On the relative proof complexity of deep inference via atomic flows

Das, Anupam

ON THE RELATIVE PROOF COMPLEXITY OF DEEP INFERENCE VIA ATOMIC FLOWS

ANUPAM DAS

École Normale Supérieure de Lyon (ENS Lyon), France
e-mail address: anupam.das@ens-lyon.fr

ABSTRACT. We consider the proof complexity of the minimal complete fragment, KS, of standard deep inference systems for propositional logic. To examine the size of proofs we employ *atomic flows*, diagrams that trace structural changes through a proof but ignore logical information. As results we obtain a polynomial simulation of versions of Resolution, along with some extensions. We also show that these systems, as well as bounded-depth Frege systems, cannot polynomially simulate KS, by giving polynomial-size proofs of certain variants of the propositional pigeonhole principle in KS.

1. INTRODUCTION

Deep inference is a relatively recent proof methodology whose systems differ from other formalisms by allowing derivations themselves to be composed by logical connectives. One of its main features is *locality*, i.e. inference steps can be checked in constant time, a property that is impossible to achieve in Gentzen systems [Brü03]. In recent years there has been an increasing interest in the proof complexity of deep inference [BG09] [Jeř09] [Str12] [BGGP10] [Das11] [Das14], in particular the weaker systems initially introduced by Brünnler and Tiu [BT01]. Perhaps the most notable result is that a certain system, denoted $\mathsf{KS} \cup \{\mathsf{c}\!\uparrow\}$, quasipolynomially[1] simulates Frege systems [Jeř09] [BGGP10]. It is conjectured that this can be improved to a polynomial simulation, and so proving nontrivial lower bounds for $\mathsf{KS} \cup \{\mathsf{c}\!\uparrow\}$ is likely equivalent to proving them for Frege systems, a task which has escaped proof complexity theorists for years.

However this quasipolynomial simulation relies crucially on the presence of "dag-like behaviour", manifested in deep inference by a particular rule, *cocontraction*: $\mathsf{c}\!\uparrow \dfrac{A}{A \wedge A}$. Without it we have a minimal complete system closed under deep inference, KS. This system is free of compression mechanisms, in the sense that a proof of a conjunction can be 'partitioned' into proofs of each conjunct, unlike proofs in systems that are dag-like or contain cut. This is explained further in [Das12a].

[1] A quasipolynomial in n is a function $2^{\log^k n}$.

It is conjectured that KS is unable to polynomially simulate $\mathsf{KS} \cup \{\mathsf{c}{\uparrow}\}$ [BG09] [BGGP10] [Das11] [Str12], raising the question of where exactly it fits in the hierarchy of proof systems.

Atomic flows are diagrams that track structural changes in a proof (duplication, creation and destruction of atoms) but ignore logical information. In this work they serve as a useful abstraction because of certain rewriting procedures on them which can be used to manipulate derivations soundly without any mention of logical syntax. Atomic flows are introduced formally in [GG08] and a comprehensive account can be found in [Gun09].

In this paper we focus on upper bounds and simulations to demonstrate the relative strength of KS. The starting points in our arguments are proofs in a system obtained by extending $\mathsf{KS} \cup \{\mathsf{c}{\uparrow}\}$ by the *coweakening* rule: $\mathsf{w}{\uparrow}\dfrac{A}{\top}$; the resulting system $\mathsf{KS} \cup \{\mathsf{c}{\uparrow}, \mathsf{w}{\uparrow}\}$ is denoted KS^+ in this paper. We then appeal to sound rewriting rules on the atomic flows of these proofs to show that cocontraction and coweakening steps can, in certain cases, be eliminated from a proof in polynomial time.

It is worth mentioning here that the addition of coweakening makes little difference to proof complexity, indeed it is not difficult to see from the rewriting rules in Fig. 1 that coweakening steps can be eliminated in time linear in the size of the proof. Rather the real generators of complexity in our proofs are the interactions between contraction and cocontraction nodes in the atomic flows, as we show in Prop. 3.10 and Lemma 3.13.

In Sect. 4 we give a simple example of how atomic flows can be used to normalise a naïve encoding of truth table proofs in KS^+ to produce a polynomial simulation in KS. As a corollary we obtain a superpolynomial separation of KS from tree-like cut-free Gentzen systems, since they are unable to simulate truth tables [D'A92], a new proof of a result appearing in [BG09].

In Sect. 5 we consider stronger systems; we improve a result of Jeřábek's that KS^+ has polynomial-size proofs of the functional and onto versions of the propositional pigeonhole principle by showing that they can be polynomially transformed into KS proofs of the same conclusions. This immediately entails that cut-free Gentzen systems, Resolution and even bounded-depth Frege systems are exponentially separated from KS by the results of [PBI93] and [KPW95].

In Sect. 6 we consider simulations in KS of other proof systems. There is already a naïve simulation of tree-like cut-free Gentzen sequent calculi, appearing in [BG09], and what amounts to a polynomial simulation of 'Resolution with multisets' is outlined in [Gug03]. Here we formalise the latter result and also give a polynomial simulation of tree-like Resolution systems, even when sets are the basic data structure. We show that both simulations extend to so-called Resolution(f) systems, introduced by Krajíček in [Kra01] and known to be strictly stronger than usual versions of Resolution [SBI02] [EGM04].

This paper is a full version of [Das12b], and differs from that work as follows:

(1) Full proofs are given where they were brief or omitted previously.
(2) We expand on some of the preliminary work on the complexity of normalisation induced by flow rewriting in Sect. 2. In particular we provide full proofs of termination and confluence for the rewriting system norm and give explicit reduction strategies that achieve the complexity bounds given in the previous work.
(3) In Sect. 2 we use the definitions and notations found in [GG08] and [Gun09] for atomic flows, to maintain consistency with the existing literature, whereas in [Das12b] self-contained definitions were preferred for brevity.

(4) There were some errors in the statements of results in the previous work, which have been corrected here. In particular the previously stated simulations of dag-like cut-free Gentzen systems and dag-like Resolution systems are incorrect as presented and we have not been able to amend them. Here we only obtain such simulations for certain cases, namely the versions of Resolution in Sect. 6.

2. Deep inference, atomic flows and normalisation

In this section we introduce deep inference systems for propositional logic and *atomic flows*, diagrams that trace the structural changes in a proof. We consider a graph rewriting system norm on flows, corresponding to sound manipulations on proofs, and analyse the complexity of termination in this system.

2.1. **Deep inference.** We consider propositional logic with formulae constructed from literals (propositional variables and their duals), also called *atoms*, over the basis $\{\top, \bot, \wedge, \vee\}$, and use the infix symbol $\equiv$ to denote equivalence of expressions. The variables a, b, c, d range over literals, with $\bar{a}, \bar{b}, \ldots$ denoting their duals, and A, B, C, D range over formulae; both sets of variables may include subscripts or superscripts as necessary.

For clarity we use square brackets $[,]$ for disjunctions and round ones $(,)$ for conjunctions. We generally omit external brackets of an expression, and also internal ones under associativity. This does not cause any confusion when it comes to proofs, since any valid bracketing can be reduced to any other by the $=$ rule in Dfn. 2.1.

Note that we do not have a symbol for negation in our language, formulae are always in negation normal form. We may however write $\bar{A}$ to denote the De Morgan dual of a formula A, obtained by the following rules:

$$\bar{\bot} \equiv \top, \quad \bar{\top} \equiv \bot, \quad \bar{\bar{a}} \equiv a, \quad \overline{A \wedge B} \equiv \bar{A} \vee \bar{B}, \quad \overline{A \vee B} \equiv \bar{A} \wedge \bar{B}$$

Definition 2.1 (Rules and systems). An *inference rule* is a sound binary relation on formulae decidable in polynomial time, and a *system* is a set of rules. We define the rules we use below, and the systems $\mathsf{KS} = \{\mathsf{ai}\!\downarrow, \mathsf{aw}\!\downarrow, \mathsf{ac}\!\downarrow, \mathsf{s}, \mathsf{m}\}$, $\mathsf{KS}^+ = \mathsf{KS} \cup \{\mathsf{aw}\!\uparrow, \mathsf{ac}\!\uparrow\}$ and $\mathsf{SKS} = \mathsf{KS}^+ \cup \{\mathsf{ai}\!\uparrow\}$.

	Atomic structural rules		Linear logical rules

$$\mathsf{ai}\!\downarrow \frac{\top}{a \vee \bar{a}} \qquad \mathsf{aw}\!\downarrow \frac{\bot}{a} \qquad \mathsf{ac}\!\downarrow \frac{a \vee a}{a} \qquad \mathsf{s} \frac{A \wedge [B \vee C]}{(A \wedge B) \vee C}$$

$$\quad identity \qquad\quad weakening \qquad\quad contraction \qquad\qquad switch$$

$$\mathsf{ai}\!\uparrow \frac{a \wedge \bar{a}}{\bot} \qquad \mathsf{aw}\!\uparrow \frac{a}{\top} \qquad \mathsf{ac}\!\uparrow \frac{a}{a \wedge a} \qquad \mathsf{m} \frac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]}$$

$$\quad\quad cut \qquad\qquad coweakening \qquad cocontraction \qquad\qquad medial$$

Note in particular our distinction between variables for literals and formulae in the above rules, and between 'structural' and 'logical' rules.

We also have the logical rule $=$ which is obtained by closing the equations below under reflexivity, symmetry, transitivity and by applying context closure. We implicitly assume

that it is contained in every system.

Commutativity	*Associativity*	*Units*
$A \vee B = B \vee A$	$[A \vee B] \vee C = A \vee [B \vee C]$	$A \vee \bot = A$
$A \wedge B = B \wedge A$	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$	$A \wedge \top = A$
		$\top \vee \top = \top$
Context closure: if $A = B$, $\star \in \{\wedge, \vee\}$, then $C \star A = C \star B$		$\bot \wedge \bot = \bot$

A proof that $=$ is decidable in polynomial time can be found in [BG09]. Essentially both formulae are just reduced to some canonical form and then compared. Consequently we often omit occurrences of the $=$ rule in proofs and derivations.

Definition 2.2 (Proofs and derivations). We define derivations and premiss and conclusion functions (pr and cn respectively) inductively.

(1) Each formula A is a derivation with premiss and conclusion A.
(2) If Φ and Ψ are derivations and $\star \in \{\wedge, \vee\}$ then $(\Phi \star \Psi)$ is a derivation with premiss $\mathsf{pr}(\Phi) \star \mathsf{pr}(\Psi)$ and conclusion $\mathsf{cn}(\Phi) \star \mathsf{cn}(\Psi)$.
(3) If Φ and Ψ are derivations and $\rho \dfrac{\mathsf{cn}(\Phi)}{\mathsf{pr}(\Psi)}$ is an instance of some inference rule ρ then $\rho \dfrac{\Phi}{\Psi}$ is a derivation with premiss $\mathsf{pr}(\Phi)$ and conclusion $\mathsf{cn}(\Psi)$.

If $\mathsf{pr}(\Phi) \equiv \top$ then we call Φ a *proof*. If Φ is a derivation where all inference steps are instances of rules in a system $\mathcal{S}$ with premiss A, conclusion B, we write $\Phi \Vert \mathcal{S}\ \begin{smallmatrix} A \\ \\ B \end{smallmatrix}$. Furthermore, if $A \equiv \top$, i.e. Φ is a proof in a system $\mathcal{S}$, we write $\begin{smallmatrix} \Phi \Vert \mathcal{S} \\ B \end{smallmatrix}$.

