

QUASIPOLYNOMIAL NORMALISATION IN DEEP INFERENCE VIA ATOMIC FLOWS AND THRESHOLD FORMULAE

PAOLA BRUSCOLI, ALESSIO GUGLIELMI, TOM GUNDERSEN, AND MICHEL PARIGOT

ABSTRACT. Jeřábek showed that analytic propositional-logic deep-inference proofs can be constructed in quasipolynomial time from nonanalytic proofs. In this work, we improve on that as follows: 1) we significantly simplify the technique; 2) our normalisation procedure is direct, *i.e.*, it is internal to deep inference. The paper is self-contained, and provides a starting point and a good deal of information for tackling the problem of whether a polynomial-time normalisation procedure exists.

1. INTRODUCTION

Deep inference is a new methodology in proof theory, introduced in [Gug07] and subsequently developed to the point that all major logics can be expressed with analytic deep-inference proof systems (see [Gug] for a complete overview). Deep inference is more general than traditional Gentzen proof theory because proofs can be freely composed by the logical operators, instead of having a rigid formula-directed tree structure. This induces a new symmetry, which can be exploited for achieving locality of inference rules, and which is not generally achievable with Gentzen methods. Locality, in turn, makes it possible to use new methods, often with a geometric flavour, in the normalisation theory of proof systems. In this paper, these new methods find application also in proof complexity.

The standard proof system for propositional logic in deep inference is system SKS [BT01, Brü04], and it has an associated analytic fragment, which we call ‘analytic SKS’. The notion of analyticity in deep inference is more general than the one in Gentzen proof theory, but it can approximately be considered equivalent to cut-freeness, as in Gentzen theory. The normalisation methods for achieving analyticity have much in common with Gentzen’s ones, despite having to cope with a higher generality.

Recently, Jeřábek showed that analytic SKS proofs can be constructed in quasipolynomial time from (nonanalytic) SKS ones [Jeř09]. This is a very surprising result because received wisdom suggests that analyticity requires exponential-time normalisation, as is the case in Gentzen proof systems. Jeřábek obtained his result by relying on a construction over threshold functions by Atserias, Galesi and Pudlák, in the monotone sequent calculus [AGP02]. We note that the monotone sequent calculus specifies a weaker logic than propositional logic because negation is not freely applicable.

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 [Brü06b].

In this work, we demonstrate again Jeřábek’s result, still by adopting, essentially, the Atserias-Galesi-Pudlák technique, and we improve on that as follows:

Date: January 25, 2010.

Bruscoli and Guglielmi are 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*.

Gundersen is 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*.

Parigot is supported by *Project INFER—Theory and Application of Deep Inference* of the *Agence Nationale de la Recherche*.

- (1) we significantly simplify 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) our normalisation procedure is direct, *i.e.*, it is internal to system SKS.

In doing so, we provide further evidence of the usefulness of atomic flows in the study of normalisation.

As Atserias, Galesi and Pudlák argue, there is no apparent reason for this normalisation problem not to be polynomial. The difficulty in obtaining polynomiality resides in the specification of threshold functions and their properties by formulae and proofs. The advantage of performing this investigation in deep inference is that we are dealing here with full propositional logic, and the directness and the simplification of the threshold-function technique exposed in this paper give us hope of further progress.

We think that this paper provides a starting point and a good deal of information for tackling the problem of whether a polynomial-time normalisation procedure exists. We think that working in deep inference is convenient because of its flexibility, but we emphasise that a normalisation result does not necessarily depend on deep-inference peculiarities, and might be exportable to other proof-theoretic realms.

Two recent research threads involving SKS are particularly relevant to this work:

- The basic proof-complexity properties of SKS have been studied, in [BG09].
- System SKS enjoys normal forms, which include analytic proofs, that can be defined and obtained in a largely syntax-independent way, by resorting to special graphs associated with SKS derivations, called ‘atomic flows’ [GG08].

Contrary to Frege proof systems, deep-inference proof systems have a proof theory, in the sense of a normalisation theory with a notion of analyticity, like the Gentzen formalisms have. The work [BG09] shows that deep-inference proof systems are as powerful as Frege proof systems (including when augmented by extension and substitution); however, the analytic proof systems of deep inference are strictly more powerful than Gentzen analytic proof systems because they exhibit exponential speed-ups.

Atomic flows, which can be considered as specialised Buss flow graphs [Bus91], play a major role in designing and controlling the normalisation procedure presented in this paper. They contribute to the overall clarification of this highly technical matter, by reducing our dependency on syntax. Atomic flows fall in the previously mentioned category of geometry flavoured methods (together with proof nets [Gir87]). 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).

We think that it will be possible to extend this normalisation procedure to more general normal-form notions than analyticity, like the various notions of streamlining studied in [GG08]. Another future research direction will be trying to extend the results of this paper to modal logics, which already enjoy deep-inference presentations [Brü06c, HS05, SS05, Sto07].

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 three 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 crucial step: we define the cut-free form of proofs, starting from their simple form. Here, threshold formulae play a major role.

- (3) In Section 7, we show how to transform a cut-free proof into an analytic one. This step involves routine deep-inference/atomic-flow transformations, which moderately decrease the size of proofs.

Section 8 concludes the paper with some comments on future research directions.

2. PROPOSITIONAL LOGIC IN DEEP INFERENCE

In the range of the deep-inference methodology, we can define several formalisms, *i.e.* general prescriptions on how to design proof systems. For example, the sequent calculus and natural deduction are formalisms in Gentzen-style proof theory, where the structure of proofs is determined by the tree structure of the formulae they prove.

The first, and conceptually simplest, formalism that has been defined in deep inference is called the *calculus of structures*, or *CoS* [Gug07]. CoS is now well developed for classical [Brü03, Brü06a, Brü06d, BT01, Brü06b], intuitionistic [Tiu06a], linear [Str02, Str03b], modal [Brü06c, GT07, Sto07] and commutative/non-commutative logics [Gug07, Tiu06b, Str03a, Bru02, DG04, GS01, GS02, SG09, GS09, Kah06, Kah07].

The standard proof system of propositional logic in CoS is called SKS. The basic proof-complexity properties of SKS, and, as a consequence, of propositional logic in CoS, have been studied in [BG09]:

- SKS is polynomially equivalent to Frege proof systems.
- SKS can be extended with Tseitin's extension and substitution, and the proof systems so obtained are polynomially equivalent to Frege proof systems augmented by extension and substitution.
- Analytic SKS polynomially simulates analytic Gentzen proof systems, but the converse does not hold: in fact, Statman's tautologies admit polynomial proofs in analytic SKS but, as is well known, only exponential ones in analytic Gentzen [Sta78].

In this paper, we work in CoS and SKS, but we introduce a new notation for CoS. We do so to conveniently describe certain derivations related to threshold formulae, which would seem very cumbersome otherwise (we mainly have in mind Definition 20). In related work, we are defining a new formalism, currently dubbed *Formalism A*, which generalises CoS and formally allows for the new notation.

In this section, we quickly introduce all the necessary notions. The standard reference for SKS in CoS and its typical constructions is [Brü04]; an introduction to SKS with an emphasis on proof complexity is in [BG09].

Formulae, denoted by A, B, C and D are freely built from: *units*, f (false), t (true); *atoms*, denoted by a, b, c, d and e ; *disjunction* and *conjunction*, $[A \vee B]$ 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. On the set of atoms a (non-identical) involution $\bar{\cdot}$ is defined and called *negation*; a and \bar{a} are *dual* atoms. We denote *contexts*, *i.e.*, formulae with a hole, by $K\{ \}$ and $H\{ \}$; 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 $b \wedge [(a \wedge d) \vee c]$.

Note that negation is only defined for atoms, and this is not a limitation because, thanks to De Morgan laws, negation can always be 'pushed to' atoms. Also, note that there are no negative or positive atoms in an absolute sense; we can only say that if we arbitrarily consider \bar{a} positive, then a must be negative, for example.

A CoS (*inference*) rule ρ is an expression $\rho \frac{A}{B}$, where the formulae A and B are called *premiss* and *conclusion*, respectively; an inference rule *instance* $\rho \frac{C}{D}$, where C and D are

instances of A and B , respectively, generates an (*inference*) step $\rho \frac{K\{C\}}{K\{D\}}$, for each context $K\{\}$. A *derivation*, Φ , from A (*premiss*) to B (*conclusion*) is a chain of inference steps with A at the top and B at the bottom, and is usually indicated by

$$\begin{array}{c} A \\ \Phi \parallel^{\mathcal{S}} \\ B \end{array},$$

where \mathcal{S} is the name of the proof system or a set of inference rules (we might omit Φ and \mathcal{S}); a *proof*, often denoted by Π , is a derivation with premiss t ; besides Φ , we denote derivations with Ψ . Sometimes we group $n \geq 0$ inference steps of the same rule ρ together into one step, and we label the step with $n \cdot \rho$.

The *size* $|A|$ of a formula A , and the *size* $|\Phi|$ of a derivation Φ , is the number of unit and atom occurrences appearing in it.

