countermodel.
- $(w \vee x') \wedge (w' \vee x) \wedge ((y \wedge y') \vee (z \wedge z'))$. In this case $\{x', w', y, y'\}$ is a countermodel.
- $((w \wedge w') \vee (x \wedge x')) \wedge (y \vee z) \wedge (y' \vee z')$. In this case $\{x, x', y, z'\}$ is a countermodel.
- $((w \wedge w') \vee (x \wedge x')) \wedge (y \vee z') \wedge (y' \vee z)$. In this case $\{w, w', z', y'\}$ is a countermodel.

Any switch application to the LHS must be on the top-level conjunction, and will have the form $(a \wedge a') \vee X$, for $a \in \{w, x, y, z\}$. However, $\{w, w'\}$, $\{x, x'\}$, $\{y, y'\}$ and $\{z, z'\}$ are each countermodels for the RHS.

Now let us consider the possible rule applications leading to the RHS. There are two possible medial instances, coming from the following new RHSs:

- $(w \wedge y) \vee (x \wedge x' \wedge y') \vee (w' \wedge z' \wedge z)$. In this case $\{x, x', z, z'\}$ is a countermodel.
- $(w \wedge y) \vee (x \wedge z) \vee (w' \wedge z' \wedge x' \wedge y')$. In this case $\{w, w', z, z'\}$ is a countermodel.

Now let us consider the switch instances:

- If the contractum of the switch is $x \vee (w' \wedge z')$, then $\{x, x', y, y'\}$ is a countermodel.
- If the contractum of the switch is $(x' \wedge y') \vee z$, then $\{w, w', z, z'\}$ is a countermodel.
- If the redex of the switch has the form $w \wedge X$ or $y \wedge X$, then $\{x, x', z, z'\}$ is a countermodel.
- If the redex of the switch has the form $X \wedge (x \vee (w' \wedge z'))$ or $X \wedge ((x' \wedge y') \vee z)$, then $\{w, w', y, y'\}$ is a countermodel.