A Quasipolynomial Normalisation Procedure in Deep Inference

Paola Bruscoli[1⋆], Alessio Guglielmi[1⋆⋆], Tom Gundersen[2⋆⋆⋆], and Michel Parigot[2†]

[1] University of Bath (UK)
[2] Laboratoire PPS, UMR 7126, CNRS & Université Paris 7 (France)

Abstract. Jeřábek showed in 2008 that cuts in propositional-logic deep-inference proofs can be eliminated in quasipolynomial time. The proof is an indirect one relying on a result of Atserias, Galesi and Pudlák about monotone sequent calculus and a correspondence between this system and cut-free deep-inference proofs. In this paper we give a direct proof of Jeřábek's result: we give a quasipolynomial-time cut-elimination procedure in propositional-logic deep inference. The main new ingredient is the use of a computational trace of deep-inference proofs called atomic flows, which are both very simple (they trace only structural rules and forget logical rules) and strong enough to faithfully represent the cut-elimination procedure. We also show how the technique can be extended to obtain a more general notion of normalisation called streamlining.

1 Introduction

Deep inference is a deduction framework (see [Gug07,BT01,Brü04]), where deduction rules apply arbitrarily deep inside formulae, contrary to traditional proof systems such as natural deduction and the sequent calculus, where deduction rules deal only with the outermost structure of formulae. This greater freedom is both a source of immediate technical difficulty and the promise, in the long run, of new powerful proof-theoretic methods. A general methodology allows to design deep-inference deduction systems having more symmetries and finer structural properties than the sequent calculus. For instance, cut and identity become really dual of each other, whereas they are only morally so in the sequent calculus, and all structural rules can be reduced to their atomic form, whereas this is false for contraction in the sequent calculus.

All usual logics have deep-inference deduction systems enjoying cut elimination (see [Gug] for a complete overview). The traditional methods of cut elimination of the

⋆ Supported by EPSRC grant EP/E042805/1 *Complexity and Non-determinism in Deep Inference* and by an ANR *Senior Chaire d'Excellence* titled *Identity and Geometric Essence of Proofs*.

⋆⋆ Supported by EPSRC grant EP/E042805/1 *Complexity and Non-determinism in Deep Inference* and by an ANR *Senior Chaire d'Excellence* titled *Identity and Geometric Essence of Proofs*.

⋆⋆⋆ Supported by an *Overseas Research Studentship* and a *Research Studentship*, both of the University of Bath, and by an ANR *Senior Chaire d'Excellence* titled *Identity and Geometric Essence of Proofs*.

† Supported by *Project INFER—Theory and Application of Deep Inference* of the *Agence Nationale de la Recherche*.

sequent calculus can be adapted to a large extent to deep inference, despite having to cope with a higher generality. New methods are also achievable. The standard proof system for propositional classical logic in deep inference is system SKS [BT01,Brü04]. Its cut elimination has been proved in several different ways [BT01,Brü04,GG08].

Recently, Jeřábek showed that cut elimination in SKS proofs can be done in quasipolynomial time [Jeř08], i.e., in time $n^{O(log(n))}$. The result is surprising because all known cut-elimination methods for classical-logic proof systems require exponential time, in particular for Gentzen's sequent calculus. Jeřábek obtained his result by relying on a construction over threshold functions by Atserias, Galesi and Pudlák, in the monotone sequent calculus [AGP02].

The technique that Jeřábek adopts is indirect because normalisation is performed over proofs in the sequent calculus, which are, in turn, related to deep-inference ones by polynomial simulations, originally studied in [Brü06] and [BG09].

In this paper we first give a direct proof of Jeřábek's result: that is, we give a quasipolynomial-time cut-elimination procedure in propositional-logic deep inference, which, in addition to being internal, has a strong computational flavour. We then extend this result to a more general normalisation procedure called streamlining, which applies to arbitrary derivations instead of proofs: instead of requiring the absence of cuts, one requires the absence of paths from an axiom to a cut. Our proof uses two ingredients:

1. an adaptation of Atserias-Galesi-Pudlák technique to deep inference, which simplifies the technicalities associated with the use of threshold functions; in particular, the formulae and derivations that we adopt are simpler than those in [AGP02];
2. a computational trace of deep-inference proofs called atomic flows, which are both very simple (they trace only structural rules and forget logical rules) and strong enough to faithfully represent cut elimination.

Atomic flows, which can be considered as specialised Buss flow graphs [Bus91], play a major role in designing and controlling the cut elimination procedure presented in this paper. They contribute to the overall clarification of this highly technical matter, by reducing our dependency on syntax. The techniques developed via atomic flows tolerate variations in the proof system specification. In fact, their geometric nature makes them largely independent of syntax, provided that certain linearity conditions are respected (and this is usually achievable in deep inference).

The paper is self-contained. Sections 2 and 3 are devoted, respectively, to the necessary background on deep inference and atomic flows. Threshold functions and formulae are introduced in Section 5.

We normalise proofs in two steps, each of which has a dedicated section in the paper:

1. We transform any given proof into what we call its 'simple form'. No use is made of threshold formulae and no significant proof complexity is introduced. This is presented in Section 4, which constitutes a good exercise on deep inference and atomic flows.
2. In Section 6, we show the cut elimination step, starting from proofs in simple form. Here, threshold formulae play a major role.

In section 7, we show how to extend the result to streamlining of arbitrary derivations.

In section 8, we discuss the relations of theses results with the work of Jeřábek, especially his extension of Atserias, Galesi and Pudlák result to MCLK.

Section 9 concludes the paper with some comments on future research directions.

Parts of this paper were presented at LPAR 16 [BGGP10].

2 Propositional Logic in Deep Inference: The SKS System

Formulae and Contexts

Two *units*, f (false) and t (true) and a countable set of *propositional letters*, denoted by p and q, are given. A primitive *negation* $\bar{\cdot}$ is defined on propositional letters: to each propositional letter p is associated its negation $\bar{p}$. *Atoms* are propositional letters and their negation; they are denoted a, b, c, d and e. Negation is extended to the set of atoms by defining $\bar{\bar{p}} = p$, for each negated propositional letter $\bar{p}$. This being classical logic, one can always exchange an atom with its negation: at the level of atoms, it does not matter which one is the propositional letter or its negation.

Formulae, denoted by A, B, C and D, are freely built from units and atoms using *disjunction* and *conjunction*. The disjunction and conjunction of two formulae A and B are denoted respectively $[A \vee B]$ and $(A \wedge B)$: the different brackets have the only purpose of improving legibility. We usually omit external brackets of formulae and sometimes we omit superfluous brackets under associativity. Negation can be extended to arbitrary formulae in an obvious way using De Morgan's laws, but we do not need it in this paper. We write $A \equiv B$ for literal equality of formulae.

We denote (*formula*) *contexts*, i.e., formulae with a hole, by $K\{\ \}$; for example, if $K\{a\}$ is $b \wedge [a \vee c]$, then $K\{\ \}$ is $b \wedge [\{\ \} \vee c]$, $K\{b\}$ is $b \wedge [b \vee c]$ and $K\{a \wedge d\}$ is $b \wedge [(a \wedge d) \vee c]$.

Derivations and Proofs

An (*inference*) *rule* ρ is an expression $\rho \dfrac{A}{B}$, where the formulae A and B are called *premiss* and *conclusion*, respectively.

The derivations are the *synchronal derivations* of open deduction, where connectives are acting derivations (see [GGP10] for the general definition).

Derivations are defined together with their *premiss* and *conclusion*, inductively as follows:

- if Φ is a unit or an atom, then Φ is a derivation with premiss Φ and conclusion Φ;
- if Φ is a derivation with premiss A and conclusion C and Ψ is a derivation with premiss B and conclusion D, then
  - $(\Phi \wedge \Psi)$ is a derivation with premiss $(A \wedge B)$ and conclusion $(C \wedge D)$
  - $[\Phi \vee \Psi]$ is a derivation with premiss $[A \vee B]$ and conclusion $[C \vee D]$
- if Φ is a derivation with premiss A and conclusion C, Ψ is a derivation with premiss B and conclusion D and $\rho \dfrac{C}{B}$ an inference rule, then the inference composition of Φ and Ψ is a derivation denoted $\dfrac{\Phi}{\Psi}$ with premiss A and conclusion D .

Concrete derivations are written in the natural two dimensional way. For example, if $\frac{A}{B}$, $\frac{B}{C}$ and $\frac{B \wedge E}{F}$ are rules, then

$$\left(\frac{B}{C} \wedge D\right) \vee \left[\frac{\left(\frac{A}{B} \wedge E\right)}{F} \vee G\right]$$

is a derivation with premiss $(B \wedge D) \vee [(A \wedge E) \vee G]$ and conclusion $(C \wedge D) \vee [F \vee G]$

A derivation, Φ, *from* A (*premiss*) *to* B (*conclusion*) is usually indicated in figures by

$$\begin{array}{c} A \\ \Phi \| \\ B \end{array} \qquad \text{or} \qquad \begin{array}{c} A \\ \| \\ B \end{array}$$

In text parts, we also use the notation $\Phi : A \to B$.

A *proof*, often denoted by Π, is a derivation with premiss $\mathtt{t}$.

The *size* $|A|$ (resp., $|\Phi|$) of a formula A (resp., derivation Φ) is the number of unit and atom occurrences appearing in it. The *length* (resp., *width*) of an atom is 1, the length (resp., width) of conjunction or disjunction of two derivations is the maximum (resp., sum) of the two subderivations, and the length (resp., width) of an inference composition of two derivations is the sum (resp., maximum) of the two subderivations. It follows that the size of a derivation is bounded by the product of its length and its width.

We allow derivations in formula contexts, for example, if $K\{\ \}$ is $b \wedge [\{\ \} \vee c]$, then $K\left\{\frac{\mathtt{t}}{a \vee \bar{a}}\right\}$ is $b \wedge \left[\frac{\mathtt{t}}{a \vee \bar{a}} \vee c\right]$.