By $A\{a_1/B_1, \dots, a_b/B_b\}$, we denote the operation of simultaneously substituting formulae B_1, \dots, B_b into all the occurrences of the atoms a_1, \dots, a_b in the formula A , respectively; note that the occurrences of $\bar{a}_1, \dots, \bar{a}_b$ are not automatically substituted. Often, we only substitute certain occurrences of atoms, and these are indicated with superscripts that establish a relation with atomic flows. As a matter of fact, we extend the notion of substitution to derivations in the natural way, but this requires a certain care. The issue is clarified in Section 3 (see, in particular, Notations 3 and 5 and Proposition 4).

System SKS is a CoS proof system, defined by the following *structural* inference rules:

$$\begin{array}{ccc} \text{ai}\downarrow \frac{t}{a \vee \bar{a}} & \text{aw}\downarrow \frac{f}{a} & \text{ac}\downarrow \frac{a \vee a}{a} \\ \textit{identity} & \textit{weakening} & \textit{contraction} \\ \\ \text{ai}\uparrow \frac{a \wedge \bar{a}}{f} & \text{aw}\uparrow \frac{a}{t} & \text{ac}\uparrow \frac{a}{a \wedge a} \\ \textit{cut} & \textit{coweakening} & \textit{cocontraction} \end{array},$$

and by the following two *logical* inference rules:

$$\begin{array}{cc} \frac{A \wedge [B \vee C]}{(A \wedge B) \vee C} & \frac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \\ \textit{switch} & \textit{medial} \end{array}.$$

In addition to these rules, there is a rule $=$, such that C and D are opposite sides in one of the following equations:

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

We do not always show the instances of rule $=$, and when we do show them, we gather several contiguous instances into one. We consider the $=$ rule as implicitly present in all systems. The first row in Figure 2 shows some SKS example derivations.

The equality relation $=$ on formulae is defined by closing the equations in (1) by reflexivity, symmetry, transitivity and by stipulating that $A = B$ implies $K\{A\} = K\{B\}$; to indicate literal equality of the formulae A and B we adopt the notation $A \equiv B$.

A *cut-free* derivation is a derivation where $\text{ai}\uparrow$ is not used, *i.e.*, a derivation in $\text{SKS} \setminus \{\text{ai}\uparrow\}$. Of special importance in this paper is the following proof system:

Definition 1. *Analytic* SKS is the system $\text{aSKS} = \text{SKS} \setminus \{\text{ai}\uparrow, \text{aw}\uparrow\}$.

The notion of analyticity in deep inference has similarities and differences with analyticity in Gentzen formalisms. The similarities mainly reside in the normalisation theory of deep inference, which has similar properties to the Gentzen theory ones. On the other hand, analyticity in Gentzen is based on the subformula property, which guarantees certain properties on the depth of formulae in derivations. Such properties do not hold in deep inference, unless we properly restrict it. These issues are discussed in [BG07, BG09, Jeř09].

Besides SKS, another standard deep-inference system is SKSg, which is the same as SKS, except that it does not contain medial and its structural rules are not restricted to atoms. In particular, we use in this paper the rules

$$w\downarrow \frac{f}{A} \quad , \quad w\uparrow \frac{A}{t} \quad , \quad c\downarrow \frac{A \vee A}{A} \quad \text{and} \quad c\uparrow \frac{A}{A \wedge A} \quad .$$

Clearly, a derivation in SKS is also a derivation in SKSg. It can easily be proved that SKS and all its fragments containing the logical and = rules polynomially simulate, respectively, SKSg and its corresponding fragments [BG09]. For example, $\{s, m, =, ac\downarrow\}$ polynomially simulates $\{s, =, c\downarrow\}$, and $aSKS = \{s, m, =, ai\downarrow, aw\downarrow, ac\downarrow, ac\uparrow\}$ polynomially simulates $\{s, =, i\downarrow, w\downarrow, c\downarrow, c\uparrow\}$ (where $i\downarrow$ is the nonatomic identity). This allows us to transfer properties from SKS to SKSg; in particular, the main result in this paper, *i.e.*, that SKS proofs can be transformed into analytic ones in quasipolynomial time, holds also for SKSg proofs. One reason to work with SKS instead of SKSg, as we do in this paper, is that atomicity of rules allows us to use atomic flows more conveniently.

A notable analytic system is $KS = \{s, m, =, ai\downarrow, aw\downarrow, ac\downarrow\}$, which is complete for propositional logic [BT01, Brü04]; this, of course, entails completeness for all the systems that contain KS, such as aSKS and SKS.

We can replace instances of nonatomic structural rules by derivations with the same premiss and conclusion, and that only contain atomic structural rules. The price to pay is a quadratic growth in size. This is stated by the following, routine proposition (keep in mind that, from now on, we consider the = rule as implicitly present in all systems). An example is the rightmost upper derivation in Figure 2, which stands for a nonatomic cocontraction.

Proposition 2. *Rule instances of $w\downarrow$, $w\uparrow$, $c\downarrow$ and $c\uparrow$ can be derived in quadratic time by derivations in $\{aw\downarrow\}$, $\{aw\uparrow\}$, $\{m, ac\downarrow\}$ and $\{m, ac\uparrow\}$, respectively.*

Sometimes, we use a nonatomic rule instance to stand for some derivation in SKS that derives that instance, as per Proposition 2.

For CoS proofs, we adopt a special notation that allows us considerable efficiency in describing derivations, especially in the crucial Definition 20. We denote the result of including every formula of Φ into the context $K\{ \}$ by $K\{\Phi\}$: since we adopt deep inference, $K\{\Phi\}$ is a valid derivation. Then, given the two derivations

$$\begin{array}{c} A \\ \Phi\parallel \\ B \end{array} \quad \text{and} \quad \begin{array}{c} C \\ \Psi\parallel \\ D \end{array} \quad ,$$

by $\Phi \vee \Psi$ and $\Phi \wedge \Psi$ we denote, respectively,

$$= \frac{\left[\begin{array}{c} A \\ \Phi\parallel \vee C \\ B \end{array} \right]}{\left[\begin{array}{c} C \\ B \vee \Psi\parallel \\ D \end{array} \right]} \quad \text{and} \quad = \frac{\left(\begin{array}{c} A \\ \Phi\parallel \wedge C \\ B \end{array} \right)}{\left(\begin{array}{c} C \\ B \wedge \Psi\parallel \\ D \end{array} \right)} \quad ,$$

or any other CoS derivations obtained by interleaving Φ and Ψ and respecting the specified logical relations between Φ and Ψ . We call this the *Formalism A* notation. Examples of Formalism A derivations are in the second row of Figure 2, in correspondence with CoS derivations in the first row. Note that we omit structural rule names in Formalism A notation (since they are easily inferable, this improves legibility).

The size of CoS derivations is, obviously, at most quadratic in the size of Formalism A derivations denoting them. We use this fact implicitly throughout the paper, and we always measure the CoS size of derivations, even if we show them in Formalism A notation. Because of its convenience, the Formalism A notation is currently being developed as a full-fledged deep-inference formalism.

3. ATOMIC FLOWS

Atomic flows, which have been introduced in [GG08], are, essentially, specialised Buss flow graphs [Bus91]. They are particular directed graphs associated with SKS derivations: every derivation yields one atomic flow obtained by tracing the atom occurrences in the derivation. Infinitely many derivations correspond to each atomic flow; this suggests that much of the information in a derivation is lost in its associated atomic flow; in particular, there is no information about instances of logical rules, only structural rules play a role. As shown in [GG08], it turns out that atomic flows contain sufficient structure to control normalisation procedures, providing in particular induction measures that can be used to ensure termination. Such normalisation procedures require exponential time on the size of the derivation to be normalised. In the present work, we improve the complexity of proof normalisation to quasipolynomial time, but an essential role is played by the complex logical relations of threshold formulae, which are external and independent from the given proof. This means that atomic flows are not sufficient to define the normalisation procedure; however, they still are a very convenient tool for defining and understanding several of its aspects.

We can single out three features of atomic flows that, in general, and not just in this work, help in designing normalisation procedures:

- (1) Atomic flows conveniently express the topological structure of atom occurrences in a proof. This is especially useful for defining the ‘simple form’ of proofs, in Definition 8.
- (2) Atomic flows provide for an efficient way to control substitutions for atom occurrences in derivations. This is especially useful for defining the ‘cut-free form’ of proofs, in Definition 24.
- (3) We can define graph rewriting systems over atomic flows that control normalisation procedures on derivations. This could be used for obtaining the ‘analytic form’ of proofs, as we do in Theorem 27.

Our aim now is to quickly and informally provide the necessary notions about atomic flows, especially concerning aspects (1) and (2) above. Although the feature (3) of atomic flows did help us in obtaining proofs in analytic form, we estimate that formally introducing the necessary machinery is unjustified in this paper. In fact, given our limited needs here, we can operate directly on derivations, without the intermediate support of atomic flows. Nonetheless, being aware of the underlying atomic-flow methods is useful for the reader who wishes to further investigate this matter. So, we informally provide, in Section 7, enough material to make the connection with the atomic-flow techniques that are fully developed in [GG08].