While our structural rules only have atoms in their premisses and conclusions, the notion of derivation above allows us to extend these to arbitrary formulae, as stated in the proposition below. We often use these 'generic rules' rather than their full derivations for convenience.

Proposition 2.3 (Generic rules). *Each rule below is derivable from* s, m, *and its respective atomic structural rule in polynomial time.*

$$ \mathsf{i}\!\downarrow \frac{\top}{A \vee \bar{A}} \qquad \mathsf{w}\!\downarrow \frac{\bot}{A} \qquad \mathsf{c}\!\downarrow \frac{A \vee A}{A} \qquad \mathsf{i}\!\uparrow \frac{A \wedge \bar{A}}{\bot} \qquad \mathsf{w}\!\uparrow \frac{A}{\top} \qquad \mathsf{c}\!\uparrow \frac{A}{A \wedge A} $$

Proof. See [BT01] for full proofs. Here we just give an example of the case for contraction, since that is the only structural rule of the sequent calculus that cannot be reduced to atomic form [Brü03]. The proof is by induction on the depth of the conclusion of a $\mathsf{c}\!\downarrow$ step.

$$ \mathsf{c}\!\downarrow \frac{[A \vee B] \vee [A \vee B]}{A \vee B} \quad \rightarrow \quad = \frac{[A \vee B] \vee [A \vee B]}{\mathsf{c}\!\downarrow \dfrac{A \vee A}{A} \vee \mathsf{c}\!\downarrow \dfrac{B \vee B}{B}} $$

$$ \mathsf{c}\!\downarrow \frac{(A \wedge B) \vee (A \wedge B)}{A \wedge B} \quad \rightarrow \quad \mathsf{m} \frac{(A \wedge B) \vee (A \wedge B)}{\mathsf{c}\!\downarrow \dfrac{A \vee A}{A} \wedge \mathsf{c}\!\downarrow \dfrac{B \vee B}{B}} $$

Note that the case for cocontraction is dual to this: one can just flip the derivations upside down and replace every formula with its De Morgan dual. $\square$

Definition 2.4 (Complexity). We define the size $|\Phi|$ of a derivation Φ to be the number of atom occurrences in Φ. A system $\mathcal{S}$ *polynomially simulates* a system $\mathcal{T}$ if each proof in $\mathcal{T}$ can be polynomially transformed into a proof in $\mathcal{S}$ of the same conclusion.

2.2. Atomic flows.

We give only an informal definition of atomic flows here, but refer the reader to [GG08], [Gun09] for a formal account of atomic flows.

Definition 2.5 (Atomic flows). For an SKS derivation Φ we define its *atomic flow*, $fl(\Phi)$, to be the diagram obtained by tracing the path of each atom through the derivation, designating the creation, duplication and destruction of atoms by the following corresponding nodes:

We do not have nodes for s, m or = since they do not create, destroy or duplicate any atom occurrences.

More generally an atomic flow, not necessarily of a derivation, is a (vertically) directed graph embedded in the plane generated from the six types of node above.

Atomic flows are considered equivalent up to continuous deformation preserving the (vertical) ordering of connected edges. Note that edges may be *pending* at either end.

We define the size of a flow ϕ, denoted $|\phi|$, to be its number of edges.

In previous works atomic flows have been equipped with a labeling of the edges, or a polarity assignment, for example to avoid the following impossible situation:

Since we are only concerned with the complexity of flows and their transformations we do not include this extra structure; this does not affect the soundness or termination of our rewriting systems, and in fact is crucial in order to obtain confluence, for which labellings of edges can cause problems.

We do, however, insist that edges are vertically directed and so we are often able to talk about one node being 'above' another node. Notice that this order is not generally preserved under deformation, e.g. if two nodes are in disconnected components. Whenever we use this notion in arguments it should be clear that it is being used correctly.

Definition 2.6. A *flow rewriting rule* is an ordered pair of flows, written $\phi \to \psi$. A *flow rewriting system* (FRS) is a set of flow rewriting rules. A *one-step reduction* of a flow ϕ in an FRS r is a flow ψ that is obtainable from ϕ by replacing some induced subgraph that is the left hand side of some rule in r with its right hand side.

Definition 2.7. We define a graph rewriting system norm on flows in Fig. 1.

The system norm is essentially the system c∪w in [GG08], without the rules for i↑. The proof of termination that follows is similar to that for cycle-free flows in [GG08], and the proof of confluence is similar to that in [GGS10].
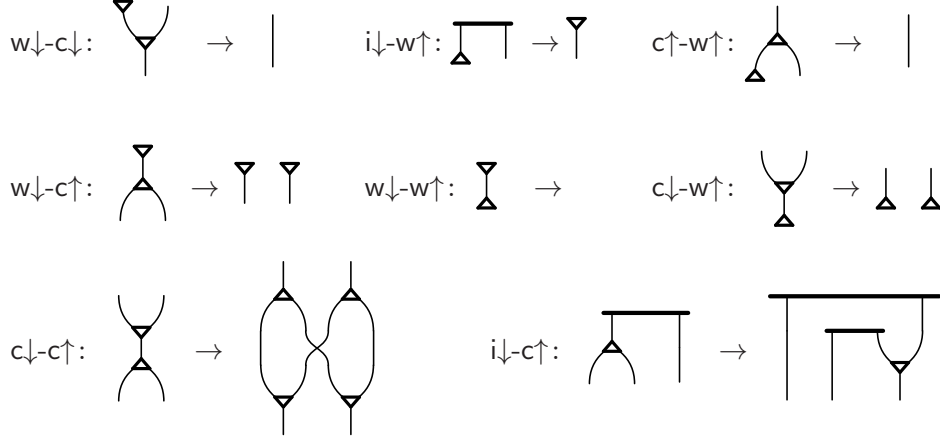
Figure 1: Local rewriting rules for the system norm.

Notation 2.8. Let r be a flow rewriting system. We use the following notation:

- We write $\phi \to_r \psi$ if there is a one-step reduction from a flow ϕ to a flow ψ using a rule in r.
- We denote by $\xrightarrow{*}_r$ the reflexive transitive closure of $\to_r$.
- If a flow ϕ has a unique normal form under r we denote it by $\phi\downarrow_r$.[2]

In all cases we might omit the subscript r if it is clear from context.

Example 2.9. We give an example of a flow associated with a derivation in Fig. 2, as well as a reduction under norm, as defined in Dfn. 2.7, applying w↑ rules first.

The first equality follows by the definition of a flow, the second by deformation and the final by definition again. The intermediate steps are as follows:

(1) Apply c↓-w↑ on the left and c↓-c↑ on the right.
(2) Apply w↓-w↑ on the left, i↓-w↑ in the middle and i↓-c↑ on the right.
(3) Apply w↓-c↑ on the left.
(4) Apply w↓-c↓ twice on the left.

The use of colours in the initial and final flow identifies which edges corresponds to which atoms; in the intermediate flows the colours should aid the reader in reconstructing the corresponding transformations on the derivation.

We now proceed to prove that reducing under norm is terminating and confluent. For this we use the usual critical pair lemma, that one only needs to check that the one-step contracta of every overlapping pair of redexes are joinable in order to conclude local confluence. It is simple to see that this is still valid in the flow rewriting setting, since the only other case to check for flows is when the redexes are disjoint, which is trivial. We point out that Newman's lemma, that any locally confluent terminating system is confluent, is true more generally for any binary relation on any set, and so indeed holds in this setting.

Theorem 2.10. $\xrightarrow{}_{\text{norm}}$ *is terminating and confluent.*

---

[2]Note that we are using downward arrows in both the names of deep inference rules and to denote normal forms under rewriting systems. Unfortunately both notations are standard in their respective literature, however there should be no ambiguity in their usage so hopefully this will cause little confusion.
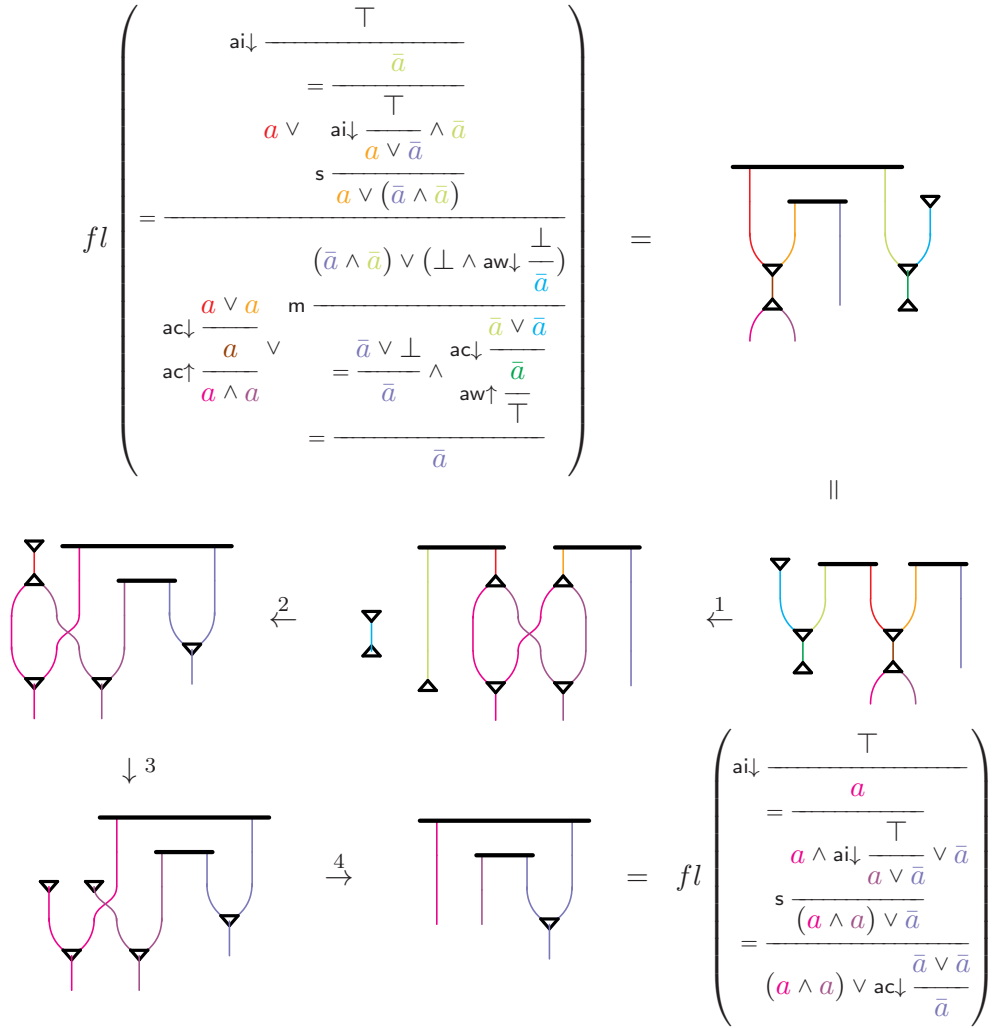
Figure 2: A proof, its flow and a reduction under norm.

Proof. For a node ν in a flow ϕ, let $d(\nu, \phi)$ be the distance of ν from the top of ϕ, i.e. the minimum length of a (vertically) directed path from ν to an ai↓ node, an aw↓ node or an edge with upper end pending. For an atomic structural rule ρ, let $D(\rho, \phi)$ be the sequence of natural numbers that counts how many ρ nodes in ϕ have each d-value, i.e. the sequence (n_i) such that, for each i, n_i is the number of ρ nodes ν in ϕ with $d(\nu, \phi) = i$, and consider the lexicographical ordering $<$ on such sequences, with $(m_i) < (n_i)$ just if $m_k < n_k$ for some k and $m_i \leq n_i$ for $i > k$.