Substitution

By $A\{a_1/B_1, \ldots, a_h/B_h\}$, we denote the operation of simultaneously substituting formulae B_1, …, B_h into all the occurrences of the atoms a_1, …, a_h in the formula A, respectively. By defining the substitution at the level of atoms, where atoms and their negation are equal citizens, we mean that the substitution to the occurrences of an atom does not touch the occurrences of its negation. Often, we only substitute certain occurrences of atoms: there will be no ambiguity because this is done in the context of atomic flows, where occurrences are distinguished with superscripts. The notion of substitution is extended to derivations in the natural way.

Inference Rules of SKS

Structural inference rules:

$$\mathsf{ai}\!\downarrow \frac{\mathsf{t}}{a \vee \bar{a}} \qquad \mathsf{aw}\!\downarrow \frac{\mathsf{f}}{a} \qquad \mathsf{ac}\!\downarrow \frac{a \vee a}{a}$$

identity (interaction) *weakening* *contraction*

$$\mathsf{ai}\!\uparrow \frac{a \wedge \bar{a}}{\mathsf{f}} \qquad \mathsf{aw}\!\uparrow \frac{a}{\mathsf{t}} \qquad \mathsf{ac}\!\uparrow \frac{a}{a \wedge a} \quad .$$

cut (cointeraction) *coweakening* *cocontraction*

Logical inference rules:

$$\mathsf{s}\,\frac{A \wedge [B \vee C]}{(A \wedge B) \vee C} \qquad \mathsf{m}\,\frac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \quad .$$

switch *medial*

There are also *equality* rules $= \frac{C}{D}$, for C and D on opposite sides in one of the following equations:

$$\begin{array}{rcl} A \vee B = B \vee A & \qquad & A \vee \mathsf{f} = A \\ A \wedge B = B \wedge A & & A \wedge \mathsf{t} = A \\ [A \vee B] \vee C = A \vee [B \vee C] & & \mathsf{t} \vee \mathsf{t} = \mathsf{t} \\ (A \wedge B) \wedge C = A \wedge (B \wedge C) & & \mathsf{f} \wedge \mathsf{f} = \mathsf{f} \end{array} \quad . \qquad (1)$$

Conventions

(a) In derivations we freely use equality rules without mentioning them. For instance

$$= \frac{= \dfrac{A}{A \wedge \mathsf{t}}}{\dfrac{\mathsf{t}}{p \vee \bar{p}} \wedge A} \quad \text{could be written} \quad = \frac{A}{\dfrac{\mathsf{t}}{p \vee \bar{p}} \wedge A} \quad .$$

(b) The structural rules have been given in atomic form in SKS. This is possible because in deep inference the general form of the structural rules, given below, is derivable from their atomic form, moreover it is derivable with a polynomial cost.

$$\mathsf{i}\!\downarrow \frac{\mathsf{t}}{A \vee \bar{A}} \qquad \mathsf{w}\!\downarrow \frac{\mathsf{f}}{A} \qquad \mathsf{c}\!\downarrow \frac{A \vee A}{A}$$

$$\mathsf{i}\!\uparrow \frac{A \wedge \bar{A}}{\mathsf{f}} \qquad \mathsf{w}\!\uparrow \frac{A}{\mathsf{t}} \qquad \mathsf{c}\!\uparrow \frac{A}{A \wedge A} \quad .$$

We will freely use a nonatomic rule instance to stand for some derivation in SKS that derives that instance.

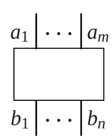
3 Atomic Flows

Atomic flows, which have been introduced in [GG08], are, essentially, specialised Buss flow graphs [Bus91]. They are directed graphs associated with SKS derivations: every derivation yields one atomic flow obtained by tracing the atoms (propositional letters and their negation) occurrences in the derivation. More precisely, one traces the behaviour of these occurrences through the structural rules: creation / destruction / duplication. No information about instances of logical rules is kept, only structural rules play a role and, as a consequence, infinitely many derivations correspond to each atomic flow. As shown in [GG08], it turns out that atomic flows contain sufficient structure to control cut elimination procedures, providing in particular induction measures that can be used to ensure termination. Such cut-elimination procedures require exponential time on the size of the derivation to be normalised. In the present work, we improve the complexity of cut elimination to quasipolynomial time, using in addition threshold formulae, which are independent from the given proof.

Abstract Notation of Atomic Flows

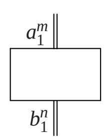
All edges of atomic flows are directed, but we do not explicitly show the orientation. Instead, we consider it as implicitly given by the way we draw them, i.e., edges are oriented along the vertical direction. Edges correspond to occurrences of atoms in a derivation, vertices to structural inferences rules. Edges are labelled by atom occurrences. Labels are omitted when not needed.

Top and down edges, i.e. those related to at most one a vertex, which correspond respectively to the premiss and conclusion of the derivation, play a special role for composing atomic flows. When only these edges matter, we adopt the following abstract representation of atomic flows



,

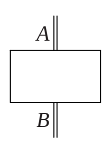
We also use a double line notation for representing multiple edges. The previous atomic flow becomes then



,

where a_1^m represents the sequence $a_1, \ldots, a_m$ and b_1^n, the sequence $b_1, \ldots, b_n$.

The flow associated to a derivation with premiss A and conclusion B will also be simply represented as



,

where A replaces the sequence $\overline{a}_m$ of atom occurrences of A and B replaces the sequence $\overline{b}_n$ of atom occurrences of B.

Atomic Flow Associated to a Derivation

To each derivation we associate inductively (say, in a top-down manner) an atomic flow as follows:

– to a unit we associate the empty flow, *e.g.,*

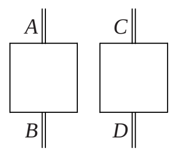
$$\mathsf{t} \quad \to \quad ;$$

– to an atom we associate the flow consisting of one edge, *e.g.,*

$$a \quad \to \quad a\big| \;;$$

– to the conjunction (resp., disjunction) of two derivations we associate the disjoint union of the flows of the derivations, *e.g.,*

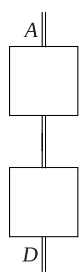
$$\begin{array}{c} A \\ \| \\ B \end{array} \vee \begin{array}{c} C \\ \| \\ D \end{array} \quad \to$$



;

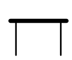
– to the inference composition of two derivations by a linear or equality rule we associate the disjoint union of the flows of the two derivations where the lower edges of the first flow are pairwise identified by the upper edges of the second flow according to the correspondance between atom occurrences in the inference rule instance (this is a one-to-one correspondence as the rule is linear), *e.g.,*

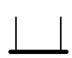
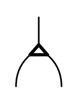
$$\rho \frac{\begin{array}{c} A \\ \| \\ B \end{array}}{\begin{array}{c} C \\ \| \\ D \end{array}} \quad \to$$



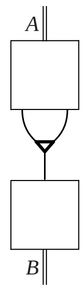
; and

– to the inference composition of two derivations by an atomic inference rule we associate the disjoint union of the flows of the two derivations where all the lower edges of the first flow and all the upper edges of the second flow are connected to a new vertex as shown below:

$\mathsf{ai}{\downarrow}\dfrac{\mathsf{t}}{a \vee \bar{a}} \quad \to$  $\qquad \mathsf{aw}{\downarrow}\dfrac{\mathsf{f}}{a} \quad \to$  $\qquad \mathsf{ac}{\downarrow}\dfrac{a \vee a}{a} \quad \to$ 

$\mathsf{ai}{\uparrow}\dfrac{a \wedge \bar{a}}{\mathsf{f}} \quad \to$  $\qquad \mathsf{aw}{\uparrow}\dfrac{a}{\mathsf{t}} \quad \to$  $\qquad \mathsf{ac}{\uparrow}\dfrac{a}{a \wedge a} \quad \to$  , *e.g.*,

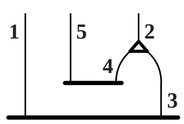
$$\begin{array}{c} A \\ \| \\ \dfrac{a \vee a}{a} \\ \| \\ B \end{array} \quad \rightarrow$$



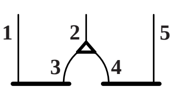
.

We qualify each vertex according to the rule it corresponds to: for example, in a given atomic flow, we might talk about a *contraction vertex*, or a *cut vertex*, and so on. Note that the vertices corresponding to dual rules are mutually distinct: for example, an identity vertex and a cut vertex are different because the orientation of their edges is different.

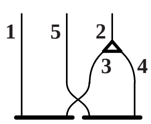
The horizontal direction plays no role in distinguishing atomic flows; this corresponds to commutativity of logical relations. Here are for instance three representations of the same flow:



,



and

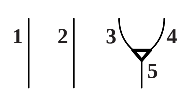


.

It should be noted that atomic flows built from derivations have no directed cycles and bear a natural polarity assignment (corresponding to atoms versus negated atoms in the derivation), that is a mapping of each edge to an element of $\{-,+\}$, such that the two edges of each identity or cut vertex map to different values and the three edges of each contraction or cocontraction vertex map to the same value. We denote atomic flows by ϕ and ψ.

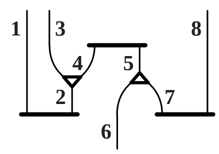
Examples of Atomic Flows Associated to Derivations

$$a^{\mathbf{1}} \wedge \left[b^{\mathbf{2}} \vee \frac{a^{\mathbf{3}} \vee a^{\mathbf{4}}}{a^{\mathbf{5}}} \right] \quad \rightarrow$$



.