We obtain one atomic flow from each derivation by tracing all its atom occurrences and by keeping track of their creation and destruction (in identity/cut and weakening/oweakening instances), their duplication (in contraction/cocontraction instances) and their duality (in identity/cut instances). Technically, atomic flows are directed graphs

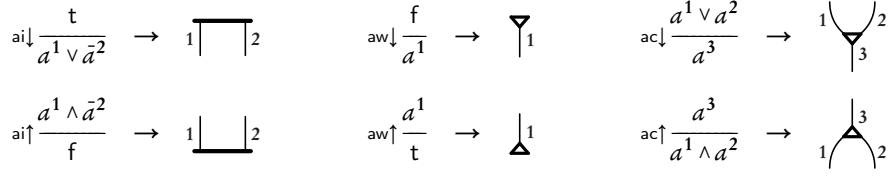


FIGURE 1. Vertices of atomic flows.

of a special kind, but it is more intuitive to consider them as diagrams generated by composing *elementary atomic flows* that belong to one of seven kinds.

The first kind of elementary atomic flow is the *edge*



which corresponds to one or more occurrences of the same atom in a given derivation, all of which are not active in any structural rule instance, *i.e.*, they are not the atom occurrences that instantiate a structural rule.

The other six kinds of elementary diagrams are associated with the six structural inference rules, as shown in Figure 1, and they are called *vertices*; each vertex has some incident edges. At the left of each arrow, we see an instance of a structural rule, where the atom occurrences are labelled by small numerals; at the right of the arrow, we see the vertex corresponding to the rule instance, whose incident edges are labelled in accord with the atom occurrences they correspond to. 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. Instead of small numerals, sometimes we use ϵ or ι or colour to label edges (as well as atom occurrences), but we do not always use labels.

All edges are directed, but we do not explicitly show the orientation. Instead, we consider it as implicitly given by the way we draw them, namely, edges are oriented along the vertical direction. So, the vertices corresponding to dual rules, in Figure 1, are distinct, for example, an identity vertex and a cut vertex are different because the orientation of their edges is different. On the other hand, the horizontal direction plays no role in distinguishing atomic flows; this corresponds to commutativity of logical relations.

We can define (*atomic flows*) as the smallest set of diagrams containing elementary atomic flows, and closed under the composition operation consisting in identifying zero or more edges such that no cycle is created. In addition, for a diagram to be an atomic flow, it must be possible to assign it a polarity, according to the following definition. A *polarity assignment* 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 .

Let us see some examples. The flow

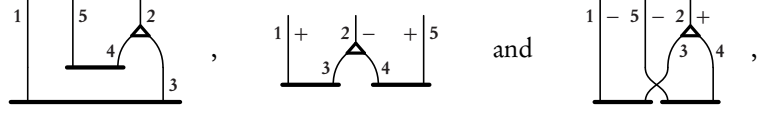


is obtained by juxtaposing (*i.e.*, composing by identifying zero edges):

- three edges,
- a flow obtained by composing a cut vertex with a cocontraction vertex, and
- a flow obtained by composing an identity vertex with a cut vertex.

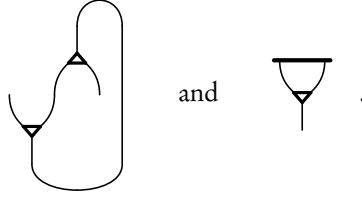
Note that there are no cycles in the flow, and that we can find 32 different polarity assignments, *i.e.*, two for each of the five connected components of the flow (this is a general rule).

Let us see another example. These are three different representations of the same flow:



where we label edges to show their correspondence. In the two rightmost flows, we indicate the two different polarity assignments that are possible.

The following two diagrams are not atomic flows:



The left one is not a flow because it contains a cycle, and the right one because there is no possible polarity assignment.

Let us see how to extract atomic flows from derivations. Given an SKS derivation Φ , we obtain, by the following prescriptions, a unique atomic flow ϕ , such that there is a surjective map between atom occurrences in Φ and edges of ϕ :

- Each structural inference step in Φ is associated with one and only one vertex in ϕ , such that active atom occurrences in the rule instance map to edges incident with the vertex. The correspondence is indicated in Figure 1. For example, the flow associated with the inference step at the left is indicated at the right:

$$\text{ac}\downarrow \frac{a^1 \wedge [b^2 \vee [a^3 \vee a^4]]}{a^1 \wedge [b^2 \vee a^5]} \quad \text{and} \quad \begin{array}{c} 1 \quad 2 \\ | \quad | \\ 3 \quad 4 \\ \vee \\ 5 \end{array} .$$

Note that the nonactive atoms are ‘traced’ by associating each trace with one edge; this corresponds well to abbreviating, say, the inference step $\text{ac}\downarrow \frac{K\{a \vee a\}}{K\{a\}}$ by $K\left\{\frac{a \vee a}{a}\right\}$.

- For each other inference step in Φ , all the atom occurrences in the premiss are respectively mapped to the same edges of ϕ as the atom occurrences in the conclusion. For example, the flow associated with the inference step

$$\text{m} \frac{a^1 \wedge [(b^2 \wedge c^3) \vee (d^4 \wedge e^5)]}{a^1 \wedge ([b^2 \vee d^4]) \wedge ([c^3 \vee e^5])} \quad \text{is} \quad \begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \\ | \quad | \quad | \quad | \quad | \end{array} .$$

The flow ϕ so obtained is called the atomic flow *associated with* the derivation Φ . We show three examples in Figure 2: in the top row we see three SKS derivations in the standard CoS syntax; in the row below, we show the same derivations in the Formalism A notation; in the bottom row, we see the three corresponding atomic flows.

Perhaps surprisingly, it can be proved that every flow is associated with infinitely many SKS derivations (see [GG08]).

We introduce now some graphical shortcuts. When certain details of a flow are not important, but only the vertex kinds and its upper and lower edges are, we can use boxes, labelled with all the vertex kinds that can appear in the flow they represent. For example, the following left and centre flows could represent the previously seen flow (2), whereas

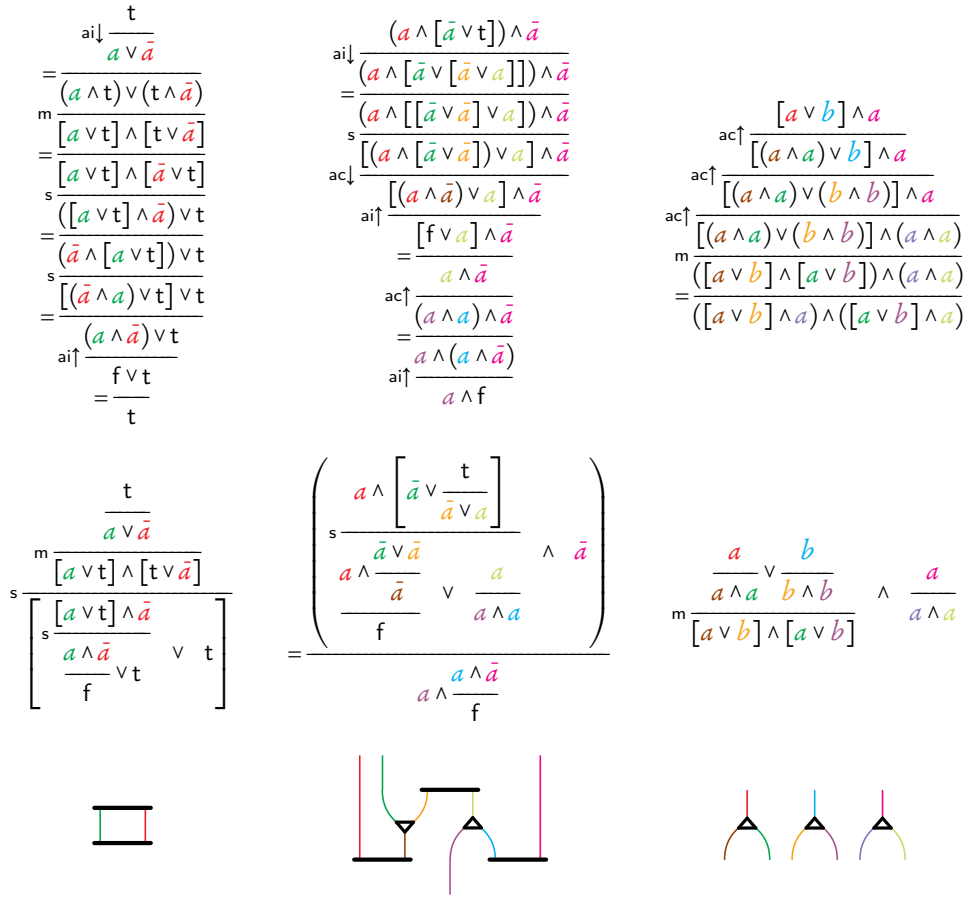
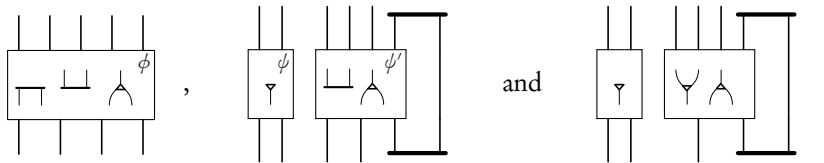


FIGURE 2. Examples of derivations in CoS and Formalism A notation, and associated atomic flows.

the right flow cannot:

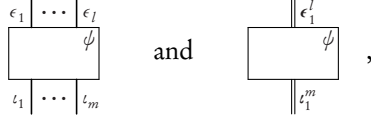


The flow at the right cannot represent flow (2) because it has the wrong number of lower edges and because a necessary cut vertex is not allowed by the labelling of the boxes. As just shown, we sometimes label boxes with the name of the flow they represent. For example, flow ϕ above could represent flow (2), and, if the centre flow stands for (2), then flows ψ and ψ' are, respectively,



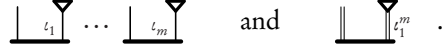
When no vertex labels appear on a box, we assume that the vertices in the corresponding flow can be any (so, it does not mean that there are no vertices in the flow).