Clearly the rules c↓-c↑, i↓-c↑ and w↓-c↑ strictly reduce the $D(\text{ac}\uparrow, \cdot)$ value of a flow, while the other rules of norm (as well as w↓-c↑) strictly reduce a flow's size, while not increasing the $D(\text{ac}\uparrow, \cdot)$ value. Therefore each application of a norm rule strictly reduces the lexicographical product $D(\text{ac}\uparrow, \cdot) \times | \cdot |$.

Since norm is terminating, it suffices to check local confluence for the critical pairs, which are the following by inspection:

(1) , (2) , (3) , (4) , (5)

(6) , (7) , (8) , (9)
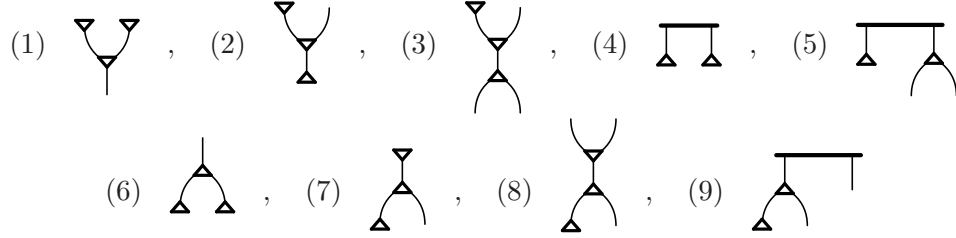
Note that every other overlapping pair can be deformed so that each rule application trivially commutes. We consider each case below.

(1) $\xleftarrow{w\downarrow\text{-}c\downarrow}$ $\xrightarrow{w\downarrow\text{-}c\downarrow}$

(2) $\xleftarrow{w\downarrow\text{-}c\downarrow}$ $\xrightarrow{c\downarrow\text{-}w\uparrow}$ $\xrightarrow{w\downarrow\text{-}w\uparrow}$

(3) $\xleftarrow{w\downarrow\text{-}c\downarrow}$ $\xrightarrow{c\downarrow\text{-}c\uparrow}$ $\xrightarrow{w\downarrow\text{-}c\uparrow}$ $\xrightarrow[w\downarrow\text{-}c\downarrow]{2}$

(4) $\xleftarrow{w\downarrow\text{-}w\uparrow}$ $\xleftarrow{i\downarrow\text{-}w\uparrow}$ $\xrightarrow{i\downarrow\text{-}w\uparrow}$ $\xrightarrow{w\downarrow\text{-}w\uparrow}$

(5) $\xleftarrow{w\downarrow\text{-}c\uparrow}$ $\xleftarrow{i\downarrow\text{-}w\uparrow}$ $\xrightarrow{i\downarrow\text{-}c\uparrow}$ $\xrightarrow{c\downarrow\text{-}w\uparrow}$ $\xrightarrow[i\downarrow\text{-}w\uparrow]{2}$
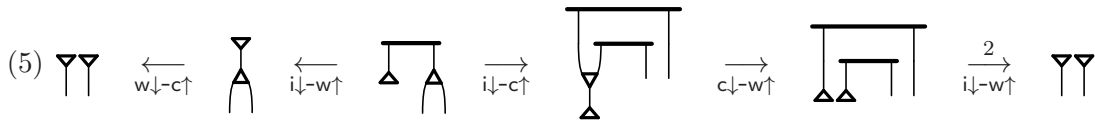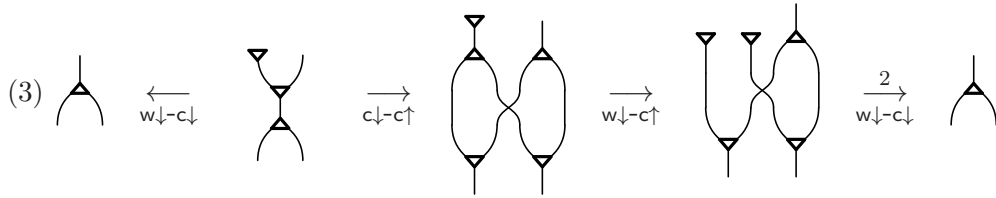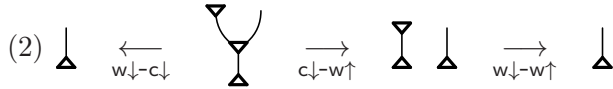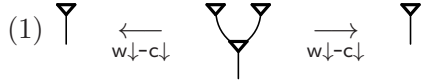
(6) This case is dual to (1).
(7) This case is dual to (2).
(8) This case is dual to (3).

(9) $\xleftarrow{w\downarrow\text{-}c\downarrow}$ $\xleftarrow{i\downarrow\text{-}w\uparrow}$ $\xleftarrow{i\downarrow\text{-}c\uparrow}$ $\xrightarrow{c\uparrow\text{-}w\uparrow}$
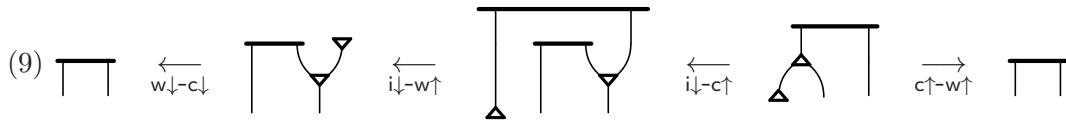
The cases where we appeal to duality follow by simply flipping the indicated reductions upside down and relabeling nodes and reduction steps appropriately. □

The significance of the rewriting system norm is evident from the following results, where we show that reduction steps correspond to sound manipulations of SKS derivations.

Definition 2.11. If $\mathcal{R}$ is a relation on flows we say that $\mathcal{R}$ *lifts polynomially* to SKS if, whenever $(fl(\Phi), \psi) \in \mathcal{R}$, we can construct a derivation Ψ in time polynomial in $|\Phi| + |\psi|$ with the same premiss and conclusion as Φ and flow ψ. If f is a function on flows then we say that f lifts polynomially to SKS just if the relation $f(\cdot) = \cdot$ lifts polynomially to SKS.

Sometimes we simply say that an individual flow rewrite rule is *sound* rather than saying that it lifts polynomially to SKS.

Remark 2.12. In fact a derivation can always be manipulated (preserving premiss, conclusion and flow) so that it has size at most polynomial in the size of its flow, as (essentially) shown in [Das13], so the dependence on size of derivation in Dfn. 2.11 is somewhat redundant. However this is beyond the scope of this work, and it does no harm for us to include this dependence.

Theorem 2.13. $\underset{\text{norm}}{\longrightarrow}$ *lifts polynomially to* SKS.

Proof. See [GG08]. Essentially the proof shows that each local rewrite step on a flow of a derivation induces a sound manipulation of polynomial size on that derivation. We give as an example the case for $\mathsf{c}{\downarrow}\text{-}\mathsf{c}{\uparrow}$, since that rule is of particular interest in this work.

Let $\xi\{\ \}$ denote an arbitrary context, i.e. a formula with a single hole occurring in place of a subformula, and let $\xi\{A\}$ denote the result of substituting a formula A for the hole in $\xi\{\ \}$. The manipulation is as follows:

$$\mathsf{c}{\downarrow}\text{-}\mathsf{c}{\uparrow}: \qquad
\begin{array}{c}
\xi\left\{\mathsf{ac}{\downarrow}\dfrac{a \vee a}{a}\right\} \\[4pt]
\Phi\Big\| \\[4pt]
\zeta\left\{\mathsf{ac}{\uparrow}\dfrac{a}{a \wedge a}\right\}
\end{array}
\qquad \rightarrow \qquad
\begin{array}{c}
\xi\left\{\mathsf{m}\dfrac{\mathsf{ac}{\uparrow}\dfrac{a}{a \wedge a} \vee \mathsf{ac}{\uparrow}\dfrac{a}{a \wedge a}}{\mathsf{ac}{\downarrow}\dfrac{a \vee a}{a} \wedge \mathsf{ac}{\downarrow}\dfrac{a \vee a}{a}}\right\} \\[6pt]
\Phi\{(a/(a \wedge a)\}\Big\| \\[4pt]
\zeta\{a \wedge a\}
\end{array}$$

where $\Phi\{a/(a \wedge a)\}$ is obtained by replacing a by $(a \wedge a)$ everywhere in Φ. $\qquad\square$

Corollary 2.14. *Given an* SKS *derivation* Φ *and a reduction* $fl(\Phi) = \phi_1 \underset{\text{norm}}{\longrightarrow} \cdots \underset{\text{norm}}{\longrightarrow} \phi_n = fl(\Phi){\downarrow}_{\mathsf{norm}}$ *we can construct an* SKS *derivation with the same premiss and conclusion as* Φ *and with flow* $fl(\Phi){\downarrow}_{\mathsf{norm}}$ *in time polynomial in* $|\Phi| + \sum_{i=1}^{n} |\phi_i|$.

Proof. By induction on the length n of a norm derivation. $\qquad\square$

2.3. **Reduction strategies.** We analyse the complexity of normalising a flow under norm, presenting a class of reduction strategies that optimise the size of a $\underset{\text{norm}}{\longrightarrow}$ derivation from a flow to its normal form, up to a polynomial. Our aim is to prove the following theorem:

Theorem 2.15. *The function* $\cdot{\downarrow}_{\mathsf{norm}}$ *mapping a flow to its unique normal form under* $\underset{\text{norm}}{\longrightarrow}$ *lifts polynomially to* SKS.

The result follows from Cor. 2.14 if we can find appropriate reductions with size only polynomially dependent on the initial derivation and normal form of its flow.

Example 2.16. Consider the following flow,
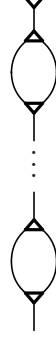
where there are n ac↑ nodes. Notice that this flow has normal form ⊻, but the complexity of a derivation witnessing this can vary significantly. If we apply c↓-c↑ steps first then the length of a derivation to normal form is $\Theta(2^n)$, whereas applying $\{$w↓-c↓, w↓-c↑$\}$ steps first results in length $\Theta(n)$. These bounds follow from later results in this section, but are not difficult to prove directly.

In fact, this sort of unnecessary exponential blowup can always be avoided by applying 'weakening' rules first.

Definition 2.17. We define the following subsystems of norm:
(1) wk = $\{$w↓-c↓, i↓-w↑, c↑-w↑, w↓-c↑, w↓-w↑, c↓-w↑$\}$.
(2) cont = norm \ wk = $\{$c↓-c↑, i↓-c↑$\}$.

Proposition 2.18. *Given a flow ϕ we have:*
(1) *If $\phi \underset{\mathsf{wk}}{\rightarrow} \psi$ then $|\phi| > |\psi|$.*
(2) *If $\phi \underset{\mathsf{cont}}{\rightarrow} \psi$ then $|\phi| < |\psi|$.*

Lemma 2.19. *Given a flow ϕ we have $(\phi\!\downarrow_{\mathsf{wk}})\!\downarrow_{\mathsf{cont}} = \phi\!\downarrow_{\mathsf{norm}}$.*

Proof. By inspecting the rules of cont, we observe that if there are no wk-redexes in a flow ψ and $\psi \underset{\mathsf{cont}}{\rightarrow} \theta$ then there are no wk-redexes in θ. By induction on the length of a $\underset{\mathsf{cont}}{\rightarrow}$-derivation we then have that there are no wk-redexes in $(\phi\!\downarrow_{\mathsf{wk}})\!\downarrow_{\mathsf{cont}}$. But since there are also no cont-redexes, by definition of normal form, and wk $\cup$ cont = norm we can conclude that $(\phi\!\downarrow_{\mathsf{wk}})\!\downarrow_{\mathsf{cont}}$ is already in normal form for norm. $\qquad\square$

We can now give a proof of the main theorem of this section.