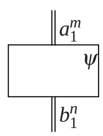
$$
= \dfrac{\left(\mathsf{s}\,\dfrac{\left(a^{\mathbf{1}} \wedge {=}\,\dfrac{\left[\bar{a}^{\mathbf{3}} \vee \dfrac{\mathsf{t}}{\bar{a}^{\mathbf{4}} \vee a^{\mathbf{5}}} \right]}{\left[\bar{a}^{\mathbf{3}} \vee \bar{a}^{\mathbf{4}} \right] \vee a^{\mathbf{5}}} \right)}{{=}\,\dfrac{\left[\dfrac{\left(a^{\mathbf{1}} \wedge \dfrac{\bar{a}^{\mathbf{3}} \vee \bar{a}^{\mathbf{4}}}{\bar{a}^{\mathbf{2}}} \right)}{\mathsf{f}} \vee a^{\mathbf{5}} \right]}{\dfrac{a^{\mathbf{5}}}{a^{\mathbf{6}} \wedge a^{\mathbf{7}}}}} \wedge a^{\mathbf{8}} \right)}{\left(a^{\mathbf{6}} \wedge \dfrac{a^{\mathbf{7}} \wedge \bar{a}^{\mathbf{8}}}{\mathsf{f}} \right)} \quad \rightarrow
$$



.

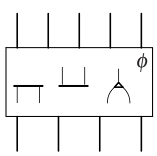
More Abstract Notation of Atomic Flows

When needed, we give names, like ϕ or ψ, to flows by putting the name on the corresponding box as follows:

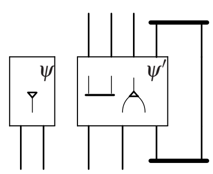


.

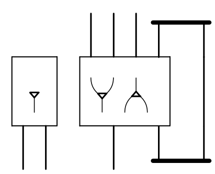
When certain information on the kind of vertices appearing in an atomic flow is needed, we enrich the previous representation by labelling the boxes with all the vertex kinds that can appear in the flow. For example:



,



and



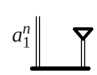
.

Note that when there no label on a box for specifying the vertex kinds that can appear, it means that the vertices in the corresponding flow can be any (so, it does not mean that there are no vertices in the flow).

We extend the double line notation to collections of isomorphic flows. For example, for $l \geq 0$, the following diagrams represent the same flow:

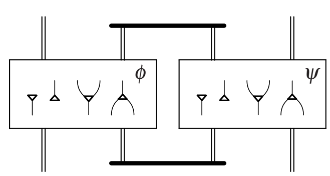
...

and



.

We observe that the flow of every SKS derivation can always be represented as follows:



.

4 Normalisation Step 1: Simple Form

The first step in our normalisation procedure, defined here, consists in routine deep-inference manipulations, which are best understood in conjunction with atomic flows. For this reason, this section is a useful exercise for a reader who is not familiar with deep inference and atomic flows.

In Theorem 5 of this section, we show that every proof can be transformed into 'simple form'. Proofs in simple form are such that we can substitute formulae for all the atom occurrences that appear in cut instances, without substituting for atom occurrences that appear in the conclusion of the derivation. Of course, doing this would invalidate identity and cut instances, but in Section 6 we see how we can build a valid cut-free proof from the broken derivations obtained by substituing formulae into derivations in simple form.

We first show some standard deep-inference results. We will see how we can permute all the identity (resp., cut) rule instances to the top (resp., bottom) of a proof, without changing the atomic flow of the proof, and without significantly changing the size of the proof.

Lemma 1. *Given a context $K\{\ \}$ and a formula A, there exist derivations*

$$\begin{array}{c} A \wedge K\{\mathsf{t}\} \\ \| \{\mathsf{s}\} \\ K\{A\} \end{array} \quad \textit{and} \quad \begin{array}{c} K\{A\} \\ \| \{\mathsf{s}\} \\ K\{\mathsf{f}\} \vee A \end{array} ,$$

each of whose width is the size of $K\{A\}$ plus one and length is bounded by a polynomial in the size of $K\{\ \}$.

Proof. The result follows by structural induction on $K\{\ \}$: The base cases are:

$$= \frac{A \wedge \{\mathsf{t}\}}{\{A\}} \quad \text{and} \quad = \frac{\{A\}}{\{\mathsf{f}\} \vee A} .$$

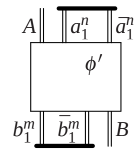
The inductive cases are

$$\begin{array}{c} = \dfrac{A \wedge (B \wedge K\{\mathsf{t}\})}{B \wedge (A \wedge K\{\mathsf{t}\})} \\ \| \{\mathsf{s}\} \\ B \wedge K\{A\} \end{array} , \quad \begin{array}{c} \mathsf{s} \dfrac{A \wedge [B \vee K\{\mathsf{t}\}]}{B \vee (A \wedge K\{\mathsf{t}\})} \\ \| \{\mathsf{s}\} \\ B \vee K\{A\} \end{array} \quad \text{and}$$

$$\begin{array}{c} B \wedge K\{A\} \\ \| \{\mathsf{s}\} \\ \mathsf{s} \dfrac{B \wedge [K\{\mathsf{f}\} \vee A]}{(B \wedge K\{\mathsf{f}\}) \vee A} \end{array} , \quad \begin{array}{c} B \vee K\{A\} \\ \| \{\mathsf{s}\} \\ = \dfrac{B \vee [K\{\mathsf{f}\} \vee A]}{[B \vee K\{\mathsf{f}\}] \vee A} \end{array} .$$

□

Lemma 2. *Given a derivation $\Phi : A \to B$, with flow*

$\phi =$  ,

there exists a derivation

$$\Psi = \begin{array}{c} \left(A \wedge \dfrac{\mathsf{t}}{a_1 \vee \bar{a}_1} \wedge \cdots \wedge \dfrac{\mathsf{t}}{a_n \vee \bar{a}_n} \right) \\ \Psi' \| \\ \left[\dfrac{b_1 \wedge \bar{b}_1}{\mathsf{f}} \vee \cdots \vee \dfrac{b_m \wedge \bar{b}_m}{\mathsf{f}} \vee B \right] \end{array} ,$$

for some atoms $a_1, \ldots, a_n, b_1, \ldots, b_m$ and some derivation Ψ', such that the flow of Ψ is ϕ, the flow of Ψ' is ϕ' and the size of Ψ is bounded by a polynomial in the size of Φ.

Proof. For every relevant interaction we perform the following transformation:

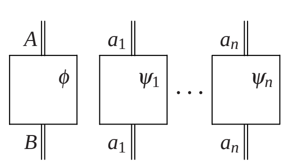
$$\begin{array}{c} A \\ \Phi' \| \\ K\left\{ \dfrac{\mathsf{t}}{a \vee \bar{a}} \right\} \\ \Phi'' \| \\ B \end{array} \quad \text{is transformed into} \quad \begin{array}{c} \left(\dfrac{\mathsf{t}}{a \vee \bar{a}} \wedge \begin{array}{c} A \\ \Phi' \| \\ K\{\mathsf{t}\} \end{array} \right) \\ \| \{\mathsf{s}\} \\ K\,[a \vee \bar{a}] \\ \Phi'' \| \\ B \end{array} ,$$

which is possible by Lemma 1. Instances of cut rules can be dealt with in a symmetric way.

Each transformation increases the width of the derivation by a constant and increases the length by at most a polynomial in the width of the derivation. Hence, the size of Ψ is bounded by a polynomial in the size of Φ. □

We now show how to extend substitutions from formulae to derivations. Using atomic flows, we single out some atom occurrences that we substitute for. Substitutions play a crucial role in Theorem 5 and in Lemma 11. It is important to notice that a substitution only copies atomic flows, it does not introduce new vertices; and that the cost of substitution is polynomial.

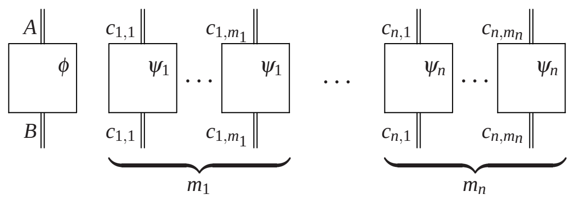
Lemma 3. *Given a derivation $\Phi : A \to B$, let its associated flow have shape*

 ,

such that, for $1 \leq i \leq n$, all the edges of ψ_i are mapped to from occurrences of a_i, then, for any formulae $C_1, \ldots, C_n$ there exists a derivation

$$\Psi \; = \; \begin{array}{c} A\{a_1^{\psi_1}/C_1, \ldots, a_n^{\psi_n}/C_n\} \\ \| \\ B\{a_1^{\psi_1}/C_1, \ldots, a_n^{\psi_n}/C_n\} \end{array} \quad ,$$

whose flow is



where, for every $1 \leq i \leq n$, the atom occurrences of C_i are $c_{i,1}, \ldots, c_{i,m_i}$; moreover, the size of Ψ is bounded by a polynomial in the size of Φ and, for each $1 \leq i \leq n$, the size of C_i.

Proof. We sketch the proof. For each $1 \leq i \leq n$, we can proceed by structural induction on C_i and then on ψ_i. For the two cases of $C_i \equiv D \vee E$ and $C_i \equiv D \wedge E$ we have to consider, for each vertex of ψ_i, one of the following situations (notice that ψ_i can not contain interaction or cut vertices):

$$= \frac{\mathsf{f}}{\left[\dfrac{\mathsf{f}}{D} \vee \dfrac{\mathsf{f}}{E}\right]} \quad , \qquad = \frac{\mathsf{f}}{\left(\dfrac{\mathsf{f}}{D} \wedge \dfrac{\mathsf{f}}{E}\right)} \quad ,$$

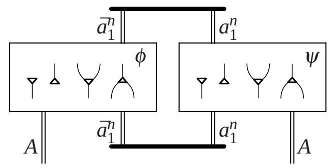
$$= \frac{[D \vee E] \vee [D \vee E]}{\left[\dfrac{D \vee D}{D} \vee \dfrac{E \vee E}{E}\right]} \quad , \qquad \mathsf{m}\,\frac{(D \wedge E) \vee (D \wedge E)}{\left(\dfrac{D \vee D}{D} \wedge \dfrac{E \vee E}{E}\right)} \quad ,$$

and their dual ones. □

Notation 4. When we write $\Phi\{a_1^{\psi_1}/C_1, \ldots, a_n^{\psi_n}/C_n\}$, we mean the derivation Ψ obtained in the proof of Lemma 3.