We sometimes use a double line notation for representing multiple edges. For example, the following diagrams represent the same flow:

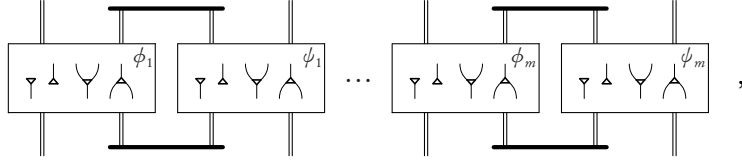


where $l, m \geq 0$; note that we use ϵ_1^l to denote the vector $(\epsilon_1, \dots, \epsilon_l)$. We might label multiple edges with one of the formulae that the associated atom occurrences form in a derivation.

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



We observe that the flow of every SKS derivation can always be represented as a collection of $m \geq 0$ connected components as follows:



such that each edge in flow ϕ_i is associated with some occurrence of some atom a_i , and each edge in flow ψ_i is associated with some occurrence of atom \bar{a}_i . Note that it might happen that for $i \neq j$ we have $a_i \equiv a_j$. If we do not insist on dealing with connected components, we can adopt the same representation as above and stipulate that $i \neq j$ implies $a_i \not\equiv a_j, \bar{a}_j$. This would mean that the derivation only contains occurrences of atoms a_1, \dots, a_m , such that these atoms and their dual are all mutually distinct.

Note that no matter how we assign a polarity, all the edges in ϕ_i and all those in ψ_i are respectively mapped to dual polarity values. Given a polarity assignment, we talk about *negative* and *positive* rule instances of (co)weakening and (co)contraction rules, according to whether the edges incident with the associated vertices map to $-$ or $+$, respectively.

In the following, when informally dealing with derivations, we freely transfer to them notions defined for their flows. For example, we can say that an atom occurrence is negative for a given polarity assignment (if the edge associated with the atom occurrence maps to $-$) or that two atom occurrences are connected (if the associated edges belong to the same connected component). In fact, one of the advantages of working with flows is that they provide us with convenient geometrical notions.

As we mention at the beginning of this section, atomic flows help in selectively substituting for atom occurrences. In fact, given a derivation and its associated flow, we can use edges and boxes to individuate atom occurrences in the derivation, and then possibly substitute for them. For example, let us suppose that we are given the following associated derivation and flow:

$$\Phi = \left[\begin{array}{c} (a \wedge f) \vee \left(a \wedge \frac{f}{\bar{a}} \right) \\ \frac{a \vee a}{a} \wedge \frac{f \vee \bar{a}}{\bar{a}} \\ \hline f \end{array} \right] \vee \bar{a} \quad \text{and} \quad \begin{array}{c} \vee \\ \hline \vee_1 \end{array} \Big| .$$

We can then distinguish between the three occurrences of \bar{a} that are mapped to edge 1 and the one that is not, as in

$$\Phi = \left[\begin{array}{c} (a \wedge f) \vee \left(a \wedge \frac{f}{\bar{a}^1} \right) \\ \frac{a \vee a}{a} \wedge = \frac{f \vee \bar{a}^1}{\bar{a}^1} \\ \frac{\quad}{f} \end{array} \vee \bar{a} \right] ;$$

we can also substitute for these occurrences, for example by $\{\bar{a}^1/f\}$; such a situation occurs in the proof of Theorem 12. Note that simply substituting f for \bar{a}^1 would invalidate this derivation because it would break the cut and weakening instances; however, the proof of Theorem 12 specifies how to fix such broken instances.

We generalise this labelling mechanism to boxes. For example, we can use a different representation of the flow of Φ to individuate two classes a^ϕ and \bar{a}^ϕ of atom occurrences, as follows:

$$\Phi = \left[\begin{array}{c} (a \wedge f) \vee \left(a \wedge \frac{f}{\bar{a}^\phi} \right) \\ \frac{a \vee a}{a^\phi} \wedge = \frac{f \vee \bar{a}^\phi}{\bar{a}^\phi} \\ \frac{\quad}{f} \end{array} \vee \bar{a}^\phi \right] \quad \text{and} \quad \begin{array}{c} \text{Y-shape} \\ \text{Box } \phi \\ \text{Bottom bar} \end{array} .$$

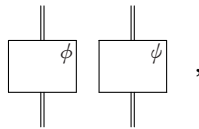
In order to define the notion of cut-free form (Definition 24), we need the following proposition, which we state here because it constitutes a good exercise about atomic flows. Note that, in the following, we use several boxes labelled by ϕ : this means that we are dealing with several copies of the same flow ϕ .

Notation 3. Given a formula A in a derivation whose associated atomic flow contains a flow ϕ , we indicate with a^ϕ every occurrence of the atom a in A whose associated edge is in ϕ . So, as in the following Proposition 4, $A\{a^\phi/B, \bar{a}^\phi/\bar{B}\}$ stands for the formula A where the atom occurrences of a and its dual, whose associated edges are in ϕ , are substituted with formula B and its dual, respectively.

Proposition 4. *Given a derivation*

$$\frac{A}{\Phi \parallel_{\text{SKS}} A'}$$

let its associated flow have shape



such that ϕ is a connected component each of whose edges is associated with atom a or \bar{a} ; then, for any formula B , there exists a derivation

$$\frac{A\{a^\phi/B, \bar{a}^\phi/\bar{B}\}}{\Psi \parallel_{\text{SKS}} A'\{a^\phi/B, \bar{a}^\phi/\bar{B}\}}$$

doing this would invalidate identity and cut instances, but we actually only need the simple core of the proof.

Our normalisation procedure essentially relies on gluing together simple cores, where we substitute the a_i atom occurrences that map to edges in ϕ_i with certain formulae called ‘pseudocomplements’ (see Section 5 and Definition 24).

Remark 9. A proof in simple form over a_1^0 is cut-free.

In order to prove Theorem 12, we need two facts, Proposition 10 and Lemma 11.

In the following (routine) proposition, we use the switch rule s to ‘push outside’ or ‘pull inside’ a formula A , relative to a context $K\{ \}$.

Proposition 10. *For any context $K\{ \}$ and formula A , there exist derivations whose size is less than $|K\{A\}|^2$ and have shape*

$$\begin{array}{c} K\{A\} \\ \parallel_{\{s\}} \\ A \vee K\{f\} \end{array} \quad \text{and} \quad \begin{array}{c} A \wedge K\{t\} \\ \parallel_{\{s\}} \\ K\{A\} \end{array} .$$

Proof. We only build the derivation at the left in the claim, the construction being dual for the one at the right. We reason by induction on the number n of \vee - \wedge alternations in the formula-tree branch of $\{ \}$ in $K\{ \}$. If $n = 0$, then $K\{A\} = A \vee K\{f\}$. If $n > 0$, consider

$$\left[\begin{array}{c} \left(\begin{array}{c} H\{A\} \\ \parallel_{\{s\}} \\ A \vee H\{f\} \end{array} \wedge B \right) \vee C \\ \text{\scriptsize } s \frac{}{A \vee (H\{f\} \wedge B)} \end{array} \right] ,$$

for some context $H\{ \}$ and formulae B and C , such that $K\{ \} = (H\{ \} \wedge B) \vee C$ and the number of \vee - \wedge alternations in the formula-tree branch of $\{ \}$ in $H\{ \}$ is $n - 1$. The number of s instances is n , and we have that $n \leq |K\{f\}|$. \square

Note that the atomic flows of the derivations in the previous proposition only consist of edges because no structural rules appear.

To prove Theorem 12, we could now proceed as follows. Given a proof, we assign it (and its flow) an arbitrary polarity, under certain assumptions that we can always easily satisfy. We then focus on the negative paths connecting identity and cut vertices. If cocontraction vertices lie along these paths, we have a potential problem because some atoms in the conclusion of the proof might be connected to atoms in some identity instances. This would prevent us from substituting pseudocomplements, as previously mentioned, because by doing so we would alter the conclusion of the proof.

However, we can solve the problem by replacing each cocontraction vertex by an appropriate flow involving identity, cut and contraction vertices, in such a way that the only contraction vertex so introduced is positive. Actually, the lemma below takes a more radical approach, which simplifies exposition and also has broader application: we replace all negative contraction and cocontraction instances. This unnecessarily bloats the proof, but still stays well inside polynomial bounds.