Proof of Thm. 2.15. Let Φ be an SKS derivation with flow ϕ whose normal form under norm is ψ, and fix a derivation $\phi = \phi_1 \underset{\mathsf{wk}}{\rightarrow} \cdots \underset{\mathsf{wk}}{\rightarrow} \phi_m = \psi_1 \underset{\mathsf{cont}}{\rightarrow} \cdots \underset{\mathsf{cont}}{\rightarrow} \psi_n = \psi$, which must exist by Lemma 2.19.

By Cor. 2.14 we can construct an SKS derivation Ψ with same premiss and conclusion as Φ and flow ψ in time polynomial in $|\Phi| + \sum_{i=1}^{m} |\phi_i| + \sum_{j=1}^{n} |\psi_i|$. However, by Prop. 2.18

we have that $|\phi_i| < |\phi_1| = |\phi| \leq |\Phi|$ for all i and $|\psi_j| < |\psi_n| = |\psi|$ for all j, and so this construction can be done in time polynomial in $(m + n)(|\Phi| + |\psi|)$.

Finally, notice also by Prop. 2.18 that each $\underset{\mathsf{wk}}{\rightarrow}$ step decreases the size of a flow, so m is bounded above by $|\phi|$ (and so also $|\Phi|$), and each cont step increases the size of a flow, so n is bounded above by $|\psi|$, whence the result follows. $\qquad\square$

3. Complexity of normal forms

In this section we specialise previous results to KS^+ flows, reducing the complexity of norm reduction to counting the number of certain paths in the initial flow. We then give two simple ways of estimating this number, which we use in later sections to obtain complexity results.

Proposition 3.1. *If ϕ is the flow of a KS^+ proof, with normal form ψ under norm, then ψ is free of $\mathsf{aw}{\uparrow}$ and $\mathsf{ac}{\uparrow}$ nodes, i.e. contains just KS nodes.*

Proof. We argue by contradiction. Notice that by $\mathsf{c}{\downarrow}$-$\mathsf{c}{\uparrow}$ we can assume that all $\mathsf{ac}{\uparrow}$ nodes are above all $\mathsf{ac}{\downarrow}$ nodes in ψ, by deformation, and so must have upper end directly connected to an $\mathsf{aw}{\downarrow}$ or $\mathsf{ai}{\downarrow}$ node, since ϕ is associated with a proof. However then ψ can be reduced by either $\mathsf{i}{\downarrow}$-$\mathsf{c}{\uparrow}$ or $\mathsf{w}{\downarrow}$-$\mathsf{c}{\uparrow}$, contradicting normality. An $\mathsf{aw}{\uparrow}$ node, similarly, must be above all $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ nodes by $\mathsf{c}{\downarrow}$-$\mathsf{w}{\uparrow}$ and $\mathsf{c}{\uparrow}$-$\mathsf{w}{\uparrow}$, and so must have upper end directly connected to a $\mathsf{aw}{\downarrow}$ or $\mathsf{ai}{\downarrow}$ node, and again ψ can be reduced by $\mathsf{w}{\downarrow}$-$\mathsf{w}{\uparrow}$ or $\mathsf{i}{\downarrow}$-$\mathsf{w}{\uparrow}$, contradicting normality. $\square$

Notice that the above proposition, along with previous results in this section, allows us to transform KS^+ proofs to KS proofs of the same conclusion. We state this formally in Thm. 3.8 once we have determined more about the complexity of this transformation, which we address in the proceeding results.

Definition 3.2. An ai-*path* is a (simple) path that changes (vertical) direction only at $\mathsf{ai}{\downarrow}$ and $\mathsf{ai}{\uparrow}$ nodes. We say that an ai-path is *maximal* if it cannot be extended, and that it is *open* if it begins and ends at a pending end of an edge.

The *inversion* of a path is just the same path in the reverse direction.

Example 3.3. The paths on the right, and their inversions, are exactly all the maximal ai-paths of the flow below. All of these are open except 0, since one of its ends is a $\mathsf{aw}{\downarrow}$ node. More generally all open paths are maximal but not vice-versa.

$$23679, 23678,$$
$$4578, 4579,$$
$$0, 1.$$
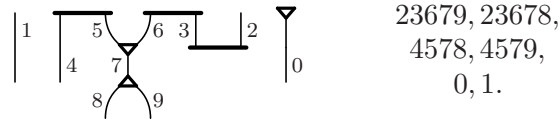
The following results allow us to estimate the size of the normal form of a flow, under norm, without actually constructing it.

Observation 3.4. $\underset{\mathsf{norm}}{\longrightarrow}$ preserves the number of open ai-paths in a flow.

Notation 3.5. We write $\ulcorner\phi\urcorner$ to denote the number of open ai-paths in a flow ϕ, modulo inversions, and $\#(\rho, \phi)$ to denote the number of ρ nodes in ϕ for an atomic structural rule ρ.

Lemma 3.6. *If ϕ is the flow of a KS proof then $\ulcorner\phi\urcorner = \#(\mathsf{ai}{\downarrow}, \phi)$.*

Proof. Since the flow of a proof can have no edge with upper end pending, every edge must be path-connected to a $\mathsf{aw}{\downarrow}$ or $\mathsf{ai}{\downarrow}$ node. Since no open path goes through an $\mathsf{aw}{\downarrow}$ node, and there are no $\mathsf{ai}{\uparrow}$ nodes, every open ai-path goes through a unique $\mathsf{ai}{\downarrow}$ node, and every $\mathsf{ai}{\downarrow}$ node accommodates at least one such path since there are no $\mathsf{aw}{\uparrow}$ nodes.

Now, the only other node a path can go through is an $\mathsf{ac}{\downarrow}$ node. Consequently every node in an open ai-path has in-degree 1 before passing its unique $\mathsf{ai}{\downarrow}$ node, and out-degree 1 after. Hence each $\mathsf{ai}{\downarrow}$ node accommodates exactly one open ai-path. □

Lemma 3.7. *If ϕ is the flow of a KS^+ proof, with normal form ψ under* norm*, then $|\psi| = O(|\phi| + \ulcorner\phi\urcorner)$.*

Proof. Let $\chi = \phi{\downarrow}_{\mathsf{wk}}$ so that $\phi \xrightarrow[\mathsf{wk}]{*} \chi \xrightarrow[\mathsf{cont}]{*} \psi$ by Lemma 2.19. By inspection of the rules of wk we must be able to decompose χ into two disjoint components: χ_1, consisting of just $\mathsf{ai}{\downarrow}$, $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ nodes, and χ_2 consisting of just $\mathsf{aw}{\downarrow}$ and $\mathsf{aw}{\uparrow}$ nodes. In fact, since ϕ was the flow of a proof, χ_2 consists of just $\mathsf{aw}{\downarrow}$ nodes.

Now notice that any $\xrightarrow[\mathsf{cont}]{}$ derivation from χ to ψ acts only on χ_1, and so ψ can be decomposed into two disjoint components: ψ_1, which is the normal form of χ_1 under norm, consisting of just $\mathsf{ai}{\downarrow}$ and $\mathsf{ac}{\downarrow}$ nodes by Prop. 3.1, and $\psi_2 = \chi_2$, consisting of just $\mathsf{aw}{\downarrow}$ nodes.

We have that $|\psi_2| = |\chi_2| \leq |\chi| \leq |\phi|$ by Prop. 2.18.

Notice that $|\psi_1| = 2 \cdot \#(\mathsf{ai}{\downarrow}, \psi_1) + \#(\mathsf{ac}{\downarrow}, \psi_1)$, since each $\mathsf{ai}{\downarrow}$ node has two edges and each $\mathsf{ac}{\downarrow}$ node adds a single extra edge. Since an $\mathsf{ac}{\downarrow}$ node has in-degree 2 and out-degree 1, the number of $\mathsf{ac}{\downarrow}$ nodes cannot outnumber the number of edges emanating from $\mathsf{ai}{\downarrow}$ nodes in ψ_1, i.e. $\#(\mathsf{ac}{\downarrow}, \psi_1) \leq 2 \cdot \#(\mathsf{ai}{\downarrow}, \psi_1)$, so we have $|\psi_1| \leq 4 \cdot \#(\mathsf{ai}{\downarrow}, \psi_1)$. By Lemma 3.6 we then have that $|\psi_1| \leq 4 \cdot \ulcorner\psi\urcorner$, and by Obs. 3.4 that $|\psi_1| \leq 4 \cdot \ulcorner\phi\urcorner$.

Putting these together we obtain $|\psi| = |\psi_1| + |\psi_2| \leq |\phi| + 4 \cdot \ulcorner\phi\urcorner$ as required. □

Theorem 3.8. *If Φ is a KS^+ proof with flow ϕ then we can transform it to a KS proof of the same conclusion in time polynomial in $|\Phi| + \ulcorner\phi\urcorner$.*

Proof. Let ψ be the normal form of ϕ under norm. By Prop. 3.1 we have that ψ contains just KS nodes, and so by Thm. 2.15 we have that a KS proof with same conclusion as Φ can be constructed in time polynomial in $|\Phi| + |\psi|$. The statement follows by the bound $|\psi| = O(|\phi| + \ulcorner\phi\urcorner) = O(|\Phi| + \ulcorner\phi\urcorner)$ given by Lemma 3.7. □

3.1. **Length of atomic flows.** It is not difficult to see that the main contributor to an increase of flow size reducing under norm is the rule $\mathsf{c}{\downarrow}$-$\mathsf{c}{\uparrow}$. It can sometimes cause an exponential blowup, as evident in Ex. 2.16 if there were no $\mathsf{aw}{\downarrow}$ node at the top.

The following result provides a simple estimate of the number of paths in a flow, and also the complexity of flow normalisation under norm, and is just an adaptation of well-known techniques to estimate the number of paths in any directed acyclic graph.

Definition 3.9 (Dimensions of a flow)**.** The *length* of a flow is the maximum number of times the type of node changes in an ai-path. The *width* of a flow is the maximum size of a connected subflow containing just one type of node. The *breadth* of a flow is the number of connected components it has.

The above definition is perhaps most easily understood by allowing $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ nodes to have unbounded in-degree and out-degree respectively. For example, by just collapsing

any configuration of $n-1$ connected $\mathsf{ac}{\downarrow}$ nodes to a single node $\bigvee\limits_{n}$, and similarly for configurations of $\mathsf{ac}{\uparrow}$ nodes. Then we can consider the length of the flow to be essentially just the maximum length, in the usual sense, of an ai-path, and the width is the maximum out-degree or in-degree.

It is worth mentioning here that replacing $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ nodes with these 'super' nodes can be lifted polynomially to proofs, in the sense that $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ steps in a proof can be soundly replaced by similar 'super' steps with only linear change in size.

Proposition 3.10. *If ϕ is a flow consisting of just KS^+ nodes with width w, length l and breadth b, then $\ulcorner\phi\urcorner = b \cdot w^{\frac{l}{2}+O(1)}$.*

Proof. For simplicity we write $\bigvee\limits_{n}$ for some configuration of $n-1$ connected $\mathsf{c}{\downarrow}$ nodes, and similarly for a $\mathsf{c}{\uparrow}$ configuration. Notice that it suffices to consider the case when $b = 1$, since paths in different connected components are disjoint.