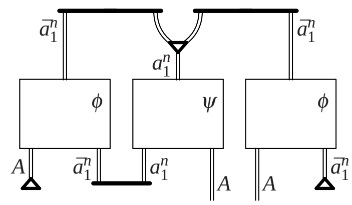
We now present the main result of this section. We show that any derivation can be transformed into a derivation whose atomic flow is in what we call 'simple form'. Referring to the second flow in Theorem 5, we observe that we could substitute for the atom occurrences corresponding to the leftmost copy of ϕ without substituting for any atom occurrence appearing in the conclusion of the proof. This is one of the two main ingredients in our normalisation procedure.

Theorem 5. *Given a proof Φ of A, with flow*



,

where only occurrences of the atoms $\bar{a}_1, \ldots, \bar{a}_n$ are mapped to edges in ϕ, there exists a proof Ψ of A, with flow



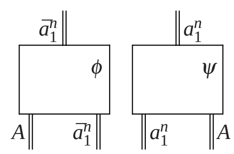
,

such that the size of Ψ is bounded by a polynomial in the size of Φ.

Proof. For every $1 \leq i \leq n$, let m_i (resp., m'_i) be the number of interactions (resp., cuts) where $\bar{a}_i^\phi$ appear in Φ, and consider the derivation

$$\Phi' = \begin{array}{c} \bigwedge_{0\leq j\leq m_1}\left[a_1^\psi \vee \bar{a}_1^\phi\right] \wedge \cdots \wedge \bigwedge_{0\leq j\leq m_n}\left[a_n^\psi \vee \bar{a}_n^\phi\right] \\ \| \\ \bigvee_{0\leq j\leq m'_1}\left(a_1^\psi \wedge \bar{a}_1^\phi\right) \vee \cdots \vee \bigvee_{0\leq j\leq m'_n}\left(a_n^\psi \wedge \bar{a}_n^\phi\right) \vee A \end{array} ,$$

with atomic flow



,

which exists and whose size is bounded by a polynomial in the size of Φ by Lemma 2. We will now replace every atom instance mapped to from the flow ϕ by two copies of itself, and thus duplicate the flow ϕ. Let

$$\sigma = \{\bar{a}_1^\phi / (\bar{a}_1 \wedge \bar{a}_1), \ldots, \bar{a}_n^\phi / (\bar{a}_n \wedge \bar{a}_n)\} \ ,$$

and construct Ψ:

$$\left(\mathsf{s}\frac{\dfrac{\mathsf{t}}{\left(\bar{a}_1^\phi \wedge \dfrac{\mathsf{t}}{\bar{a}_1^\phi \vee a_1}\right) \vee a_1}}{\left[\left(\bar{a}_1^\phi \wedge \bar{a}_1^\phi\right) \vee \dfrac{a_1 \vee a_1}{a_1^\psi}\right]} \wedge \cdots \wedge \mathsf{s}\frac{\dfrac{\mathsf{t}}{\left(\bar{a}_n^\psi \wedge \dfrac{\mathsf{t}}{\bar{a}_n^\psi \vee a_n}\right) \vee a_n}}{\left[\left(\bar{a}_n^\phi \wedge \bar{a}_n^\phi\right) \vee \dfrac{a_n \vee a_n}{a_n^\psi}\right]} \right) ,$$

$$\Phi'\sigma \|$$

$$\left[\left(\frac{\bar{a}_1^\phi \wedge a_1^\psi}{\mathsf{f}} \wedge \frac{\bar{a}_1^\phi}{\mathsf{t}} \right) \vee \cdots \vee \left(\frac{\bar{a}_n^\phi \wedge a_n^\psi}{\mathsf{f}} \wedge \frac{\bar{a}_n^\phi}{\mathsf{t}} \right) \vee \begin{array}{c} A\sigma \\ \|\{\mathsf{aw}\uparrow\} \\ A \end{array} \right]$$

with the required atomic flow, where, by Lemma 3, the derivation $\Phi'\sigma$ exists and its size is bounded by a polynomial in the size of Φ'. □

5 Threshold Formulae

Threshold formulae realise boolean threshold functions, which are defined as boolean functions that are true if and only if at least k of n inputs are true (see [Weg87] for a thorough reference on threshold functions).

There are several ways of encoding threshold functions into formulae, and the problem is to find, among them, an encoding that allows us to obtain Theorem 10. Efficiently obtaining the property stated in Theorem 10 crucially depends also on the proof system we adopt.

The following class of threshold formulae, which we found to work for system SKS, is a simplification of the one adopted in [AGP02].

In the rest of this paper, whenever we have a sequence of atoms $a_1, \ldots, a_n$, we will assume, without loss of generality, that n is a power of two.

Definition 6. *For every $n = 2^m$, with $m \geq 0$, and $k \geq 0$, we define the operator θ^n_k inductively as follows:*

$$\theta^n_k(a_1,\ldots,a_n) = \\ = \begin{cases} \mathsf{t} & \text{if } k = 0 \\ \mathsf{f} & \text{if } k > n \\ a_1 & \text{if } n = k = 1 \\ \bigvee_{\substack{i+j=k \\ 0\leq i,j\leq n/2}} \left(\theta^{n/2}_i(a_1,\ldots,a_{n/2}) \wedge \theta^{n/2}_j(a_{n/2+1},\ldots,a_n)\right) & \text{otherwise.} \end{cases}$$

For any n atoms $a_1, \ldots, a_n$, we call $\theta^n_k(a_1,\ldots,a_n)$ the threshold formula at level k *(with respect to $a_1, \ldots, a_n$).*

See, in Figure 1, some examples of threshold formulae.

The size of the threshold formulae dominates the cost of the normalisation procedure, so, we evaluate their size.

Lemma 7. *For any $n = 2^m$, with $m \geq 0$, and $k \geq 0$ the size of $\theta^n_k(a_1,\ldots,a_n)$ has a quasipolynomial bound in n.*

Proof. We show that the size of $\theta^n_k(a_1,\ldots,a_n)$ is bounded by $n^{2\log n}$. We reason by induction on n; the case $n = 1$ trivially holds. For $n > 1$, we consider that the size of $\theta^n_k(a_1,\ldots,a_n)$ is bounded by

$$\sum_{\substack{i+j=k \\ 0\leq i\leq n/2 \\ 0\leq j\leq n/2}} 2(n/2)^{2\log n/2} \quad .$$

$$
\begin{aligned}
\theta^2_0(a,b) &\equiv \mathtt{t} \quad , \\
\theta^2_1(a,b) &\equiv (\theta^1_1(a)\wedge\theta^1_0(b))\vee(\theta^1_0(a)\wedge\theta^1_1(b)) \equiv (a\wedge \mathtt{t})\vee(\mathtt{t}\wedge b) \\
&= a\vee b \quad , \\
\theta^2_2(a,b) &\equiv \theta^1_1(a)\wedge\theta^1_1(b) \\
&\equiv a\wedge b \quad , \\
\theta^3_0(a,b,c) &\equiv \mathtt{t} \quad , \\
\theta^3_1(a,b,c) &\equiv (\theta^1_1(a)\wedge\theta^2_0(b,c))\vee(\theta^1_0(a)\wedge\theta^2_1(b,c)) \equiv (a\wedge \mathtt{t})\vee(\mathtt{t}\wedge[(b\wedge \mathtt{t})\vee(\mathtt{t}\wedge c)]) \\
&= a\vee b\vee c \quad , \\
\theta^3_2(a,b,c) &\equiv (\theta^1_1(a)\wedge\theta^2_1(b,c))\vee(\theta^1_0(a)\wedge\theta^2_2(b,c)) \\
&= (a\wedge[b\vee c])\vee(b\wedge c) \quad , \\
\theta^3_3(a,b,c) &\equiv \theta^1_1(a)\wedge\theta^2_2(b,c) \equiv (a\wedge(b\wedge c)) \\
&= a\wedge b\wedge c \quad , \\
\theta^5_0(a,b,c,d,e) &\equiv \mathtt{t} \quad , \\
\theta^5_1(a,b,c,d,e) &\equiv (\theta^2_1(a,b)\wedge\theta^3_0(c,d,e))\vee(\theta^2_0(a,b)\wedge\theta^3_1(c,d,e)) \\
&= a\vee b\vee c\vee d\vee e \quad , \\
\theta^5_2(a,b,c,d,e) &\equiv (\theta^2_2(a,b)\wedge\theta^3_0(c,d,e))\vee(\theta^2_1(a,b)\wedge\theta^3_1(c,d,e))\vee(\theta^2_0(a,b)\wedge\theta^3_2(c,d,e)) \\
&= (a\wedge b)\vee([a\vee b]\wedge[c\vee d\vee e])\vee(c\wedge[d\vee e])\vee(d\wedge e) \quad , \\
\theta^5_3(a,b,c,d,e) &\equiv (\theta^2_2(a,b)\wedge\theta^3_1(c,d,e))\vee(\theta^2_1(a,b)\wedge\theta^3_2(c,d,e))\vee(\theta^2_0(a,b)\wedge\theta^3_3(c,d,e)) \\
&= (a\wedge b\wedge[c\vee d\vee e])\vee([a\vee b]\wedge[(c\wedge[d\vee e])\vee(d\wedge e)])\vee(c\wedge d\wedge e) \quad , \\
\theta^5_4(a,b,c,d,e) &\equiv (\theta^2_2(a,b)\wedge\theta^3_2(c,d,e))\vee(\theta^2_1(a,b)\wedge\theta^3_3(c,d,e)) \\
&= (a\wedge b\wedge[(c\wedge[d\vee e])\vee(d\wedge e)])\vee([a\vee b]\wedge c\wedge d\wedge e) \quad , \\
\theta^5_5(a,b,c,d,e) &\equiv \theta^2_2(a,b)\wedge\theta^3_3(c,d,e) \\
&= a\wedge b\wedge c\wedge d\wedge e \quad , \\
\theta^5_6(a,b,c,d,e) &\equiv \mathtt{f} \quad .
\end{aligned}
$$

Fig. 1. Examples of threshold formulae.