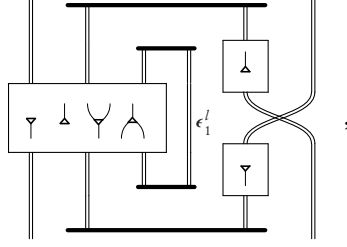
Lemma 11. *Given any derivation*

$$\begin{array}{c} A \\ \Phi \parallel_{\text{SKS}} \\ B \end{array} ,$$

we can, in linear time in the size of Φ , construct a derivation

$$\begin{array}{c} A \\ \parallel_{\text{SKS}} \\ B \end{array}$$

such that its atomic flow has shape

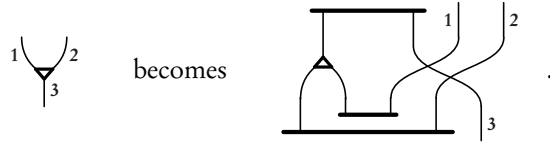


and such that no two atoms associated with $\epsilon_1, \dots, \epsilon_l$ are dual, for some $l \geq 0$.

Proof. Assign a polarity to the flow of Φ such that no two dual atoms are both associated with negative edges; then replace each negative contraction instance as follows:

$$\begin{array}{c} A \\ \Psi \parallel_{\text{SKS}} \\ K \left\{ \begin{array}{c} \bar{a} \vee \bar{a} \\ \bar{a} \end{array} \right\} \\ \Psi' \parallel_{\text{SKS}} \\ B \end{array} \quad \text{becomes} \quad K \left\{ \begin{array}{c} \frac{t}{a \vee \bar{a}} \wedge [\bar{a} \vee \bar{a}] \\ \frac{a}{a \wedge a} \wedge [\bar{a} \vee \bar{a}] \\ \frac{s}{a \wedge [(a \wedge \bar{a}) \vee \bar{a}]} \vee \bar{a} \\ \frac{s}{\frac{a \wedge \bar{a}}{f} \vee \frac{a \wedge \bar{a}}{f}} \\ \Psi' \parallel_{\text{SKS}} \\ B \end{array} \right\} .$$

This corresponds, in the flow, to replacing each negative contraction vertex as follows:



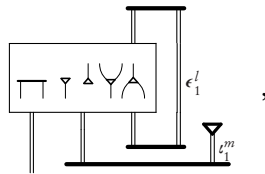
Proceed analogously with negative cocontraction instances. \square

We are now ready to prove the main result of this section.

Theorem 12. *Given any proof Π of A in SKS, we can, in cubic time in the size of Π , construct a proof of A in simple form.*

Proof. We proceed in three steps.

- (1) By Lemma 11, we can transform Π , in linear time in its size, into a proof $\Pi' \parallel_{\text{SKS}}^A$, whose flow has shape

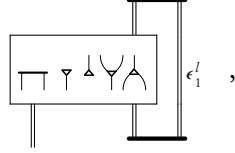


where $l, m \geq 0$ and such that no two atoms associated with $\epsilon_1, \dots, \epsilon_l$ are dual. For $1 \leq i \leq m$, we successively transform Π' as follows, for some $\Pi'', \Phi, \Phi', K \{ \}$

and $H\{ \}$ }:

$$\begin{array}{ccc}
 \begin{array}{c} \Pi'' \parallel \\ K \left\{ \frac{f}{\bar{a}^{l_i}} \right\} \\ \Phi \parallel \\ H \left\{ \frac{a \wedge \bar{a}^{l_i}}{f} \right\} \\ \Phi' \parallel \\ A \end{array} & \text{becomes} & \begin{array}{c} \Pi'' \parallel \\ K \{f\} \\ \Phi \{ \bar{a}^{l_i} / f \} \parallel \\ H \left\{ \frac{a}{t} \wedge f \right\} \\ \Phi' \parallel \\ A \end{array} .
 \end{array}$$

This way, we obtain, in linear time, a proof $\Pi''' \parallel_A^{\text{SKS}}$, whose flow is



and whose size is smaller than $|\Pi'|$.

- (2) Thanks to Proposition 10, for $1 \leq i \leq l$, we successively transform Π''' as follows, for some Ψ , Ψ' and $K'\{ \}$ }:

$$\begin{array}{ccc}
 \begin{array}{c} \Psi \parallel \\ K' \left\{ \frac{t}{a \vee \bar{a}^{\epsilon_i}} \right\} \\ \Psi' \parallel \\ A \end{array} & \text{becomes} & \begin{array}{c} \frac{t}{a \vee \bar{a}^{\epsilon_i}} \\ [a \vee \bar{a}] \wedge \Psi \parallel \\ [a \vee \bar{a}^{\epsilon_i}] \wedge K' \{t\} \\ \parallel \{s\} \\ K' \{a \vee \bar{a}^{\epsilon_i}\} \\ \Psi' \parallel \\ A \end{array} ;
 \end{array}$$

we also apply the dual transformation for each $a_i \uparrow$ instance. This way, we obtain a proof

$$\begin{array}{c}
 \frac{t}{a_1 \vee \bar{a}_1^{\epsilon_1}} \wedge \dots \wedge \frac{t}{a_l \vee \bar{a}_l^{\epsilon_l}} \\
 \Psi'' \parallel \\
 A \vee \frac{a_1 \wedge \bar{a}_1^{\epsilon_1}}{f} \vee \dots \vee \frac{a_l \wedge \bar{a}_l^{\epsilon_l}}{f}
 \end{array} ,$$

whose flow is the same as that of Π''' because each transformation conserves the flow. If $|\Pi'''| = n$, and given that $n > 2l$, the size of each derivation introduced by virtue of Proposition 10 is at most $4n^2$. So, each of the $2l$ transformations increases the size of the proof by $O(n^2)$, which makes for a total complexity of $O(n^3)$.

- (3) Consider \mathbf{b}_1^n such that b_1, \dots, b_n are distinct and $\{a_1, \dots, a_l\} = \{b_1, \dots, b_n\}$. We can build, in linear time, the proof

$$\left[A \vee \begin{array}{c} \frac{t}{b_1 \vee \bar{b}_1} \wedge \dots \wedge \frac{t}{b_n \vee \bar{b}_n} \\ \parallel_{\{\epsilon\uparrow\}} \\ [a_1 \vee \bar{a}_1^{\epsilon_1}] \wedge \dots \wedge [a_l \vee \bar{a}_l^{\epsilon_l}] \\ \parallel_{\Psi''} \\ (a_1 \wedge \bar{a}_1^{\epsilon_1}) \vee \dots \vee (a_l \wedge \bar{a}_l^{\epsilon_l}) \\ \parallel_{\{\epsilon\downarrow\}} \\ \left[\frac{b_1 \wedge \bar{b}_1}{f} \vee \dots \vee \frac{b_n \wedge \bar{b}_n}{f} \right] \end{array} \right],$$

which is in simple form over \mathbf{b}_1^n . We can then obtain a proof in SKS in time $O(n^2)$, because of Proposition 2. \square

The transformation in Step (1) in the previous proof is a case of ‘weakening reduction’ for atomic flows, studied in [GG08]. In Section 7 we comment more on this.

Remark 13. In general, given a proof Π and by the construction in the proof of Theorem 12, we can obtain several different simple forms from Π . In fact, apart from permutations of rule instances, commutativity and associativity, the simple forms depend on the choice of a polarity assignment (Lemma 11).

5. THRESHOLD FORMULAE

We present here the main construction of this paper, *i.e.*, a class of derivations Γ that only depend on a given set of atoms and that allow us to normalise any proof containing those atoms. The complexity of the Γ derivations dominates the complexity of the normal proof, and is due to the complexity of certain ‘threshold formulae’, on which the Γ derivations are based. The Γ derivations are constructed in Definition 20; this directly leads to Theorem 22, which states a crucial property of the Γ derivations and which is the main result of this section.

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).

In the following, $\lfloor x \rfloor$ denotes the maximum integer n such that $n \leq x$.

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 22. Efficiently obtaining the property stated in Theorem 22 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].

Definition 14. Consider $n > 0$, distinct atoms a_1, \dots, a_n , and let $p = \lfloor n/2 \rfloor$ and $q = n - p$; for $k \geq 0$, we define the *threshold formulae* $\theta_k^n \mathbf{a}_1^n$ as follows:

- for any $n > 0$ let $\theta_0^n \mathbf{a}_1^n \equiv t$;
- for any $n > 0$ and $k > n$ let $\theta_k^n \mathbf{a}_1^n \equiv f$;
- $\theta_1^1(a_1) \equiv a_1$;
- for any $n > 1$ and $0 < k \leq n$ let $\theta_k^n \mathbf{a}_1^n \equiv \bigvee_{\substack{i+j=k \\ 0 \leq i \leq p \\ 0 \leq j \leq q}} (\theta_i^p \mathbf{a}_1^p \wedge \theta_j^q \mathbf{a}_{p+1}^n)$.