In the worst case scenario we just have a sequence of $\substack{w \\ \displaystyle\bigvee \\ \displaystyle\bigwedge \\ w}$ configurations in series vertically, and each configuration multiplies the number of paths by w. It does not make much difference if there is a $\bigwedge\bigwedge$ configuration at the top of the flow because of the way we have defined an ai-path; the exponent differs by just the addition of a constant. $\square$

Remark 3.11. We generally use the trivial upper bound of size of flow for width and breadth, yielding the estimate $\ulcorner\phi\urcorner = |\phi|^{\frac{l}{2}+O(1)}$ for a KS^+ flow ϕ.

3.2. **Contraction loops in atomic flows.** Sometimes the estimate of number of open ai-paths given via length of a flow in the previous section is not sufficiently accurate. An example of this is given in Sect. 6.2 where the length of flows in the translation R is polynomial in their size, and so the estimate is exponential, yet the actual number of open ai-paths is bounded above by a polynomial.

Unsurprisingly, it is only certain interactions between $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ nodes that generate significant complexity in a KS^+ flow; in this section we focus on one type of interaction we call a 'contraction loop'. In the absence of these the number of open ai-paths in a flow is still polynomial in its size, regardless of its length.

Definition 3.12. A *contraction loop* in a flow is a pair of $(\mathsf{ac}{\uparrow}, \mathsf{ac}{\downarrow})$ nodes (ν_1, ν_2) such that there are at least two disjoint (directed) paths between ν_1 and ν_2

For example we give the following flow and all its contraction loops,

$$(u, y), (v, z).$$
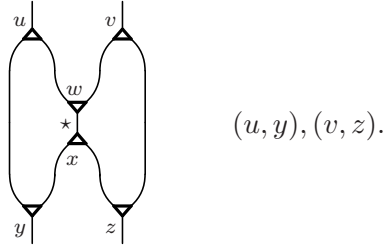
whereas every other pair has only one path between them. If the edge $\star$ were broken and there was no path from w to x then there would be no contraction loops at all.

Lemma 3.13. *If there are no contraction loops in a* KS^+ *flow* ϕ *then* $\ulcorner\phi\urcorner \leq |\phi|^3$.

Proof. For an edge ϵ in ϕ consider the following two notions:

- The *weight* of ϵ, denoted $w(\epsilon)$, is the number of directed paths from ϵ to the bottom of ϕ, i.e. to a $\mathsf{aw}{\uparrow}$ node or an edge with lower end pending.
- For an atomic structural rule ρ let $N(\rho, \epsilon)$ denote the number of ρ nodes below ϵ that are connected to ϵ by a directed path.

We show that $w(\epsilon) \leq N(\mathsf{aw}{\uparrow}, \epsilon) + N(\mathsf{ac}{\uparrow}, \epsilon) + 1$ by induction on the distance of ϵ from the bottom of ϕ. The inequality is clear for the base cases when ϵ is directly connected to a $\mathsf{aw}{\uparrow}$ node, $\epsilon\!\!\begin{smallmatrix}\\ \blacktriangle\end{smallmatrix}$, and when ϵ has lower end pending, $\epsilon|$. We have two inductive steps:

(1) ϵ is an upper edge of a $\mathsf{ac}{\downarrow}$ node, $\begin{smallmatrix}\epsilon\\ \blacktriangledown\\ \delta\end{smallmatrix}$. In this case we clearly have that $w(\epsilon) = w(\delta)$

and so the inequality follows by the inductive hypothesis.

(2) ϵ is the upper edge of a $\mathsf{ac}{\uparrow}$ node, $\begin{smallmatrix}\epsilon\\ \gamma\blacktriangle\delta\end{smallmatrix}$. Observe that, since there are no contraction

loops in ϕ and $\mathsf{ac}{\downarrow}$ is the only node type with in-degree greater than 1, any node below this $\mathsf{ac}{\uparrow}$ node can be directed-path-connected to at most one of γ or δ. Consequently we have that $N(\mathsf{aw}{\uparrow}, \epsilon) = N(\mathsf{aw}{\uparrow}, \gamma) + N(\mathsf{aw}{\uparrow}, \delta)$ and $N(\mathsf{ac}{\uparrow}, \epsilon) = N(\mathsf{ac}{\uparrow}, \delta) + N(\mathsf{ac}{\uparrow}, \gamma) + 1$. Therefore,

$$\begin{aligned}
w(\epsilon) \ &= w(\delta) + w(\gamma) \\
&\leq (N(\mathsf{aw}{\uparrow}, \delta) + N(\mathsf{ac}{\uparrow}, \delta) + 1) + (N(\mathsf{aw}{\uparrow}, \gamma) + N(\mathsf{ac}{\uparrow}, \gamma) + 1) \\
&\leq (N(\mathsf{aw}{\uparrow}, \delta) + N(\mathsf{aw}{\uparrow}, \gamma) + (N(\mathsf{ac}{\uparrow}, \delta) + N(\mathsf{ac}{\uparrow}, \gamma) + 1) + 1 \\
&\leq N(\mathsf{aw}{\uparrow}, \epsilon) + N(\mathsf{ac}{\uparrow}, \epsilon) + 1
\end{aligned}$$

Finally, notice that $|\phi| \geq N(\mathsf{aw}{\uparrow}, \phi) + N(\mathsf{ac}{\uparrow}, \phi) + 1$ and so we have shown that $w(\epsilon) \leq |\phi|$.

Clearly the number of open ai-paths going through an edge with upper end pending is bounded above by its weight, and so by $|\phi|$ by the bound above, while the number of open ai-paths going through any $\mathsf{ai}{\downarrow}$ node is bounded above by the product of the weights of each of its edges, and so by $|\phi|^2$. In particular we have that the number of open ai-paths going through any edge at the top of a flow is bounded above by $|\phi|^2$, and there are at most $|\phi|$ many such edges, whence the bound follows.

$\square$

Example 3.14. In fact the bound given above is optimal, up to multiplication by a constant. Consider the flow ϕ below, where there are n ai↓ nodes:
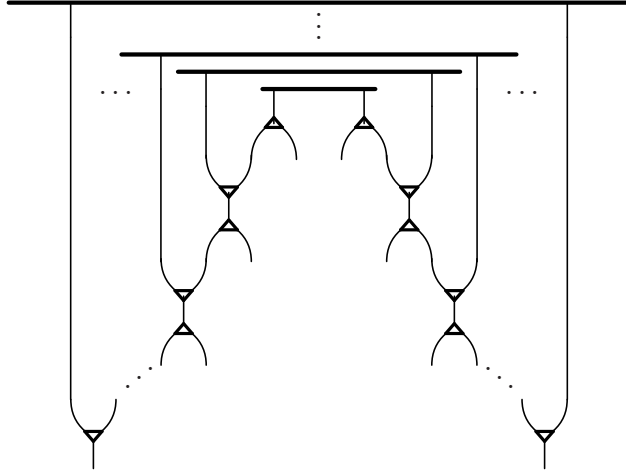
Clearly the flow has size linear in n and notice that the weights (as defined in the above proof) of the topmost edges, starting from the left, are $1, 2, \ldots, n, n, \ldots, 1$ respectively. Consequently the number of open ai-paths going through the ai↓ nodes, starting from the outside, is $1^2, 2^2, \ldots, n^2$ respectively, and so taking the sum we obtain $\ulcorner\phi\urcorner = \Omega(n^3)$.

Notice also that ϕ has length linear in n and width 2, and so the upper bound on $\ulcorner\phi\urcorner$ given by Prop. 3.10 is exponential, considerably worse than the bound given in Lemma 3.13.

4. Truth tables and tree-like Gentzen systems

KS polynomially simulates tree-like cut-free Gentzen systems since its rules are generalisations of Gentzen rules [BT01] [BG09]. In the other direction Bruscoli and Guglielmi have proved in [BG09] that the converse does not hold, by way of the so-called 'Statman tautologies'. We offer a new proof here, via truth tables, as an exercise prior to the main results.

Let LK^- denote the cut-free sequent calculus, and tree-LK^- denote the system of tree-like proofs in this calculus.[3]

Proposition 4.1 (D'Agostino). *Tree-LK^- cannot polynomially simulate truth tables.*

Proof. See [D'A92]. □

To expand slightly on the above proposition, truth tables are efficient when there are exponentially many occurrences of each atom, and some such tautologies are hard for tree-LK^-. One such example, used by D'Agostino, is simply the disjunction of every assignment on k propositional variables, what we call $\bigvee_{\mathcal{A}} \gamma_{\mathcal{A}}$ below. Such a formula has size $k \cdot 2^k$, although any tree-like cut-free sequent proof must contain at least $k!$ branches, while a truth table contains 2^k rows and $k \cdot 2^k$ columns.

Lemma 4.2. KS$^+$ *polynomially simulates truth tables.*

---

[3]It does not matter too much which version of the sequent calculus we choose as complexity properties are typically preserved across common variations.

Proof. Let τ be a tautology. For each partial assignment $\mathcal{A}$, defined on just those variables appearing in τ, and each formula A satisfied by $\mathcal{A}$ construct a derivation $\Phi_{\mathcal{A}}(A)$ by structural induction on A as follows:

$$\Phi_{\mathcal{A}}(a) \equiv a \quad , \quad \Phi_{\mathcal{A}}(B \wedge C) \equiv \Phi_{\mathcal{A}}(B) \wedge \Phi_{\mathcal{A}}(C) \quad , \quad \Phi_{\mathcal{A}}(B \vee C) \equiv \dfrac{= \dfrac{\Phi_{\mathcal{A}}(B)}{B \vee \mathsf{w}\downarrow \dfrac{\perp}{C}}}{}$$

where, in the last case, when A is a disjunction, the disjunct B was chosen such that B is satisfied by $\mathcal{A}$. It is clear that each $\Phi_{\mathcal{A}}(\tau)$ has conclusion τ and premiss a conjunction of literals; moreover this conjunction of literals is satisfied by $\mathcal{A}$.

Let $\gamma_{\mathcal{A}}$ be the conjunction of all literals satisfied by $\mathcal{A}$, so that each variable appears exactly once. Then we can easily construct derivations $\begin{array}{c}\gamma_{\mathcal{A}} \\ \Big\| \{\mathsf{aw}\uparrow,\mathsf{ac}\uparrow\} \\ \mathsf{pr}(\Phi_{\mathcal{A}}(\tau))\end{array}$.