We then have

$$\sum_{\substack{i+j=k\\0\leq i\leq n/2\\0\leq j\leq n/2}} 2(n/2)^{2\log n/2} \leq \sum_{\substack{i+j=n/2\\0\leq i\leq n/2\\0\leq j\leq n/2}} 2(n/2)^{2\log n/2} \leq (n+2)(n/2)^{2\log n/2} \quad ,$$

and since $n+2 \leq n^2$ and $n/2 < n$, we have

$$(n+2)(n/2)^{2\log n/2} \leq n^2 n^{2\log n/2} = n^{2\log n - 2\log 2 + 2} = n^{2\log n} \quad ,$$

as required. □

Lemma 8. *For any $n = 2^m$, with $m \geq 0$, $k \geq 0$ and $1 \leq i \leq n$, there exists a derivation*

$$\Gamma_k^i \; = \; \begin{array}{c} \theta_k^n(a_1,\ldots,a_n)\{a_i/\mathsf{f}\} \\ \| \{\mathsf{aw}\downarrow,\mathsf{aw}\uparrow\} \\ \theta_{k+1}^n(a_1,\ldots,a_n)\{a_i/\mathsf{t}\} \end{array} \quad ,$$

whose size has a quasipolynomial bound in n.

Proof. The result follows by Lemma 7 and structural induction on Definition 6:

– For $n > 1$ and $1 \leq l \leq n$, we define the derivations $\Upsilon_k^l(a_1,\ldots,a_n)$ and $\Delta_k^l(a_1,\ldots,a_n)$ as follows:

$$\Upsilon_k^l(a_1,\ldots,a_n) = \begin{cases} \mathsf{w}\uparrow \dfrac{(\theta_{n/2}^{n/2}(a_1,\ldots,a_{n/2}))\{a_l/\mathsf{f}\} \wedge \theta_{k-n/2}^{n/2}(a_{n/2+1},\ldots,a_n)}{\mathsf{f}} & \text{if } n/2 \leq k \leq n \text{ and } l \leq n/2 \\[2ex] \mathsf{w}\uparrow \dfrac{\theta_{k-n/2}^{n/2}(a_1,\ldots,a_{n/2}) \wedge (\theta_{n/2}^{n/2}(a_{n/2+1},\ldots,a_n))\{a_l/\mathsf{f}\}}{\mathsf{f}} & \text{if } n/2 \leq k \leq n \text{ and } n/2 < l \\[2ex] \mathsf{f} & \text{otherwise} \end{cases}$$

and

$$\Delta_k^l(a_1,\ldots,a_n) = \begin{cases} \mathsf{w}\downarrow \dfrac{\mathsf{f}}{\theta_k^{n/2}(a_{n/2+1},\ldots,a_n)} & \text{if } 0 < k \leq n/2 \text{ and } l \leq n/2 \\[2ex] \mathsf{w}\downarrow \dfrac{\mathsf{f}}{\theta_k^{n/2}(a_1,\ldots,a_{n/2})} & \text{if } 0 < k \leq n/2 \text{ and } n/2 < l \\[2ex] \mathsf{f} & \text{otherwise} \end{cases} \quad .$$

– For $k \geq 0$ and $1 \leq l \leq n$, we define the derivations $\Gamma_k^l(a_1,\ldots,a_n)$, recursively on n, as follows:
  - $\Gamma_0^1(a_1) = \mathsf{t}$;
  - for $k > 0$, $\Gamma_k^1(a_1) = \mathsf{f}$;
  - for $k > n$, $\Gamma_k^l(a_1,\ldots,a_n) = \mathsf{f}$;

- for $n > 1$, $k \leq n$ and $l \leq n/2$, let $\Gamma_k^l(a_1, \ldots, a_n)$ be

$$\bigvee_{\substack{i+j=k \\ 0 \leq i < n/2 \\ 0 \leq j \leq n/2}} \Big(\Gamma_i^l(a_1, \ldots, a_{n/2}) \wedge \theta_j^{n/2}(a_{n/2+1}, \ldots, a_n)\Big) \vee \Upsilon_k^l(a_1, \ldots, a_n) \vee \Delta_{k+1}^l(a_1, \ldots, a_n)$$

- for $n > 1$, $k \leq n$ and $n/2 < l$, let $\Gamma_k^l(a_1, \ldots, a_n)$ be

$$\bigvee_{\substack{i+j=k \\ 0 \leq i \leq n/2 \\ 0 \leq j < n/2}} \Big(\theta_i^{n/2}(a_1, \ldots, a_{n/2}) \wedge \Gamma_j^{l-n/2}(a_{n/2+1}, \ldots, a_n)\Big) \vee \Upsilon_k^l(a_1, \ldots, a_n) \vee \Delta_{k+1}^l(a_1, \ldots, a_n) .$$

It is worth noting that both the premiss and the conclusion of Γ_k^i are logically equivalent to $\theta_k^{n-1}(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. □

See, in Figure 2, some examples of Γ_k^i.

Lemma 9. *Given a formula A and an atom a that occurs in A, there exist derivations* $\begin{array}{c} a \wedge A\{a/\mathsf{t}\} \\ \|\{\mathsf{ac}\uparrow,\mathsf{s}\} \\ A \end{array}$ *and* $\begin{array}{c} A \\ \|\{\mathsf{ac}\downarrow,\mathsf{s}\} \\ A\{a/\mathsf{f}\} \vee a \end{array}$ *such that their sizes are both bounded by a polynomial in the size of A.*

Proof. The result follows by induction on the number of occurrences of a in A, and Lemma 1. □

We now present the main result of this section. We show that, using threshold formulae, we are able to deduce a conjunction of disjunctions from a disjunction of (slightly different) conjunctions. This construction is based on (seen top-down) contractions meeting cocontrations, and can be thought of as a generalisation of the simple sharing mechanism that allows us to deduce $a \wedge \cdots \wedge a$ from $a \vee \cdots \vee a$.

In Lemma 11 we will see how using this sharing mechanism allows us to glue together several 'broken' derivations in order to build a cut-free proof.

Theorem 10. *Let, for some $n = 2^m$ with $m \geq 0$, $a_1, \ldots, a_n$ be distinct atoms. Then, for every $1 \leq k \leq n+1$, there exists a derivation*

$$\Gamma_k \ = \ \begin{array}{c} \big(a_1 \wedge \theta_{k-1}^n(a_1, \ldots, a_n)\{a_1/\mathsf{f}\}\big) \vee \cdots \vee \big(a_n \wedge \theta_{k-1}^n(a_1, \ldots, a_n)\{a_n/\mathsf{f}\}\big) \\ \Big\|\mathsf{SKS}\setminus\{\mathsf{ai}\downarrow,\mathsf{ai}\uparrow\} \\ \big[a_1 \vee \theta_k^n(a_1, \ldots, a_n)\{a_1/\mathsf{f}\}\big] \wedge \cdots \wedge \big[a_n \vee \theta_k^n(a_1, \ldots, a_n)\{a_n/\mathsf{f}\}\big] \end{array} ,$$

such that the size of Γ_k has a quasipolynomial bound in n.

$$\varGamma_0^1\,\hat{a} = \mathsf{t} \vee \frac{\mathsf{f}}{b} \vee \frac{\mathsf{f}}{c \vee d \vee e} \quad ,$$

$$\varGamma_1^1\,\hat{a} = b \vee \left(\left[\mathsf{t} \vee \frac{\mathsf{f}}{b} \right] \wedge [c \vee d \vee e] \right) \vee \frac{\mathsf{f}}{(c \wedge [d \vee e]) \vee (d \wedge e)} \quad ,$$

$$\varGamma_2^1\,\hat{a} = (b \wedge [c \vee d \vee e]) \vee \left(\left[\mathsf{t} \vee \frac{\mathsf{f}}{b} \right] \wedge [(c \wedge [d \vee e]) \vee (d \wedge e)] \right) \vee \frac{\mathsf{f} \wedge b}{\mathsf{f}} \vee \frac{\mathsf{f}}{c \wedge d \wedge e} \quad ,$$

$$\varGamma_3^1\,\hat{a} = (b \wedge [(c \wedge [d \vee e]) \vee (d \wedge e)]) \vee \left(\left[\mathsf{t} \vee \frac{\mathsf{f}}{b} \right] \wedge c \wedge d \wedge e \right) \vee \frac{\mathsf{f} \wedge b \wedge [c \vee d \vee e]}{\mathsf{f}} \quad ,$$

$$\varGamma_4^1\,\hat{a} = (b \wedge c \wedge d \wedge e) \vee \frac{\mathsf{f} \wedge b \wedge [(c \wedge [d \vee e]) \vee (d \wedge e)]}{\mathsf{f}} \quad ,$$

$$\varGamma_5^1\,\hat{a} = \frac{\mathsf{f} \wedge b \wedge c \wedge d \wedge e}{\mathsf{f}} \quad ,$$

$$\varGamma_0^3\,\hat{a} = \mathsf{t} \vee \frac{\mathsf{f}}{d \vee e} \vee \frac{\mathsf{f}}{a \vee b} \quad ,$$