$$\begin{aligned}
\theta_0^2(a, b) &\equiv \mathbf{t} \quad , \\
\theta_1^2(a, b) &\equiv (\theta_1^1(a) \wedge \theta_0^1(b)) \vee (\theta_0^1(a) \wedge \theta_1^1(b)) \equiv (a \wedge \mathbf{t}) \vee (\mathbf{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_0^3(a, b, c) &\equiv \mathbf{t} \quad , \\
\theta_1^3(a, b, c) &\equiv (\theta_1^1(a) \wedge \theta_0^2(b, c)) \vee (\theta_0^1(a) \wedge \theta_1^2(b, c)) \equiv (a \wedge \mathbf{t}) \vee (\mathbf{t} \wedge [(b \wedge \mathbf{t}) \vee (\mathbf{t} \wedge c)]) \\
&= a \vee b \vee c \quad , \\
\theta_2^3(a, b, c) &\equiv (\theta_1^1(a) \wedge \theta_1^2(b, c)) \vee (\theta_0^1(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_0^5(a, b, c, d, e) &\equiv \mathbf{t} \quad , \\
\theta_1^5(a, b, c, d, e) &\equiv (\theta_1^2(a, b) \wedge \theta_0^3(c, d, e)) \vee (\theta_0^2(a, b) \wedge \theta_1^3(c, d, e)) \\
&= a \vee b \vee c \vee d \vee e \quad , \\
\theta_2^5(a, b, c, d, e) &\equiv (\theta_2^2(a, b) \wedge \theta_0^3(c, d, e)) \vee (\theta_1^2(a, b) \wedge \theta_1^3(c, d, e)) \vee (\theta_0^2(a, b) \wedge \theta_2^3(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_3^5(a, b, c, d, e) &\equiv (\theta_2^2(a, b) \wedge \theta_1^3(c, d, e)) \vee (\theta_1^2(a, b) \wedge \theta_2^3(c, d, e)) \vee (\theta_0^2(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_4^5(a, b, c, d, e) &\equiv (\theta_2^2(a, b) \wedge \theta_2^3(c, d, e)) \vee (\theta_1^2(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_6^5(a, b, c, d, e) &\equiv \mathbf{f} \quad .
\end{aligned}$$

FIGURE 3. Examples of threshold formulae.

See, in Figure 3, some examples of threshold formulae.

The only reason why we require atoms to be distinct in threshold formulae is to avoid certain technical problems with substitutions in the definition of cut-free form, later on. However, there is no substantial difficulty in relaxing this definition to any set of atoms.

The formulae for threshold functions adopted in [AGP02] correspond, for each choice of k and n , to $\bigvee_{i \geq k} \theta_i^n a_1^n$. We presume that [AGP02] employs these more complicated formulae because the formalism adopted there, the sequent calculus, is less flexible than deep inference, requiring more information in threshold formulae in order to construct suitable derivations.

Remark 15. For $n > 0$, we have $\theta_1^n a_1^n = a_1 \vee \dots \vee a_n$ and $\theta_n^n a_1^n = a_1 \wedge \dots \wedge a_n$.

The size of the threshold formulae dominates the cost of the normalisation procedure, so, we evaluate their size. We leave as an exercise the proof of the following proposition.

Proposition 16. *For any $n > 0$ and $k \geq 0$, $\left| \theta_k^n \mathbf{a}_1^n \right| \leq \left| \theta_{\lfloor n/2 \rfloor + 1}^n \mathbf{a}_1^n \right|$.*

Lemma 17. *The size of $\theta_{\lfloor n/2 \rfloor + 1}^n \mathbf{a}_1^n$ is $n^{O(\log n)}$.*

Proof. Observe that $\left| \theta_k^n \mathbf{a}_1^n \right| \leq \left| \theta_k^{n+1} \mathbf{a}_1^{n+1} \right|$. Let $p = \lfloor n/2 \rfloor$ and $q = n - p$ and consider:

$$\begin{aligned}
 \left| \theta_{p+1}^n \mathbf{a}_1^n \right| &= \sum_{\substack{i+j=p+1 \\ 0 \leq i \leq p \\ 0 \leq j \leq q}} \left(\left| \theta_i^p \mathbf{a}_1^p \right| + \left| \theta_j^q \mathbf{a}_1^q \right| \right) \\
 (3) \quad &\leq \sum_{\substack{i+j=p+1 \\ 0 \leq i, j \leq q}} \left(\left| \theta_i^q \mathbf{a}_1^q \right| + \left| \theta_j^q \mathbf{a}_1^q \right| \right) \\
 &\leq 2(q+1) \left| \theta_{\lfloor q/2 \rfloor + 1}^q \mathbf{a}_1^q \right|,
 \end{aligned}$$

where we use Proposition 16. We show that, for $h = 2/(\log 3 - \log 2)$ and for any $n > 0$, we have $\left| \theta_{\lfloor n/2 \rfloor + 1}^n \mathbf{a}_1^n \right| \leq n^{h \log n}$. We reason by induction on n ; the case $n = 1$ trivially holds. By the inequality (3), and for $n > 1$, we have

$$\begin{aligned}
 \left| \theta_{\lfloor n/2 \rfloor + 1}^n \mathbf{a}_1^n \right| &\leq 2(n - \lfloor n/2 \rfloor + 1)(n - \lfloor n/2 \rfloor)^{h \log(n - \lfloor n/2 \rfloor)} \\
 &\leq n^2 n^{h \log(2n/3)} = n^{h \log n - h(\log 3 - \log 2) + 2} = n^{h \log n}. \quad \square
 \end{aligned}$$

Theorem 18. *For any $k \geq 0$ the size of $\theta_k^n \mathbf{a}_1^n$ is $n^{O(\log n)}$.*

Proof. It immediately follows from Proposition 16 and Lemma 17. \square

Given a threshold formula $\theta_k^n \mathbf{a}_1^n$, we can consider, for each a_l such that $1 \leq l \leq n$, the formulae $(\theta_k^n \mathbf{a}_1^n)\{a_l/f\}$ and $(\theta_{k+1}^n \mathbf{a}_1^n)\{a_l/t\}$: we call both of them, informally, ‘pseudocomplements’ of a_l . The reason for this name is that we can manage to replace, in a given proof, all occurrences of those \bar{a}_l that appear in cut instances with the pseudocomplements of a_l . The cut instances and their corresponding identity instances are then removed, leaving us with derivations whose premiss and conclusion contain each a threshold formula. Moreover, the k -level of the threshold formula in the premiss is one less than the k -level of the threshold formula in the conclusion. This way, we obtain several derivations, corresponding to increasing values of k , that we are able to stitch together until we get a normalised proof.

All this, of course, needs clarification, but we think that it is helpful to provide a summary here of the main constructions that allow for this stitching operation. Let us read derivations top-down; the following are the steps that we need to perform, for $0 \leq k \leq n$.

(1) Build

$$\theta_k^n \mathbf{a}_1^n \quad \parallel \quad a_l \vee (\theta_k^n \mathbf{a}_1^n)\{a_l/f\},$$

i.e., create, from a k -level threshold formula, a disjunction between a_l and its pseudocomplement $(\theta_k^n \mathbf{a}_1^n)\{a_l/f\}$ (Proposition 23); then replace the pseudocomplement into \bar{a}_l , for each identity instance.

(2) Increase the k -level by using the derivations

$$\begin{aligned}
 &(\theta_k^n \mathbf{a}_1^n)\{a_l/f\} \\
 &\parallel \\
 &(\theta_{k+1}^n \mathbf{a}_1^n)\{a_l/t\}
 \end{aligned}$$

(Theorem 22); these are the Γ derivations mentioned in the introduction to this section.

- (3) For each cut instance, collect the conjunction between a_l and its pseudocomplement $(\theta_{k+1}^n \mathbf{a}_1^n)\{a_l/t\}$; then build

$$a_l \wedge (\theta_{k+1}^n \mathbf{a}_1^n)\{a_l/t\} \parallel \theta_{k+1}^n \mathbf{a}_1^n ,$$

i.e., create a $(k+1)$ -level threshold formula (Proposition 23).

The derivations mentioned above do not require any use of identity and cut, and allow us to move, in $n+1$ steps, from $\theta_0^n \mathbf{a}_1^n \equiv t$ to $\theta_{n+1}^n \mathbf{a}_1^n \equiv f$, which is the secret to success. The constructions in 1 and 3 are deep-inference routine and introduce low complexity. We deal now with the crucial step 2, by designing Definition 20, and then checking it carefully, so as to get the property stated in Theorem 22.

Definition 20 is technical, but its philosophy is simple; all one has to do to build the derivations required by Theorem 22 is:

- identify the atom occurrences that must occur in the premiss and that must not occur in the conclusion and remove them using coweakening, and
- identify the atom occurrences that must occur in the conclusion and that must not occur in the premiss and add them using weakening.

We have implemented Definition 20 as a program [Gug09]. It can be useful to read the definition together with the examples in Figures 4 and 3, which have been generated by the program.

Remark 19. Given $n > 1$, let $p = \lfloor n/2 \rfloor$ and $q = n - p$. For $0 \leq k \leq q$ and $1 \leq l \leq p$, the following derivation is well defined:

$$\text{w}\uparrow \frac{(\theta_p^p \mathbf{a}_1^p)\{a_l/f\} \wedge \theta_k^q \mathbf{a}_{p+1}^n}{f} = \text{w}\uparrow \frac{a_1 \wedge \dots \wedge a_{l-1} \wedge a_{l+1} \wedge \dots \wedge a_p \wedge \theta_k^q \mathbf{a}_{p+1}^n \wedge f}{t} .$$

Analogously, for $0 \leq k \leq p$ and $p+1 \leq l \leq n$, we can define the following derivation:

$$\text{w}\uparrow \frac{\theta_k^p \mathbf{a}_1^p \wedge (\theta_q^q \mathbf{a}_{p+1}^n)\{a_l/f\}}{f} = \text{w}\uparrow \frac{\theta_k^p \mathbf{a}_1^p \wedge a_{p+1} \wedge \dots \wedge a_{l-1} \wedge a_{l+1} \wedge \dots \wedge a_n \wedge f}{t} .$$

Both classes of derivations are used in Definition 20.