Now construct a proof Ψ of $\bigvee_{\mathcal{A}} \gamma_{\mathcal{A}}$ in $\{\mathsf{ai}\downarrow, \mathsf{ac}\uparrow, \mathsf{s}, \mathsf{m}\}$ by induction on the number of distinct variables, as shown below:

$$\textit{Base case:} \quad \mathsf{ai}\downarrow \dfrac{\top}{a \vee \bar{a}} \quad , \quad \textit{Inductive step:} \quad \bigvee_{\mathcal{A}}\left(2\cdot\mathsf{s} \dfrac{\Psi \Big\| \{\mathsf{ai}\downarrow,\mathsf{ac}\uparrow,\mathsf{s},\mathsf{m}\} = \dfrac{\gamma_{\mathcal{A}}}{\mathsf{ai}\downarrow \dfrac{\top}{a \vee \bar{a}} \wedge \mathsf{c}\uparrow \dfrac{\gamma_{\mathcal{A}}}{\gamma_{\mathcal{A}} \wedge \gamma_{\mathcal{A}}}}}{\bigvee_{\mathcal{A}}(a \wedge \gamma_{\mathcal{A}}) \vee \bigvee_{\mathcal{A}}(\bar{a} \wedge \gamma_{\mathcal{A}})}\right)$$

Finally we put these together and apply contractions to obtain a KS^+ proof of τ:

$$\bigvee_{\mathcal{A}}\left(\begin{array}{c} \Psi \Big\| \{\mathsf{ai}\downarrow,\mathsf{ac}\uparrow,\mathsf{s},\mathsf{m}\} \\ \gamma_{\mathcal{A}} \\ \Big\| \{\mathsf{aw}\uparrow,\mathsf{ac}\uparrow\} \\ \mathsf{pr}(\Phi_{\mathcal{A}}(\tau)) \\ \Phi_{\mathcal{A}} \Big\| \{\mathsf{w}\downarrow\} \\ \tau \end{array} \right)$$
$$\Big\| \{\mathsf{c}\downarrow\}$$
$$\tau$$

It is clear that the derivations inside the large parentheses have size polynomial in $|\tau|$, which is the number of columns in a truth table, and the number of these derivations appearing in disjunction is just the number of assignments, which is the number of rows in a truth table. $\square$

Theorem 4.3. KS *polynomially simulates truth tables.*

Proof. Notice that, in the above simulation, all $\mathsf{ac}\uparrow$ steps are above all $\mathsf{ac}\downarrow$ steps, and so the associated flow will have bounded length. The result follows by Cor. 2.14, Lemma 3.7 and Prop. 3.10. $\square$

Corollary 4.4. *Tree-LK^- cannot polynomially simulate* KS.

Proof. Immediate from Prop. 4.1 and Thm. 4.3. $\square$

Remark 4.5. It should be noted that D'Agostino's separation is only quasipolynomial,[4] and so our separation is also only quasipolynomial, while the proof using the Statman tautologies in [BG09] yields an exponential separation. Nonetheless, an exponential separation follows from the results in the next section.

5. Separations via variants of the pigeonhole principle

Jeřábek has shown that KS^+ has polynomial-size proofs of the functional and onto variants of the pigeonhole principle [Jeř09]. We show that these proofs reduce under norm to KS proofs with only polynomial increase in size, and so cut-free sequent calculi, Resolution and bounded-depth Frege systems are unable to polynomially simulate KS.

The pigeonhole principle states that, if n pigeons sit in $n-1$ holes, some hole must contain more than one pigeon. In its unrestricted formulation, the mapping from pigeons to holes can be many-many, while in the functional variant it must be many-one, i.e. a function, and the onto variant insists that each hole must be occupied. It is clear that the two variants are weaker than the unrestricted version, and the variant containing both criteria, the onto functional pigeonhole principle, is weaker still. We will see this more clearly in the following definition.

The propositional encodings of pigeonhole principle variants below are most easily understood by interpreting the atoms a_{ij} as "pigeon i sits in hole j". Recall that it does not matter how large disjunctions and conjunctions are bracketed, since any valid bracketing can be easily reduced to any other by the $=$ rule.

Definition 5.1 (Pigeonhole principles)**.** We define the following formulae,

$$\mathsf{PHP}_n \equiv \bigvee_{i=0}^{n} \bigwedge_{j=1}^{n} \bar{a}_{ij} \vee \bigvee_{i=0}^{n} \bigvee_{j<j'} (a_{ij} \wedge a_{ij'}), \quad \mathsf{F}_n \equiv \bigvee_{j=1}^{n} \bigvee_{i<i'} (a_{ij} \wedge a_{i'j}), \quad \mathsf{O}_n \equiv \bigvee_{j=1}^{n} \bigwedge_{i=0}^{n} \bar{a}_{ij}$$

and denote by FPHP_n, OPHP_n and OFPHP_n the formulae obtained by putting in disjunction the associated formulae, i.e. $\mathsf{FPHP}_n \equiv \mathsf{F}_n \vee \mathsf{PHP}_n$, $\mathsf{OPHP}_n \equiv \mathsf{O}_n \vee \mathsf{PHP}_n$ and $\mathsf{OFPHP}_n \equiv \mathsf{O}_n \vee \mathsf{F}_n \vee \mathsf{PHP}_n$.

We can see in the above definition that any variant can be obtained from a stronger variant, i.e. one with a subset of disjuncts, by a simple application of generic weakening $\mathsf{w}{\downarrow}$. Consequently upper bounds on the size of proofs of one variant yield upper bounds for all weaker variants, and lower bounds vice-versa.

The following result was proved by Beame, Impagliazzo and Pitassi, and independently by Krajíček, Pudlák and Woods.

Theorem 5.2. *Bounded-depth Frege systems have only exponential-size proofs of* OFPHP_n*.*

Proof. See [KPW95], [PBI93]. $\qquad\qquad\square$

Corollary 5.3. *Bounded-depth Frege systems, Resolution and cut-free sequent calculi have only exponential-size proofs of all variants of the pigeonhole principle.*

Proof. All the systems are just special cases of bounded-depth Frege, and a proof of any variant can be extended to one of OFPHP_n by an application of generic weakening $\mathsf{w}{\downarrow}$. $\quad\square$

---

[4]A quasipolynomial is a function of size $n^{\log^{\Theta(1)} n}$. By taking logarithms and using the integral bound it is not difficult to see that $k!$ grows quasipolynomially in 2^k.

On the other hand we have the following:

Theorem 5.4 (Buss)**.** *There are polynomial-size Frege proofs of* PHP_n*, and so all variants of the pigeonhole principle.*

Proof. See [Bus87]. □

From here, polynomial-size SKS proofs of PHP_n follow from the following observation:

Proposition 5.5 (Bruscoli and Guglielmi)**.** SKS *is polynomially equivalent to Frege systems.*

Proof. See [BG09]. □

Notice that one direction of the above proposition, indeed the direction that we require, that SKS polynomially simulates Frege systems, can be obtained by recognising that the rules of SKS are just generalisations of the rules of Gentzen systems with cut, which are well-known to be polynomially equivalent to Frege systems.

The following trick, now standard in the deep inference literature, is very useful for proving certain tautologies in KS, as we will see. The idea is that if we know there is an SKS proof of a tautology, then we can transform it into a KS proof of that tautology in disjunction with trivial contradictions. If we can find derivations from each of these contradictions to the tautology we want to prove, then the composition mechanisms for derivations in deep inference guarantee that we can build a proof of the tautology.

Lemma 5.6. *Let A be a formula over the atoms $a_1, \ldots, a_n$. Every SKS proof Φ of A can be polynomially transformed to a KS proof of $A \vee \bigvee_i (a_i \wedge \bar{a}_i)$.*

Proof. See e.g. [Jeř09], [BGGP10]. □

Lemma 5.7 (Jeřábek)**.** *There are polynomial-size proofs of FPHP_n and OPHP_n in KS^+.*

Proof. By Thm. 5.4, Prop. 5.5 and Lemma 5.6 there are polynomial-size KS proofs with conclusion $\mathsf{PHP}_n \vee \bigvee_{i,j}(a_{ij} \wedge \bar{a}_{ij})$. For each atom a_{st} we construct a derivations $\Phi_n^{a_{st}}$ in $\mathsf{KS}^+ \setminus \{\mathsf{ac}{\downarrow}\}$ with premiss $a_{st} \wedge \bar{a}_{st}$ and conclusion FPHP_n as follows:

$$
\mathsf{w}{\downarrow}\,\frac{\bigwedge_j \bar{a}_{sj} \vee \left(\mathsf{(n-2)\cdot ac}{\uparrow}\,\frac{\dfrac{a_{st}}{a_{st} \wedge \cdots a_{st}} \wedge \bigvee_{j\neq t} a_{sj}}{\|\{\mathsf{s}\}\quad\quad} \atop \bigvee_{j\neq t} a_{st} \wedge a_{sj} \right)}{\mathsf{FPHP}_n}
$$

We then put these together and apply contractions to obtain proofs of FPHP_n:

$$
\begin{array}{c}
\left\|\text{KS}\right. \\[2pt]
\left[\mathsf{PHP}_n \vee \bigvee_{i,j} \left(\begin{array}{c} a \wedge \bar{a} \\ \Phi_n^{a_{ij}} \left\| \text{KS}^+\backslash\{\text{ac}\downarrow\}\right. \\ \mathsf{FPHP}_n \end{array}\right)\right] \\[2pt]
\left\|\{\text{c}\downarrow\}\right. \\[2pt]
\mathsf{FPHP}_n
\end{array}
$$

We can construct similar derivations from $a_{st} \wedge \bar{a}_{st}$ to OPHP_n, given below, and put them together in a similar way to obtain proofs of OPHP_n in KS^+.

$$
\mathsf{w}\downarrow \frac{\bigwedge_i \bar{a}_{it} \vee \left(\begin{array}{c} (n-2)\cdot\mathsf{ac}\uparrow \dfrac{a_{st}}{a_{st} \wedge \cdots \wedge a_{st}} \wedge \bigvee_{i\neq s} a_{it} \\ \left\|\{\mathsf{s}\}\right. \\ \bigvee_{i\neq s} a_{st} \wedge a_{it} \end{array}\right)}{\mathsf{OPHP}_n}
$$

where the upper part is

$$
2\cdot\mathsf{s} \frac{a_{st} \wedge \bar{a}_{st} \wedge \mathsf{i}\downarrow \dfrac{= \dfrac{a_{st} \wedge \bar{a}_{st}}{\top}}{\bigwedge_{i\neq s} \bar{a}_{it} \vee \bigvee_{i\neq s} a_{it}}}{}
$$

$\square$

Theorem 5.8. *There are polynomial-size KS proofs of FPHP_n, OPHP_n and so also OFPHP_n.*

Proof. In the proofs of FPHP_n constructed above, Lemma 5.7, notice that the only $\mathsf{ac}\uparrow$ steps occur in $\Phi_n^{a_{st}}$ where there are also no $\mathsf{ac}\downarrow$ steps, and similarly for OPHP_n. It follows that there are only two alternations between $\mathsf{ac}\downarrow$ and $\mathsf{ac}\uparrow$ steps in the path of any atom from an $\mathsf{ai}\downarrow$ step, and so the atomic flows of these proofs will have bounded length. The result follows by Thm. 3.8 and Prop. 3.10. $\square$

Corollary 5.9. *Cut-free sequent calculi, Resolution and bounded-depth Frege systems are exponentially separated from KS.*

Proof. Immediate from Cor. 5.3 and Thm. 5.8. $\square$

6. Polynomial simulations of versions of Resolution

In this section we present a polynomial simulation in KS of tree-like and multiset Resolution systems. We point out that, in previous work [Das12b], a more general result was claimed, namely a simulation of unrestricted Resolution operating with sets, however that proof contained errors that we could not amend for this version of the article. The status of those results is unresolved.

We define the Resolution system below in both its set and multiset formulations.

Definition 6.1 (Resolution)**.** We use symbols Γ, Δ etc. to range over (multi)sets of literals and write '$\cup$' to denote (multi)set union. We define the system *(mulitset-)Resolution* by the following rules:

$$\mathsf{wk}\ \frac{\Gamma}{\Gamma \cup \Delta} \quad , \quad \mathsf{res}\ \frac{\Gamma \cup \{a\} \quad \Delta \cup \{\bar{a}\}}{\Gamma \cup \Delta}$$

A *derivation* from (multi)sets $\Gamma_1, \ldots, \Gamma_s$ is a list[5] $\pi = (\Delta_1, \ldots, \Delta_n)$ where each Δ_i is some Γ_j or follows by one of the rules above whose premisses have occurred previously in the list. If we further have that $\Delta_n = \emptyset$ then we call π a *refutation* of $\Gamma_1, \ldots, \Gamma_s$.