$$\varGamma_1^3\,\hat{a} = \left([a \vee b] \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d \vee e} \right] \right) \vee d \vee e \vee \frac{\mathsf{f}}{d \wedge e} \vee \frac{\mathsf{f}}{a \wedge b} \quad ,$$

$$\varGamma_2^3\,\hat{a} = \left(a \wedge b \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d \vee e} \right] \right) \vee \left([a \vee b] \wedge \left[d \vee e \vee \frac{\mathsf{f}}{d \wedge e} \right] \right) \vee (d \wedge e) \vee \frac{\mathsf{f} \wedge [d \vee e]}{\mathsf{f}} \quad ,$$

$$\varGamma_3^3\,\hat{a} = \left(a \wedge b \wedge \left[d \vee e \vee \frac{\mathsf{f}}{d \wedge e} \right] \right) \vee \left([a \vee b] \wedge \left[(d \wedge e) \vee \frac{\mathsf{f} \wedge [d \vee e]}{\mathsf{f}} \right] \right) \vee \frac{\mathsf{f} \wedge d \wedge e}{\mathsf{f}} \quad ,$$

$$\varGamma_4^3\,\hat{a} = \left(a \wedge b \wedge \left[(d \wedge e) \vee \frac{\mathsf{f} \wedge [d \vee e]}{\mathsf{f}} \right] \right) \vee \frac{[a \vee b] \wedge \mathsf{f} \wedge d \wedge e}{\mathsf{f}} \quad ,$$

$$\varGamma_5^3\,\hat{a} = \frac{a \wedge b \wedge \mathsf{f} \wedge d \wedge e}{\mathsf{f}} \quad ,$$

$$\varGamma_0^5\,\hat{a} = \mathsf{t} \vee \frac{\mathsf{f}}{d} \vee \frac{\mathsf{f}}{c} \vee \frac{\mathsf{f}}{a \vee b} \quad ,$$

$$\varGamma_1^5\,\hat{a} = \left([a \vee b] \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d} \vee \frac{\mathsf{f}}{c} \right] \right) \vee \left(c \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d} \right] \right) \vee d \vee \frac{\mathsf{f}}{a \wedge b} \quad ,$$

$$\varGamma_2^5\,\hat{a} = \left(a \wedge b \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d} \vee \frac{\mathsf{f}}{c} \right] \right) \vee \left([a \vee b] \wedge \left[\left(c \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d} \right] \right) \vee d \right] \right) \vee (c \wedge d) \vee \frac{d \wedge \mathsf{f}}{\mathsf{f}} \quad ,$$

$$\varGamma_3^5\,\hat{a} = \left(a \wedge b \wedge \left[\left(c \wedge \left[\mathsf{t} \vee \frac{\mathsf{f}}{d} \right] \right) \vee d \right] \right) \vee \left([a \vee b] \wedge \left[(c \wedge d) \vee \frac{d \wedge \mathsf{f}}{\mathsf{f}} \right] \right) \vee \frac{c \wedge d \wedge \mathsf{f}}{\mathsf{f}} \quad ,$$

$$\varGamma_4^5\,\hat{a} = \left(a \wedge b \wedge \left[(c \wedge d) \vee \frac{d \wedge \mathsf{f}}{\mathsf{f}} \right] \right) \vee \frac{[a \vee b] \wedge c \wedge d \wedge \mathsf{f}}{\mathsf{f}} \quad ,$$

$$\varGamma_5^5\,\hat{a} = \frac{a \wedge b \wedge c \wedge d \wedge \mathsf{f}}{\mathsf{f}} \quad .$$

Fig. 2. Examples of $\varGamma_k^l\,\hat{a}$, where $\hat{a} = (a, b, c, d, e)$.

Proof. For $1 \leq k \leq n+1$, we construct:

$$\Gamma_k \quad = \quad \begin{array}{c} \left[\left(a_1 \;\wedge\; \begin{array}{c}\theta^n_{k-1}(a_1,\dots,a_n)\{a_1/\mathsf{f}\}\\ \Gamma^1_k \Big\| \{\mathsf{aw}\!\downarrow,\mathsf{aw}\!\uparrow\} \\ \theta^n_k(a_1,\dots,a_n)\{a_1/\mathsf{t}\}\end{array}\right) \;\vee\; \cdots \;\vee\; \left(a_n \;\wedge\; \begin{array}{c}\theta^n_{k-1}(a_1,\dots,a_n)\{a_n/\mathsf{f}\}\\ \Gamma^n_k \Big\| \{\mathsf{aw}\!\downarrow,\mathsf{aw}\!\uparrow\} \\ \theta^n_k(a_1,\dots,a_n)\{a_n/\mathsf{t}\}\end{array}\right)\right] \\ \Phi_1 \Big\| \{\mathsf{ac}\!\uparrow,\mathsf{s}\} \\ \bigvee_{1\le i\le n} \theta^n_k(a_1,\dots,a_n) \\ \Big\|\{\mathsf{c}\!\downarrow\} \\ \theta^n_k(a_1,\dots,a_n) \\ \Big\|\{\mathsf{c}\!\uparrow\} \\ \bigwedge_{1\le i\le n}\theta^n_k(a_1,\dots,a_n) \\ \Phi_2\Big\|\{\mathsf{ac}\!\downarrow,\mathsf{s}\} \\ \left[a_1\vee\theta^n_k(a_1,\dots,a_n)\{a_1/\mathsf{f}\}\right]\wedge\cdots\wedge\left[a_n\vee\theta^n_k(a_1,\dots,a_n)\{a_n/\mathsf{f}\}\right] \end{array} \quad ,$$

where Φ_1 and Φ_2 exist by Lemma 9 and, for $1 \leq i \leq n$, Γ^i_k exists by Lemma 8. The size of Γ_k is quasipolynomial in n, by Lemma 8 and Lemma 9. □

6 Normalisation Step 2: Cut Elimination

We now show the main construction of this paper, a cut-elimination result for derivations in simple form. The procedure uses a class of external and independent derivations in order to glue together pieces of the original proof. One valid class of such derivations are the ones shown in Section 5.

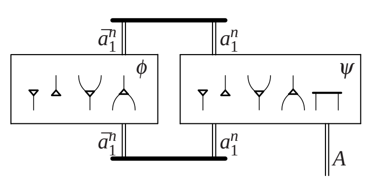
Lemma 11. *Let*

1. $N > 0$ *be an integer;*
2. $a_1, \dots, a_n$ *be distinct atoms, where* $n = 2^m$ *for some* $m \geq 0$*;*
3. *there be, for every* $0 < k < N$ *and* $1 \leq i \leq n$*, a formula* $C^{a_i}_k$*;*
4. *there be, for every* $1 \leq k \leq N$*, a derivation*

$$\Gamma_k \;=\; \begin{array}{c} \left(a_1 \wedge C^{a_1}_{k-1}\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_{k-1}\right) \\ \Big\| \mathsf{SKS}\setminus\{\mathsf{ai}\!\downarrow,\mathsf{ai}\!\uparrow\} \\ \left[a_1 \vee C^{a_1}_k\right] \wedge \cdots \wedge \left[a_n \vee C^{a_n}_k\right] \end{array} \quad ,$$

where $C^{a_1}_0 \equiv \cdots \equiv C^{a_n}_0 \equiv \mathsf{t}$ *and* $C^{a_1}_N \equiv \cdots \equiv C^{a_n}_N \equiv \mathsf{f}$.

For every proof Φ *of* A*, whose flow is*



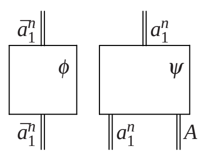
,

where only occurrences of the atoms $\bar{a}_1, \dots, \bar{a}_n$ *are mapped to edges in* ϕ*, there exists a cut-free proof* Ψ *of* A *whose size is bounded by a polynomial in* N*, the size of* Φ *and, for* $1 \leq k \leq N$*, the size of* Γ_k*.*

Proof. For every $1 \le i \le n$, let m_i (resp., m'_i) be the number of interactions (resp., cuts) where $\bar{a}_i^\phi$ appear in Φ, and consider the derivation

$$\Phi' \;=\; \begin{array}{c} \bigwedge_{0\le j\le m_1} \left[a_1^\psi \vee \bar{a}_1^\phi\right] \wedge \cdots \wedge \bigwedge_{0\le j\le m_n} \left[a_n^\psi \vee \bar{a}_n^\phi\right] \\ \| \\ A \vee \bigvee_{0\le j\le m'_1} \left(a_1^\psi \wedge \bar{a}_1^\phi\right) \vee \cdots \vee \bigvee_{0\le j\le m'_n} \left(a_n^\psi \wedge \bar{a}_n^\phi\right) \end{array} ,$$

with atomic flow



,

which exists by Lemma 2. Then, for $0 \le k \le N$, construct the following derivation:

$$\Phi_k \;=\; \begin{array}{c} \left(\begin{array}{c} a_1 \vee C_k^{a_1} \\ \|\{\mathsf{c}\uparrow,\mathsf{w}\uparrow\} \\ \bigwedge_{0\le j\le m_1} a_1 \vee C_k^{a_1} \end{array} \;\wedge\; \cdots \;\wedge\; \begin{array}{c} a_n \vee C_k^{a_n} \\ \|\{\mathsf{c}\uparrow,\mathsf{w}\uparrow\} \\ \bigwedge_{0\le j\le m_n} a_n \vee C_k^{a_n} \end{array} \right) \\ \Phi'\{\bar{a}_1^\phi / C_k^{a_1}, \ldots, \bar{a}_n^\phi / C_k^{a_n}\} \Big\| \mathsf{SKS}\setminus\{\mathsf{ai}\uparrow\} \\ \left[A \;\vee\; \begin{array}{c} \bigvee_{0\le j\le m'_1} a_1 \wedge C_k^{a_1} \\ \|\{\mathsf{c}\downarrow,\mathsf{w}\downarrow\} \\ a_1 \wedge C_k^{a_1} \end{array} \;\vee\; \cdots \;\vee\; \begin{array}{c} \bigvee_{0\le j\le m'_n} a_n \wedge C_k^{a_n} \\ \|\{\mathsf{c}\downarrow,\mathsf{w}\downarrow\} \\ a_n \wedge C_k^{a_n} \end{array} \right] \end{array} ,$$