Definition 20. Consider $n > 0$, distinct atoms a_1, \dots, a_n , and let $p = \lfloor n/2 \rfloor$ and $q = n - p$.

- For $n > 1$ and $1 \leq l \leq n$, we define the derivations $\Upsilon_{k,l}^n \mathbf{a}_1^n$ and $\Delta_{k,l}^n \mathbf{a}_1^n$ as follows:

$$\Upsilon_{k,l}^n \mathbf{a}_1^n = \begin{cases} \text{w}\uparrow \frac{(\theta_p^p \mathbf{a}_1^p)\{a_l/f\} \wedge \theta_{k-p}^q \mathbf{a}_{p+1}^n}{f} & \text{if } p \leq k \leq n \text{ and } l \leq p \\ \text{w}\uparrow \frac{\theta_{k-q}^p \mathbf{a}_1^p \wedge (\theta_q^q \mathbf{a}_{p+1}^n)\{a_l/f\}}{f} & \text{if } q \leq k \leq n \text{ and } p < l \\ f & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\Gamma_{0,1}^5 \mathbf{a} &= \mathfrak{t} \vee \frac{\mathfrak{f}}{b} \vee \frac{\mathfrak{f}}{c \vee d \vee e} \quad , \\
\Gamma_{1,1}^5 \mathbf{a} &= b \vee \left(\left[\mathfrak{t} \vee \frac{\mathfrak{f}}{b} \right] \wedge [c \vee d \vee e] \right) \vee \frac{\mathfrak{f}}{(c \wedge [d \vee e]) \vee (d \wedge e)} \quad , \\
\Gamma_{2,1}^5 \mathbf{a} &= (b \wedge [c \vee d \vee e]) \vee \left(\left[\mathfrak{t} \vee \frac{\mathfrak{f}}{b} \right] \wedge [(c \wedge [d \vee e]) \vee (d \wedge e)] \right) \vee \frac{\mathfrak{f} \wedge b}{\mathfrak{f}} \vee \frac{\mathfrak{f}}{c \wedge d \wedge e} \quad , \\
\Gamma_{3,1}^5 \mathbf{a} &= (b \wedge [(c \wedge [d \vee e]) \vee (d \wedge e)]) \vee \left(\left[\mathfrak{t} \vee \frac{\mathfrak{f}}{b} \right] \wedge c \wedge d \wedge e \right) \vee \frac{\mathfrak{f} \wedge b \wedge [c \vee d \vee e]}{\mathfrak{f}} \quad , \\
\Gamma_{4,1}^5 \mathbf{a} &= (b \wedge c \wedge d \wedge e) \vee \frac{\mathfrak{f} \wedge b \wedge [(c \wedge [d \vee e]) \vee (d \wedge e)]}{\mathfrak{f}} \quad , \\
\Gamma_{5,1}^5 \mathbf{a} &= \frac{\mathfrak{f} \wedge b \wedge c \wedge d \wedge e}{\mathfrak{f}} \quad , \\
\Gamma_{0,3}^5 \mathbf{a} &= \mathfrak{t} \vee \frac{\mathfrak{f}}{d \vee e} \vee \frac{\mathfrak{f}}{a \vee b} \quad , \\
\Gamma_{1,3}^5 \mathbf{a} &= \left([a \vee b] \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d \vee e} \right] \right) \vee d \vee e \vee \frac{\mathfrak{f}}{d \wedge e} \vee \frac{\mathfrak{f}}{a \wedge b} \quad , \\
\Gamma_{2,3}^5 \mathbf{a} &= \left(a \wedge b \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d \vee e} \right] \right) \vee \left([a \vee b] \wedge \left[d \vee e \vee \frac{\mathfrak{f}}{d \wedge e} \right] \right) \vee (d \wedge e) \vee \frac{\mathfrak{f} \wedge [d \vee e]}{\mathfrak{f}} \quad , \\
\Gamma_{3,3}^5 \mathbf{a} &= \left(a \wedge b \wedge \left[d \vee e \vee \frac{\mathfrak{f}}{d \wedge e} \right] \right) \vee \left([a \vee b] \wedge \left[(d \wedge e) \vee \frac{\mathfrak{f} \wedge [d \vee e]}{\mathfrak{f}} \right] \right) \vee \frac{\mathfrak{f} \wedge d \wedge e}{\mathfrak{f}} \quad , \\
\Gamma_{4,3}^5 \mathbf{a} &= \left(a \wedge b \wedge \left[(d \wedge e) \vee \frac{\mathfrak{f} \wedge [d \vee e]}{\mathfrak{f}} \right] \right) \vee \frac{[a \vee b] \wedge \mathfrak{f} \wedge d \wedge e}{\mathfrak{f}} \quad , \\
\Gamma_{5,3}^5 \mathbf{a} &= \frac{a \wedge b \wedge \mathfrak{f} \wedge d \wedge e}{\mathfrak{f}} \quad , \\
\Gamma_{0,5}^5 \mathbf{a} &= \mathfrak{t} \vee \frac{\mathfrak{f}}{d} \vee \frac{\mathfrak{f}}{c} \vee \frac{\mathfrak{f}}{a \vee b} \quad , \\
\Gamma_{1,5}^5 \mathbf{a} &= \left([a \vee b] \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d} \vee \frac{\mathfrak{f}}{c} \right] \right) \vee \left(c \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d} \right] \right) \vee d \vee \frac{\mathfrak{f}}{a \wedge b} \quad , \\
\Gamma_{2,5}^5 \mathbf{a} &= \left(a \wedge b \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d} \vee \frac{\mathfrak{f}}{c} \right] \right) \vee \left([a \vee b] \wedge \left[\left(c \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d} \right] \right) \vee d \right] \right) \vee (c \wedge d) \vee \frac{d \wedge \mathfrak{f}}{\mathfrak{f}} \quad , \\
\Gamma_{3,5}^5 \mathbf{a} &= \left(a \wedge b \wedge \left[\left(c \wedge \left[\mathfrak{t} \vee \frac{\mathfrak{f}}{d} \right] \right) \vee d \right] \right) \vee \left([a \vee b] \wedge \left[(c \wedge d) \vee \frac{d \wedge \mathfrak{f}}{\mathfrak{f}} \right] \right) \vee \frac{c \wedge d \wedge \mathfrak{f}}{\mathfrak{f}} \quad , \\
\Gamma_{4,5}^5 \mathbf{a} &= \left(a \wedge b \wedge \left[(c \wedge d) \vee \frac{d \wedge \mathfrak{f}}{\mathfrak{f}} \right] \right) \vee \frac{[a \vee b] \wedge c \wedge d \wedge \mathfrak{f}}{\mathfrak{f}} \quad , \\
\Gamma_{5,5}^5 \mathbf{a} &= \frac{a \wedge b \wedge c \wedge d \wedge \mathfrak{f}}{\mathfrak{f}} \quad .
\end{aligned}$$

FIGURE 4. Examples of $\Gamma_{k,l}^5 \mathbf{a}$, where $\mathbf{a} = (a, b, c, d, e)$.

and

$$\Delta_{k,l}^n \mathbf{a}_1^n = \begin{cases} \text{wl} \frac{f}{\theta_k^q \mathbf{a}_{p+1}^n} & \text{if } 0 < k \leq q \text{ and } l \leq p \\ \text{wl} \frac{f}{\theta_k^p \mathbf{a}_1^p} & \text{if } 0 < k \leq p \text{ and } p < l \\ f & \text{otherwise} \end{cases} .$$

- For $k \geq 0$ and $1 \leq l \leq n$, we define the derivations $\Gamma_{k,l}^n \mathbf{a}_1^n$, recursively on n , as follows:

- $\Gamma_{0,1}^1(a_1) = t$;
- for $k > 0$, $\Gamma_{k,1}^1(a_1) = f$;
- for $k > n$, $\Gamma_{k,l}^n \mathbf{a}_1^n = f$;
- for $n > 1$ and $k \leq n$, let

$$\Gamma_{k,l}^n \mathbf{a}_1^n = \begin{cases} \bigvee_{\substack{i+j=k \\ 0 \leq i < p \\ 0 \leq j \leq q}} \left(\Gamma_{i,l}^p \mathbf{a}_1^p \wedge \theta_j^q \mathbf{a}_{p+1}^n \right) \vee \Upsilon_{k,l}^n \mathbf{a}_1^n \vee \Delta_{k+1,l}^n \mathbf{a}_1^n & \text{if } l \leq p \\ \bigvee_{\substack{i+j=k \\ 0 \leq i \leq p \\ 0 \leq j < q}} \left(\theta_i^p \mathbf{a}_1^p \wedge \Gamma_{j,l-p}^q \mathbf{a}_{p+1}^n \right) \vee \Upsilon_{k,l}^n \mathbf{a}_1^n \vee \Delta_{k+1,l}^n \mathbf{a}_1^n & \text{if } p < l \end{cases} .$$

Example 21. See, in Figure 4, some example of derivations $\Gamma_{k,l}^n \mathbf{a}_1^n$. Note that, for clarity, we removed all instances of the trivial derivations $\Upsilon_{1,1}^2 \mathbf{a}_1^2 = \Upsilon_{1,2}^2 \mathbf{a}_1^2 = \Upsilon_{1,1}^3 \mathbf{a}_1^3 = \text{wl} \frac{f}{f}$. We can do so because these derivation instances appear as disjuncts.