We call a derivation $(\Delta_1, \ldots, \Delta_n)$ *tree-like* if each Δ_i is used at most once as the premiss of any inference step concluding some Δ_j.

To simplify the treatment of (multiset-)Resolution derivations we address certain rather pathological situations below. These are also the reason why we opt to include weakening in our formulation.[6]

Remark 6.2 (Assumptions on derivation format)**.** In a (multiset-)Resolution derivation, we can assume that neither premiss of a res step contains both a resolved atom and its dual. Any such step would have the following format,

$$\mathsf{res}\ \frac{\Gamma \cup \{a\} \quad \Delta \cup \{a, \bar{a}\}}{\Gamma \cup \Delta \cup \{a\}}$$

which can be simulated by an application of wk to the left premiss, if necessary at all.

We consequently have that no res step has identical premisses, and so there is no need to consider this case in the translations that follow.

Finally, in the case of sets, we further assume that neither the resolved atom nor its dual appear in the conclusion of a res step. I.e., in the definition of the res above, we assume that Γ and Δ contain neither a nor $\bar{a}$. Aside from the above situation this phenomenon might also occur if, say, $a \in \Gamma$ and so $\Gamma = \Gamma \cup \{a\}$, whence the step may be simulated by an application of wk in a similar way.

Before presenting our simulations in KS of versions of Resolution, we set some notational conventions below in order to easily switch between the two settings.

Notation 6.3 (Set symbols in deep inference)**.** To reduce the amount of syntax in our deep inference derivations, we will simply write Γ instead of $\bigvee \Gamma$ for the disjunction of the members of Γ, as an abuse of notation.

We similarly use other (multi)set-theoretic notation. In particular, in light of the above remark, we will always have that $\Gamma \cup \{a\} = \Gamma \vee a$ in both the set and multiset settings whenever it occurs. In the multiset setting we further have that $\Gamma \cup \Delta = \Gamma \vee \Delta$ for all Γ, Δ.

Remark 6.4. Throughout this section, when we say that a proof system polynomially simulates a refutation system, we mean that every refutation of A in the latter can be polynomially transformed to a proof of $\bar{A}$ in the former.

---

[5]We do not mind how the list is delimited. In the translations below they are written vertically for presentation reasons.

[6]It is straightforward to show that the formulation without weakening is polynomially equivalent by a rule permutation argument.

Definition 6.5 (Dual systems)**.** The dual of a deep inference rule $\rho\,\dfrac{A}{B}$ is $\bar\rho\,\dfrac{\bar B}{\bar A}$. E.g. in SKS a rule $x\downarrow$ is dual to $x\uparrow$, while s and m are self-dual. The set of duals of a system $\mathcal{S}$ is denoted $\overline{\mathcal{S}}$. Notice that a rule is sound if and only if its dual is, by the law of contraposition.

The dual of a derivation $\Phi\|\mathcal{S}$ starting from A and ending B is the derivation $\bar\Phi\|\bar{\mathcal{S}}$ starting from $\bar B$ and ending $\bar A$ obtained by flipping Φ upside down and replacing each inference step with its dual.

Similarly we define the dual of a flow-rewriting rule to be the rule flipped upside-down, replacing each node with its dual, and the dual of a rewriting system is just the set of its duals. A rule/system is sound, terminating and/or confluent if and only if its dual is.

We are now ready to define our basic translation from Resolution derivations to deep inference, on which our simulation results in later sections will be based.

Definition 6.6 (Translation of Resolution derivations)**.** We give the following translation R of a Resolution step to a $\overline{\mathsf{KS}^+}$ derivation,

$$
R:\qquad \mathsf{wk}\,\frac{\Gamma}{\Gamma\cup\Delta}\qquad\mapsto\qquad
\begin{array}{c}
=\dfrac{\Gamma}{\Gamma\vee\mathsf{w}\downarrow\dfrac{\bot}{\Delta\setminus\Gamma}}\\[6pt]
=\dfrac{}{\Gamma\cup\Delta}
\end{array}
$$

$$
R:\qquad \mathsf{res}\,\frac{\Gamma\cup\{a\}\quad \Delta\cup\{\bar a\}}{\Gamma\cup\Delta}\qquad\mapsto\qquad
\begin{array}{c}
2\cdot\mathsf{s}\,\dfrac{=\dfrac{\Gamma\cup\{a\}}{\Gamma\vee a}\ \wedge\ =\dfrac{\Delta\cup\{\bar a\}}{\Delta\vee\bar a}}{\Gamma\vee\Delta\vee\mathsf{ai}\uparrow\dfrac{a\wedge\bar a}{\bot}}\\[8pt]
=\dfrac{}{(\mathsf{c}\downarrow\dfrac{\Gamma\vee\Delta}{\Gamma\cup\Delta})}
\end{array}
$$

where the parenthesised $\mathsf{c}\downarrow$ steps are present only when Γ and Δ are sets (not multisets) that have nonempty intersection.

We extend the definition of R to any (multiset-)Resolution derivation $\pi=(\Delta_1,\dots,\Delta_n)$ from $\Gamma_1,\dots,\Gamma_s$, such that R has the following format:

$$
R\quad:\quad
\begin{array}{c}\Delta_1\\ \vdots\\ \Delta_n\end{array}
\quad\mapsto\quad
\begin{array}{c}\bigwedge\limits_{r=1}^{s}\Gamma_r\\ \Big\|\overline{\mathsf{KS}^+}\\ \bigwedge\limits_{m=1}^{n}\Delta_m\end{array}
$$

The definition is by induction on the length n of the Resolution derivation π.

If $n=0$, i.e. π is an empty list, then $\bigwedge_m\Delta_m=\top$ and we define:[7]

$$
R:\quad \pi\quad\mapsto\quad \mathsf{w}\uparrow\,\frac{\bigwedge\limits_{r=1}^{s}\Gamma_r}{\top}
$$

---

[7]Throughout this translation we shall omit certain $=$ rules required to handle units, e.g. we associate the empty disjunction with $\bot$ and the empty conjunction with $\top$. By now this is a routine consideration.

If π is extended by a (multi)set Γ_i then we define,

$$
R: \quad
\begin{array}{c}
\Delta_1 \\
\vdots \\
\Delta_n \\
\Gamma_i
\end{array}
\quad \mapsto \quad
=
\cfrac{\bigwedge\limits_{r=1}^{s} \Gamma_r}{
=
\cfrac{\bigwedge\limits_{r \neq i} \Gamma_r \wedge \mathsf{c}{\uparrow} \cfrac{\Gamma_i}{\Gamma_i \wedge \Gamma_i}}{
\begin{array}{c}
ID \left\Vert \; \cfrac{\bigwedge\limits_{r=1}^{s} \Gamma_r}{\bigwedge\limits_{m=1}^{n} \Delta_m} \quad \wedge \Gamma_i
\end{array}}}
$$

where the derivation marked ID is already defined by induction.

If π is extended by a wk step then we define,

$$
R: \quad
\begin{array}{c}
\Delta_1 \\
\vdots \\
\Delta_i \\
\vdots \\
\Delta_n \\
\Delta_i \cup \Sigma
\end{array}
\quad \mapsto \quad
=
\cfrac{ID \left\Vert \; \cfrac{\bigwedge\limits_{r=1}^{s} \Gamma_r}{\bigwedge\limits_{m=1}^{n} \Delta_m}}{
=
\cfrac{\bigwedge\limits_{m \neq i} \Delta_m \wedge \; \mathsf{c}{\uparrow} \cfrac{\Delta_i}{\Delta_i} \; \wedge R(\mathsf{wk}) \left\Vert \; \cfrac{\Delta_i}{\Delta_i \cup \Sigma}\right.}{\bigwedge\limits_{m=1}^{n+1} \Delta_m}}
$$

where $\Delta_{n+1} = \Delta_i \cup \Sigma$ and the derivation marked ID is already defined by induction.

If π is extended by a res step then we define,

$$
R: \quad
\begin{array}{c}
\Delta_1 \\
\vdots \\
\Delta_i' \cup \{a\} \\
\vdots \\
\Delta_j' \cup \{\bar{a}\} \\
\vdots \\
\Delta_n \\
\Delta_i' \cup \Delta_j'
\end{array}
\quad \mapsto \quad
=
\cfrac{ID \left\Vert \; \cfrac{\bigwedge\limits_{r=1}^{s} \Gamma_r}{\bigwedge\limits_{m=1}^{n} \Delta_m}}{
=
\cfrac{\bigwedge\limits_{m \neq i,j} \Delta_k \wedge \; \cfrac{\mathsf{c}{\uparrow}\cfrac{\Delta_i}{\Delta_i \wedge \Delta_i} \wedge \mathsf{c}{\uparrow}\cfrac{\Delta_j}{\Delta_j \wedge \Delta_j}}{\Delta_i \wedge \Delta_j} \wedge R(\mathsf{res}) \left\Vert \; \cfrac{\Delta_i \wedge \Delta_j}{\Delta_i' \cup \Delta_j'}\right.}{\bigwedge\limits_{m=1}^{n+1} \Delta_m}}
$$

where $\Delta_i = \Delta_i' \cup \{a\}$, $\Delta_j = \Delta_j' \cup \{\bar{a}\}$, $\Delta_{n+1} = \Delta_i' \cup \Delta_j'$ and the derivation marked ID is already defined by induction.

6.1. **Multiset Resolution.** We consider the multiset setting, for arbitrary derivations (i.e. not necessarily tree-like). Here we achieve a polynomial simulation in KS rather easily by using the identities pointed out in Not. 6.3.

Observation 6.7. The image of a multiset-Resolution derivation under R has no $\mathsf{ac}{\downarrow}$ steps.

Proof. The only $\mathsf{ac}{\downarrow}$ steps in the definition of R occur in the translation of individual res steps, where the format of the premiss is $\Gamma \vee \Delta$ and the conclusion $\Gamma \cup \Delta$. However, as already pointed out in Not. 6.3, these are equivalent in the multiset setting, and so no $\mathsf{ac}{\downarrow}$ steps are necessary at all. $\square$

Theorem 6.8. KS *polynomially simulates multiset resolution.*

Proof. For a multiset resolution refutation π of $\Gamma_1, \ldots, \Gamma_s$ we have that $R\pi$ is a $\overline{\mathsf{KS}^+}$ derivation from $\bigwedge\limits_{r=1}^{s} \Gamma_r$ to $\bot$, by definition, not containing $\mathsf{ac}{\downarrow}$ steps, by the observation above. Consequently we have that $\overline{R\pi}$ has the following format,

$$
\begin{array}{c}
\top \\
\overline{R\pi} \left\| \mathsf{KS} \cup \{\mathsf{aw}{\uparrow}\} \right. \\
\bigvee\limits_{r=1}^{s} \bar{\Gamma}_r
\end{array}
$$

whose flow has bounded length (due to the absence of $\mathsf{ac}{\uparrow}$ nodes) and so can be transformed to a KS proof of the same conclusion in polynomial time by Prop. 3.10 and Thm. 3.8. $\square$

6.2. **Tree-like Resolution.** We now consider Resolution derivations over sets and show a polynomial simulation of tree-like refutations in KS. Here the argument is only slightly more involved: there are both $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ steps occurring, but they do not interact in any complex way.