which exists, and whose size is bounded by a polynomial in the size of Φ and the size of Γ_k, by Lemma 3. We then construct the cut-free derivation Ψ as follows:

$$
\begin{array}{c}
\left(\left[\dfrac{\mathsf{f}}{a_1} \vee \mathsf{t}\right] \wedge \cdots \wedge \left[\dfrac{\mathsf{f}}{a_n} \vee \mathsf{t}\right]\right) \\
\Phi_0 \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\uparrow\} \\
= \dfrac{A \vee \left(a_1 \wedge C_0^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_0^{a_n}\right)}{A \quad \vee \quad \begin{array}{c} \left(a_1 \wedge C_0^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_0^{a_n}\right) \\ \Gamma_1 \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\downarrow,\mathsf{ai}\uparrow\} \\ \left[a_1 \vee C_1^{a_1}\right] \wedge \cdots \wedge \left[a_n \vee C_1^{a_n}\right] \\ \Phi_1 \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\uparrow\} \\ = \dfrac{A \vee \left(a_1 \wedge C_1^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_1^{a_n}\right)}{\vdots} \end{array}} \\
= \dfrac{\vdots}{\dfrac{A \vee A}{A} \quad \vee \quad \begin{array}{c} \left(a_1 \wedge C_{N-2}^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_{N-2}^{a_n}\right) \\ \Gamma_{N-1} \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\downarrow,\mathsf{ai}\uparrow\} \\ \left[a_1 \vee C_{N-1}^{a_1}\right] \wedge \cdots \wedge \left[a_n \vee C_{N-1}^{a_n}\right] \\ \Phi_{N-1} \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\uparrow\} \end{array}} \\
= \dfrac{A \vee \left(a_1 \wedge C_{N-1}^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_{N-1}^{a_n}\right)}{\dfrac{\dfrac{A \vee A}{A} \quad \vee \quad \begin{array}{c} \left(a_1 \wedge C_{N-1}^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_{N-1}^{a_n}\right) \\ \Gamma_N \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\downarrow,\mathsf{ai}\uparrow\} \\ \left[a_1 \vee C_N^{a_1}\right] \wedge \cdots \wedge \left[a_n \vee C_N^{a_n}\right] \\ \Phi_N \Vert \mathsf{SKS}\setminus\{\mathsf{ai}\uparrow\} \\ \left[A \vee \left(\dfrac{a_1}{\mathsf{t}} \wedge \mathsf{f}\right) \vee \cdots \vee \left(\dfrac{a_n}{\mathsf{t}} \wedge \mathsf{f}\right)\right] \end{array}}{A}}
\end{array} .
$$

□

It can be shown that if we fix $N = n+1$ in Lemma 11, the formulae $C_k^{a_i}$ must be logically equivalent to $\theta_k^{n-1}(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$.

Theorem 12. *Given a proof Φ of A, there exists a cut-free proof Ψ of A, whose size is quasipolynomially bounded by the size of Φ.*

Proof. The result follows by Theorem 5, Theorem 10 and Lemma 11. □

7 Restoring the Symmetry

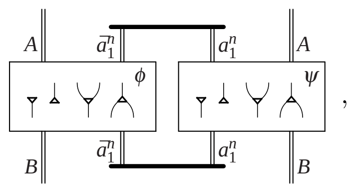
In this section we will no longer restrict ourselves to proofs, but consider all SKS-derivations. It turns out that the results can be generalised and all the proofs are analogous to the ones we have seen so far. First, we define a symmetric generalisation of cut-freeness. Rather than considering proofs without cuts, we consider derivations where there is no path from an axiom to a cut.

Definition 13. *A derivation is* weakly streamlined *if its atomic flow contains no path from an axiom to a cut. A derivation is* positively-weakly streamlined *(resp.,* negatively-weakly streamlined*) if its atomic flow contains no path from an axiom to a cut whose edges are mapped to positive (resp., negative) atom occurrences.*

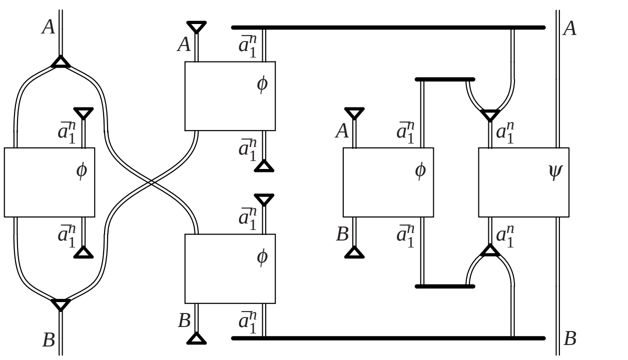
We will now show how our previous results can be generalised to obtain weakly streamlined derivations. For more details and examples of weakly streamlined derivations, see [GGS10]. Procedures to produce other normal forms from weakly streamlined derivations see [GG08,GGS10]. The most important normal form presented in these other papers is *streamlining* where there is no path from a weakening or an axiom to a coweakening or a cut, it is also shown that obtaining a streamlined derivation from a weakly streamlined one is trivial, *i.e.*, the procedure is confluent, strongly normalising and terminates in linear time.

The following is a generalisation of Theorem 5, whose proof is analogous.

Theorem 14. *Given a derivation Φ from A to B, with flow*



there exists a derivation Ψ from A to B, with flow



such that the size of Ψ is bounded by a polynomial in the size of Φ. Furthermore, if Φ is positively (resp., negatively) weakly streamlined, then so is Ψ.

Proof. We sketch the proof, by analogy to the proof of Theorem 5. Rather than making two copies of ϕ, we make four, intuitively this is done by first applying the same construction as in the asymmetric case, followed by applying the dual construction.

Every atom instance $\bar{a}$ that is mapped to from an edge in ϕ is replaced by four copies of itself: $[\bar{a} \vee \bar{a}] \wedge [\bar{a} \vee \bar{a}]$. The derivation Ψ is then created by replacing every atom instance $\bar{a}$ in the premiss (resp., conclusion) that is mapped to from an edge in ϕ by the derivation

$$\frac{\bar{a}}{\left[\bar{a} \vee \frac{\mathsf{f}}{\bar{a}}\right] \wedge \left[\bar{a} \vee \frac{\mathsf{f}}{\bar{a}}\right]} \;, \qquad \left(\text{resp., } \frac{\bar{a} \vee \bar{a}}{\bar{a}} \wedge \left[\frac{\bar{a}}{\mathsf{t}} \vee \frac{\bar{a}}{\mathsf{t}}\right]\right) \;,$$

and every interaction (resp., cut) that contains an atom instance $\bar{a}$ that is mapped to from an edge in ϕ by

$$\mathsf{s}\,\frac{\dfrac{\mathsf{t}}{\left(\bar{a} \wedge \frac{\mathsf{t}}{\bar{a} \vee a}\right) \vee a}}{\left(\left[\bar{a} \vee \frac{\mathsf{f}}{\bar{a}}\right] \wedge \left[\bar{a} \vee \frac{\mathsf{f}}{\bar{a}}\right]\right) \vee \frac{a \vee a}{a}} \;, \qquad \left(\text{resp., } \frac{\mathsf{s}\,\dfrac{\left(\left[\frac{\bar{a}}{\mathsf{t}} \vee \frac{\bar{a}}{\mathsf{t}}\right] \wedge [\bar{a} \vee \bar{a}]\right) \wedge \frac{a}{a \wedge a}}{\left[\bar{a} \vee \frac{\bar{a} \wedge a}{\mathsf{f}}\right] \wedge a}}{\mathsf{f}}\right) .$$

□

The following is a generalisation of Lemma 11, whose proof is analogous.

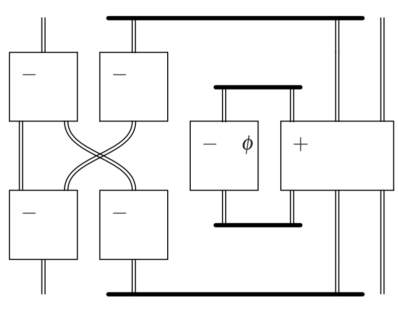
Lemma 15. *Let*

1. $N > 0$ *be an integer;*
2. $a_1, \ldots, a_n$ *be distinct atoms, where* $n = 2^m$ *for some* $m \geq 0$*;*
3. *there be, for every* $0 < k < N$ *and* $1 \leq i \leq n$*, a formula* $C_k^{a_i}$*;*
4. *there be, for every* $1 \leq k \leq N$*, a derivation*

$$\Gamma_k \;=\; \begin{array}{c} \left(a_1 \wedge C_{k-1}^{a_1}\right) \vee \cdots \vee \left(a_n \wedge C_{k-1}^{a_n}\right) \\ \Big\| \, \mathsf{SKS} \setminus \{\mathsf{ai}\!\downarrow, \mathsf{ai}\!\uparrow\} \\ \left[a_1 \vee C_k^{a_1}\right] \wedge \cdots \wedge \left[a_n \vee C_k^{a_n}\right] \end{array} \;,$$

where $C_0^{a_1} \equiv \cdots \equiv C_0^{a_n} \equiv \mathsf{t}$ *and* $C_N^{a_1} \equiv \cdots \equiv C_N^{a_n} \equiv \mathsf{f}$.