We point out that the proof of this result could perhaps be carried out more simply by writing tree-like derivations as trees and simulating steps locally, discarding premisses once they are used. However the current presentation allows us to use the same translation R for this and the multiset setting, as well as the extensions introduced in the next section. This uniform treatment is possible due to Lemma 3.13 on the number of open ai-paths in the absence of contraction loops.

Proposition 6.9. *If π is a tree-like Resolution derivation then every $\mathsf{ac}{\uparrow}$ node in $fl(R\pi)$ has one edge pending.*

Proof. By induction on the length of π. On inspection of the atomic flows, notice that each translation step introduces only $\mathsf{c}{\uparrow}$ nodes whose lower left edge is pending (with respect to the horizontal order of atoms given in the definition of R in Dfn. 6.6), so it suffices to show that new nodes are not attached to the lower edge of existing $\mathsf{c}{\uparrow}$ nodes.

If any such situation existed then, by construction, each $\mathsf{c}{\uparrow}$ node must be associated with the same set Δ_i that is the premiss of distinct steps in π, contradicting the fact that π is tree-like. $\square$

Recall the notion of contraction loop from Dfn. 3.12.

Corollary 6.10. *If π is a tree-like Resolution derivation then $fl(R\pi)$ has no contraction loops.*

Proof. Any contraction loop would violate Prop. 6.9 above. $\square$

Theorem 6.11. KS *polynomially simulates tree-like Resolution.*

Proof. For a tree-like Resolution refutation π of $\Gamma_1, \ldots, \Gamma_s$, as in the proof of Thm. 6.8, we have that $\overline{R\pi}$ has the format:

$$\overline{R\pi} \Big\|\mathsf{KS}^+ \quad \begin{matrix} \top \\ \\ \bigvee_{r=1}^{s} \bar{\Gamma}_r \end{matrix}$$

We also have that $fl(\overline{R\pi})$ contains no contraction loops, since otherwise there would also be a contraction loop in $fl(R\pi)$, violating Cor. 6.10. The result now follows from Lemma 3.13 and Thm. 3.8. $\square$

6.3. **Extensions of Resolution.** Finally, we notice that this simulation extends to some extensions of Resolution, operating on (multi)sets of conjunctions of literals, introduced by Krajíček in [Kra01]. These systems are known to be strictly stronger than their counterparts we have so far dealt with [SBI02] [EGM04].

Definition 6.12. Let s, t, etc. denote terms, i.e. conjunctions of literals, and Γ, Δ, etc. now denote sets of terms. We define the following rules,

$$\mathsf{wk} \frac{\Gamma}{\Gamma \cup \Delta} \quad , \quad \wedge \frac{\Gamma \cup \{s\} \quad \Delta \cup \{t\}}{\Gamma \cup \Delta \cup \{s \wedge t\}} \quad , \quad \mathsf{res} \frac{\Gamma \cup \{s\} \quad \Delta \cup \{\bar{a}_1, \ldots, \bar{a}_k\}}{\Gamma \cup \Delta}$$

where in the res rule we require that $s = \bigwedge_{i=1}^{k} a_i$.

For a function $f : \mathbb{N} \to \mathbb{N}$, a (tree-like) (multiset-)Resolution(f) derivation/refutation is defined analogously to Dfn. 6.1, using the rules above, with the additional proviso that no term has size bigger than $f(N)$, where N is the number of inference steps in the refutation.

Theorem 6.13. *For any $f : \mathbb{N} \to \mathbb{N}$ we have the following:*
(1) KS *polynomially simulates multiset-Resolution(f).*
(2) KS *polynomially simulates tree-like Resolution(f).*

Proof sketch. The proofs are analogous to those appearing earlier for the tree-like and multiset variants of Resolution respectively, interpreting Γ, Δ etc. as (multi)sets of terms and replacing variables $a, \bar{a}$ etc. by term variables and their duals $t, \bar{t}$ etc.[8]

---

[8]In light of the aforementioned abuse of notation identifying a (multi)set with the disjunction of its elements, we associate the dual of term $\bigwedge_{i=1}^{k} a_i$ with the (multi)set $\{\bar{a}_1, \ldots, \bar{a}_k\}$.

We extend the definition of R on individual steps to $\wedge$ steps as follows:

$$R: \quad \wedge \frac{\Gamma \cup \{s\} \quad \Delta \cup \{t\}}{\Gamma \cup \Delta \cup \{s \wedge t\}} \quad \mapsto \quad {}_{2 \cdot \mathsf{s}} \frac{= \frac{\Gamma \cup \{s\}}{\Gamma \vee s} \wedge = \frac{\Delta \cup \{t\}}{\Delta \vee t}}{(\mathsf{c}\downarrow \frac{\overline{\Gamma \vee \Delta \vee (s \wedge t)}}{\Gamma \cup \Delta \cup \{s \wedge t\}})}$$

R is extended to Resolution(f)-derivations similarly to Dfn. 6.6, dealing with $\wedge$ steps in the same way res steps.

For 1 the argument is identical to that for Thm. 6.8 since, again, $\mathsf{c}\downarrow$ steps are not introduced for the same reason, cf. Not. 6.3.

For 2 the argument is identical to that for Thm. 6.11 since the argument of Prop. 6.9 remains valid and so there are still no contraction loops in the flow of a derivation in the image of R. $\qquad \square$

7. Conclusions

We have presented a series of upper bound results for the deep inference system KS. Polynomial simulations were given for truth tables and versions of Resolution, while polynomial-size proofs of functional and onto variants of the propositional pigeonhole principle were presented, yielding exponential separations from all these systems, as well as bounded-depth Frege systems.

We have seen that atomic flows can act as a powerful tool to analyse and manipulate derivations, and that often we can avoid the possibly exponential blowup arising from the $\mathsf{c}\downarrow$-$\mathsf{c}\uparrow$ rule. A relevant pursuit would be to investigate whether we can *always* avoid this blowup via, perhaps, a polynomial-time local or global flow reduction; this would imply that KS polynomially simulates KS^+, and thus quasipolynomially simulates SKS and Frege systems. Despite much work in this direction a result, positive or negative, seems difficult to obtain; this remains arguably the most important question in the proof complexity of deep inference.

Question 7.1. Does KS (quasi)polynomially simulate KS^+?

This question has already been asked in previous works, e.g. [BG09], [Jeř09], [Str12] and [Das11], with both positive and negative answers conjectured. We point out that our contribution might be helpful to any work towards a positive answer.

We also point out that it could be that atomic flows do not themselves include sufficient information to carry out an efficient normalisation procedure from KS^+ to KS, if one exists at all. To this end there is work ongoing by various researchers to augment atomic flows with certain *logical* information, in the hope of accessing further normalisation procedures and the ability to 'rebuild' deep inference proofs efficiently from flow objects.[9]

One might argue that our proofs in Sects. 4 and 5 eventually reduced to flows of bounded length, and so the complexity of normalisation was trivially polynomial. A more sophisticated situation might involve flows of bounded width and logarithmic length, again resulting in a polynomial blowup, or quasipolynomial width and polylogarithmic length, giving a quasipolynomial blowup. We refer the reader to the recent article [Das14] for an

---

[9]This is known to be impossible in polynomial time, by [Das13], under certain hardness assumptions.

example of this, where quasipolynomial-size proofs of the unrestricted pigeonhole principle are given in KS, utilising the techniques of this paper.

All the results that appear in this work, and indeed all other works on proof complexity of deep inference, e.g. [BG09] [Jeř09] [Das14], present only simulations and upper bounds. We currently know of no technique for proving lower bounds for deep inference systems and, in light of the question above and results in [Jeř09] and [Das14], it may be that such an endeavour, even for KS, could be as difficult as that of proving lower bounds for Frege systems.

Acknowledgements

References

[BG09]   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/PrCo mplDI.pdf`.

[BGGP10] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A quasipolynomial cut-elimination procedure in deep inference via atomic flows and threshold formulae. In Edmund M. Clarke and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, volume 6355 of *Lecture Notes in Computer Science*, pages 136–153. Springer-Verlag, 2010. `http://cs.bath.ac.uk/ag/p/QPNDI.pdf`.

[Brü03]   Kai Brünnler. Two restrictions on contraction. *Logic Journal of the IGPL*, 11(5):525–529, 2003. `http://www.iam.unibe.ch/~kai/Papers/RestContr.pdf`.

[BT01]   Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. Technical Report WV-01-02, Technische Universität Dresden, 2001. `http://iccl.tu-dresden.de/~kai/LocalClassic alLogic-tr.pdf`.

[Bus87]   Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52(4):916–927, 1987.

[D'A92]   Marcello D'Agostino. Are tableaux an improvement on truth-tables? *Journal of Logic, Language and Information*, 1:235–252, 1992. 10.1007/BF00156916.

[Das11]   Anupam Das. On the proof complexity of cut-free bounded deep inference. In Kai Brünnler and George Metcalfe, editors, *Tableaux 2011*, volume 6793 of *Lecture Notes in Artificial Intelligence*, pages 134–148. Springer-Verlag, 2011. `http://www.anupamdas.com/items/PrCompII/ProofComp lexityBoundedDI.pdf`.

[Das12a] Anupam Das. Characterising aspects of proof compression. `http://www.anupamdas.com/items/ CompMech/CompMech.pdf`, 2012.

[Das12b] Anupam Das. Complexity of deep inference via atomic flows. In S. Barry Cooper, Anuj Dawar, and Benedikt Löwe, editors, *Computability in Europe*, volume 7318 of *Lecture Notes in Computer Science*, pages 139–150. Springer-Verlag, 2012. `http://www.anupamdas.com/items/RelComp/Re lComp.pdf`.

[Das13]   Anupam Das. Rewriting with linear inferences in propositional logic. In Femke van Raamsdonk, editor, *RTA*, volume 21 of *LIPIcs*, pages 158–173. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[Das14]   Anupam Das. On the pigeonhole and related principles in deep inference and monotone systems. In *Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2014. To appear.

[EGM04]   Juan Luis Esteban, Nicola Galesi, and Jochen Messner. On the complexity of resolution with bounded conjunctions. In *IN 29TH INTERNATIONAL COLLOQUIUM ON AUTOMATA, LANGUAGES AND PROGRAMMING*, pages 220–231. Springer-Verlag, 2004.

[GG08]   Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1:9):1–36, 2008. `http://www.lmcs-online.org/ojs/viewarticle.php?id=341`.

[GGS10]   Alessio Guglielmi, Tom Gundersen, and Lutz Straßburger. Breaking paths in atomic flows for classical logic. In Jean-Pierre Jouannaud, editor, *25th Annual IEEE Symposium on Logic in Computer Science*, pages 284–293. IEEE, 2010. `http://www.lix.polytechnique.fr/~lutz/papers/AFII.pdf`.

[Gug03]   Alessio Guglielmi. Resolution in the calculus of structures. `http://cs.bath.ac.uk/ag/p/AG10.pdf`, 2003.

[Gun09]   Tom Gundersen. *A General View of Normalisation Through Atomic Flows*. PhD thesis, University of Bath, 2009.

[Jeř09]   Emil Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 19(2):323–339, 2009. `http://www.math.cas.cz/~jerabek/papers/cos.pdf`.

[KPW95]   Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures & Algorithms*, 7(1):15–39, 1995.

[Kra01]   Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[PBI93]   Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993. 10.1007/BF01200117.

[SBI02]   Nathan Segerlind, Sam Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-dnf resolution (extended abstract). In *SIAM J. Comput*, pages 604–613. IEEE, 2002.

[Str12]   Lutz Straßburger. Extension without cut. *Annals of Pure and Applied Logic*, 163(12):1995–2007, 2012.