For every derivation Φ *from* A *to* B*, whose flow is*



,

where only occurrences of the atoms $\bar{a}_1, \ldots, \bar{a}_n$ *are mapped to edges in* ϕ*, there exists a negatively-weakly streamlined (resp., positively-weaklystreamlined) derivation* Ψ *from* A *to* B*, which is positively-weakly streamlined (resp., negatively-weakly streamlined) if* Φ *is, and whose size is bounded by a polynomial in* N*, the size of* Φ *and, for* $1 \leq k \leq N$*, the size of* Γ_k*.*

Proof. We sketch the proof, by analogy to the proof of Lemma 11.

For $0 \leq k \leq N$, construct the derivation

$$\Phi_k \;=\; \begin{array}{c} \left[a_1 \vee C^{a_1}_{k+1}\right] \wedge \cdots \wedge \left[a_n \vee C^{a_n}_{k+1}\right] \wedge A \\ \| \mathsf{SKS} \\ B \vee \left(a_1 \wedge C^{a_1}_{k}\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_{k}\right) \end{array} \quad ,$$

as in the asymmetric case. We then build the negatively weakly streamlined derivation Ψ as follows:

$$\begin{array}{c}
A \\ \hline
\left(\begin{array}{c} \left(\left[\dfrac{\mathsf{f}}{a_1} \vee \mathsf{t}\right] \wedge \cdots \wedge \left[\dfrac{\mathsf{f}}{a_n} \vee \mathsf{t}\right] \wedge A\right) \\ \Phi_0 \| \mathsf{SKS} \\ B \vee \left(a_1 \wedge C^{a_1}_0\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_0\right) \end{array} \right) \wedge \dfrac{A}{A \wedge A} \\ \hline
\mathsf{s}\;\; \left[B \vee \begin{array}{c} \left(\begin{array}{c} \left(a_1 \wedge C^{a_1}_0\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_0\right) \\ \Gamma_1 \| \mathsf{SKS} \setminus \{\mathsf{ai}\!\downarrow, \mathsf{ai}\!\uparrow\} \\ \left[a_1 \vee C^{a_1}_1\right] \wedge \cdots \wedge \left[a_n \vee C^{a_n}_1\right] \end{array} \wedge A \right) \\ \Phi_1 \| \mathsf{SKS} \\ B \vee \left(a_1 \wedge C^{a_1}_1\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_1\right) \end{array} \right] \wedge \dfrac{A}{A \wedge A} \\ \hline
\mathsf{s} \\
\vdots \\ \hline
\mathsf{s}\;\; \left[\dfrac{B \vee B}{B} \vee \begin{array}{c} \left(\begin{array}{c} \left(a_1 \wedge C^{a_1}_{N-2}\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_{N-2}\right) \\ \Gamma_{N-1} \| \mathsf{SKS} \setminus \{\mathsf{ai}\!\downarrow, \mathsf{ai}\!\uparrow\} \\ \left[a_1 \vee C^{a_1}_{N-1}\right] \wedge \cdots \wedge \left[a_n \vee C^{a_n}_{N-1}\right] \end{array} \wedge A \right) \\ \Phi_{N-1} \| \mathsf{SKS} \\ B \vee \left(a_1 \wedge C^{a_1}_{N-1}\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_{N-1}\right) \end{array} \right] \wedge A \\ \hline
\mathsf{s}\;\; \dfrac{B \vee B}{B} \vee \begin{array}{c} \left(\begin{array}{c} \left(a_1 \wedge C^{a_1}_{N-1}\right) \vee \cdots \vee \left(a_n \wedge C^{a_n}_{N-1}\right) \\ \Gamma_{N} \| \mathsf{SKS} \setminus \{\mathsf{ai}\!\downarrow, \mathsf{ai}\!\uparrow\} \\ \left[a_1 \vee C^{a_1}_{N}\right] \wedge \cdots \wedge \left[a_n \vee C^{a_n}_{N}\right] \end{array} \wedge A \right) \\ \Phi_{N} \| \mathsf{SKS} \\ \left[B \vee \left(\dfrac{a_1}{\mathsf{t}} \wedge \mathsf{f}\right) \vee \cdots \vee \left(\dfrac{a_n}{\mathsf{t}} \wedge \mathsf{f}\right) \right] \end{array} \\ \hline
B
\end{array} \quad .$$

Finally, we verify that Ψ does not contain any paths along edges mapping to negative atoms from an interaction to a cut, and the only way Ψ can contain a path along edges mapping to positive atoms is if Φ did. □

Theorem 16. *Given a derivation Φ from A to B, there exists a weakly streamlined derivation Ψ from A to B, whose size is bounded by a quasipolynomial in the size of Φ.*

Proof. The result follows by Theorem 10 and two applications of Theorem 14 and Lemma 15. □

8 Relations to the monotone sequent calculus

The results of this paper are, as mentioned previously, closely related to results about the monotone sequent calculus, through the translation between SKS and LK given in [Brü06]. In the following we will compare the different results by considering proofs in LK and derivations in SKS modulo this translation.

Atserias, Galesi and Pudlák, show in [AGP02] that MLK can quasipolynomialy simulate LK over monotone sequents, and as shown by Jeřábek in [Jeř08], this implies Theorem 12 above.

Furthermore, in [Jeř10], Jeřábek considers a conservative extension of MLK, called MCLK, and shows how MCLK can quasipolynomially simulate LK over arbitrary formulae. MCLK is defined by restricting LK to only allow cuts on monotone formulae.

Like MCLK, streamlined derivations are also a conservative extension of MLK and the two notions are very similar. The main difference is that the set of streamlined derivations is closed under replacing atoms by their negations, and MCLK is not.

We can show that MCLK does not contain all the streamlined derivations (consider any LK proof containing a cut where all the formulae are negative, but no negation rule appears) and vice versa (consider any SKS derivation where there is a positive path from an interaction to a cut rule). However, by combining the results from [Jeř10] with the results of this paper we can transform any proof to a streamlined MCLK proof in quasipolynomial time. This is so because an MCLK proof is also negatively weakly streamlined, and when Theorem 14 and Lemma 15 are applied to an MCLK proof the resulting proof is also in MCLK.

9 Final Comments

The quasipolynomial cut-elimination procedure makes use of the cocontraction rule. But the cocontraction rule can also be eliminated. A natural question is whether one can extend the quasipolynomial cut elimination to a cocontraction elimination or to say it in another way, whether one can eliminate cuts in quasipolynomial time without the help of cocontractions. This is probably an important question because all indications we have point to an essential role being played by cocontraction in keeping the complexity low. Cocontraction has something to do with sharing, it seems to provide a typical 'dag-like' speed-up over the corresponding 'tree-like' expansion.

The role played by cocontractions is the most immediate explanation of why quasipolynomial cut elimination works in Deep Inference and not, at the present stage, in the sequent calculus (for full propositional logic). The reason seems to be that exploiting cocontraction in the absence of cut is an intrinsic feature of deep inference, not achievable in Gentzen theory because of the lack of a top-down symmetry therein.

Another natural question is whether quasipolynomial time is the best we can do: there is no obvious objection to the existence of a polynomial cut-elimination procedure. It is possible to express threshold functions with polynomial formulae, but the hardest problem seems to be to obtain corresponding derivations of polynomial length. Deep inference flexibility in constructing derivations might help here.

The cut-elimination procedure presented here is peculiar because it achieves its result by using an external scheme, constituted by the threshold functions and the corresponding derivations, which does not depend on the particular derivation we are working on. It is as if the threshold construction was a catalyst that shortens the cut elimination. It would be interesting to interpret this phenomenon computationally, in some sort of Curry-Howard correspondence, where the threshold construction implements a clever operator. We intend to explore this path in the near future.

This leads to the wider question of a computational interpretation of deep inference. Atomic flows are a weak computational trace, which takes only the structural rules into account. It is surprising that such a trace, which forgets all the information given by the logical rules, is powerful enough to drive the cut-elimination procedure. We intend to carefully study its computational power and to see whether one can construct on this ground an original computational interpretation of proofs.

References

[AGP02] Albert Atserias, Nicola Galesi, and Pavel Pudlák. Monotone simulations of non-monotone proofs. *Journal of Computer and System Sciences*, 65(4):626–638, 2002.

[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/PrComplDI.pdf.

[BGGP10] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A quasi-polynomial cut-elimination procedure in deep inference via atomic flows and threshold formulae. In Edmund M. Clarke and Andrei Voronkov, editors, *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.

[Brü04] Kai Brünnler. *Deep Inference and Symmetry in Classical Proofs*. Logos Verlag, Berlin, 2004. http://www.iam.unibe.ch/ kai/Papers/phd.pdf.

[Brü06] Kai Brünnler. Deep inference and its normal form of derivations. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *Computability in Europe 2006*, volume 3988 of *Lecture Notes in Computer Science*, pages 65–74. Springer-Verlag, July 2006. http://www.iam.unibe.ch/ kai/Papers/n.pdf.

[BT01] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 347–361. Springer-Verlag, 2001. http://www.iam.unibe.ch/ kai/Papers/lcl-lpar.pdf.

[Bus91] Samuel R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53(1):75–102, 1991.

[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://arxiv.org/pdf/0709.1205.

[GGP10] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A proof calculus which reduces syntactic bureaucracy. In Christopher Lynch, editor, *RTA*, volume 6 of *LIPIcs*, pages 135–150. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[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.

[Gug] Alessio Guglielmi. Deep inference and the calculus of structures. Web site at http://alessio.guglielmi.name/res/cos.

[Gug07] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007. http://cs.bath.ac.uk/ag/p/SystIntStr.pdf.

[Jeř08] Emil Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 2008. In press. http://www.math.cas.cz/ jerabek/papers/cos.pdf.

[Jeř10] Emil Jeřábek. Proofs with monotone cuts. Preprint, 2010.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons Ltd and B. G. Teubner, Stuttgart, 1987.