Theorem 22. For any $n > 0$, $k \geq 0$ and $1 \leq l \leq n$, the derivation $\Gamma_{k,l}^n \mathbf{a}_1^n$ has shape

$$\frac{(\theta_k^n \mathbf{a}_1^n)\{a_l/f\}}{\|\{\text{aw}\downarrow, \text{aw}\uparrow\}}, \frac{(\theta_{k+1}^n \mathbf{a}_1^n)\{a_l/t\}}$$

and $|\Gamma_{k,l}^n \mathbf{a}_1^n|$ is $n^{O(\log n)}$.

Proof. The shape of $\Gamma_{k,l}^n \mathbf{a}_1^n$ can be verified by inspecting Definition 20. For example, this is the case when $n > 1$ and $l \leq p \leq k < q$, where $p = \lfloor n/2 \rfloor$ and $q = n - p$:

$$\frac{(\theta_k^n \mathbf{a}_1^n)\{a_l/f\}}{\Gamma_{k,l}^n \mathbf{a}_1^n} = \frac{(\theta_{k+1}^n \mathbf{a}_1^n)\{a_l/t\}}$$

$$\bigvee_{\substack{i+j=k \\ 0 \leq i < p \\ 0 \leq j \leq q}} \left(\frac{(\theta_i^p \mathbf{a}_1^p)\{a_l/f\}}{\Gamma_{i,l}^p \mathbf{a}_1^p} \wedge \theta_j^q \mathbf{a}_{p+1}^n \right) \vee \text{wl} \frac{(\theta_p^p \mathbf{a}_1^p)\{a_l/f\} \wedge \theta_{k-p}^q \mathbf{a}_{p+1}^n}{f} \vee \text{wl} \frac{f}{\theta_{k+1}^q \mathbf{a}_{p+1}^n} .$$

(Remember that

$$\theta_k^n \mathbf{a}_1^n \equiv \bigvee_{\substack{i+j=k \\ 0 \leq i \leq p \\ 0 \leq j \leq q}} \left(\theta_i^p \mathbf{a}_1^p \wedge \theta_j^q \mathbf{a}_{p+1}^n \right)$$

and $\theta_0^p \mathbf{a}_1^p \equiv t$.) General (co)weakening rule instances can be replaced by atomic ones because of Proposition 2. The size bound on $\Gamma_{k,l}^n \mathbf{a}_1^n$ follows from Proposition 2 and Theorem 18. \square

6. NORMALISATION STEP 2: CUT-FREE FORM

In this section, we define the cut-free form of proofs, based on proofs in simple form. Proofs in cut-free form have no cut instances, but can have coweakening ones, which prevent these proofs from being analytic. Theorem 25, the main result of the section, shows how to obtain a cut-free proof from any proof. Most of the ingenuity of quasipolynomially normalising an SKS proof into one in analytic SKS resides in going from a simple form to a cut-free one. Removing coweakening instances from a cut-free form is easy; we dedicate Section 7 to this.

Before defining the cut-free form, we need to establish the following fact.

Proposition 23. *For any formula A and atom a , there exist derivations whose size is cubic in $|A|$ and that have shape*

$$\frac{A}{\|\{aw\downarrow, ac\downarrow, s\}\|} \quad a \vee A\{a/f\} \quad \text{and} \quad \frac{a \wedge A\{a/t\}}{\|\{aw\uparrow, ac\uparrow, s\}\|} \quad A .$$

Proof. If there are no occurrences of a in A , the desired derivations are

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

If there are $b > 0$ occurrences of a in A , obtain, by repeatedly applying Proposition 10, the following derivations:

$$\left[\frac{A}{\|\{s\}\|} \frac{a \vee \dots \vee a}{(b-1)\text{-ac}\downarrow} \vee A\{a/f\} \right] \quad \text{and} \quad \left(\frac{a}{a \wedge \dots \wedge a} \wedge A\{a/f\} \right) \frac{\|\{s\}\|}{A} .$$

If $|A| = n$, the size of the desired derivations is $O(n^3)$ because we have to apply Proposition 10 at most $O(n)$ times. \square

Definition 24. For $n > 0$, let Π be a proof in simple form over a_1^n , such that it and its atomic flow have shape

$$\frac{\frac{t}{a_1 \vee \bar{a}_1^{\phi_1}} \wedge \dots \wedge \frac{t}{a_n \vee \bar{a}_n^{\phi_n}}}{\Psi \|\|} \quad \text{and} \quad \text{Diagram} ,$$

The diagram shows a proof structure with a top horizontal line and a bottom horizontal line. Between them are several boxes representing logical connectives. From left to right, there are boxes for \neg , \vee , \wedge , \vee , \wedge , \vee , \wedge , and \vee . The boxes for \vee and \wedge are labeled with ϕ_1 and ϕ_n respectively. The top line has \dots between the first and second boxes, and between the last and second-to-last boxes. The bottom line has \dots between the first and second boxes, and between the last and second-to-last boxes.

for some derivation Ψ . For $0 \leq i \leq n+1$, let $\theta_i \equiv \theta_i^n a_1^n$. For $0 \leq k \leq n$, we define the derivations

$$\frac{\theta_k}{\Phi_k \|\text{SKS}\{ai\uparrow\}} \quad A \vee \theta_{k+1}$$

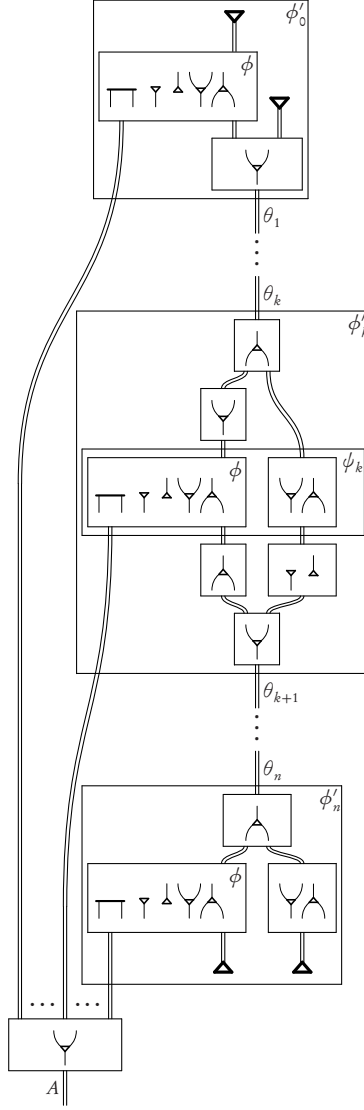


FIGURE 5. Atomic flow of a proof in cut-free form.

7. NORMALISATION STEP 3: ANALYTIC FORM

In this section, we show that we can get proofs in analytic SKS, *i.e.*, system aSKS, in quasipolynomial time from proofs in SKS.

Transforming a proof in cut-free form into an analytic one requires eliminating co-weakening rule instances. This can be done by transformations that are the dual of those over weakening instances, employed in Step (1) of the proof of Theorem 12.

Theorem 27. *Given any proof Π of A in SKS, we can construct a proof of A in aSKS in time quasipolynomial in the size of Π .*

Proof. By Theorem 25, we can obtain, from Π , a cut-free proof Π' of the same formula, in quasipolynomial time in the size of Π . We associate Π' with its atomic flow ϕ , so that we have a way to identify the atom occurrences in Π' associated with each edge of ϕ , and

substitute over them. We repeatedly examine each coweakening instance $\text{aw}\uparrow \frac{a^\epsilon}{t}$ in Π' , for some edge ϵ of ϕ , and we perform one transformation out of the following exhaustive list of cases, for some Π'' , Φ , Ψ , $K\{ \}$ and $H\{ \}$:

(1)

$$\begin{array}{c}
 \Pi'' \parallel \\
 K \left\{ \frac{t}{a^\epsilon \vee \bar{a}} \right\} \\
 \Phi \parallel \\
 H \left\{ \frac{a^\epsilon}{t} \right\} \\
 \Psi \parallel \\
 A
 \end{array}
 \text{ becomes }
 \begin{array}{c}
 \Pi'' \parallel \\
 K \left[\frac{f}{t \vee \bar{a}} \right] \\
 \Phi_{\{a^\epsilon/t\}} \parallel \\
 H\{t\} \\
 \Psi \parallel \\
 A
 \end{array}
 ;$$

(2)

$$\begin{array}{c}
 \Pi'' \parallel \\
 K \left\{ \frac{f}{a^\epsilon} \right\} \\
 \Phi \parallel \\
 H \left\{ \frac{a^\epsilon}{t} \right\} \\
 \Psi \parallel \\
 A
 \end{array}
 \text{ becomes }
 \begin{array}{c}
 \Pi'' \parallel \\
 K \left\{ \frac{f \wedge [t \vee t]}{(f \wedge t) \vee t} \right\} \\
 \Phi_{\{a^\epsilon/t\}} \parallel \\
 H\{t\} \\
 \Psi \parallel \\
 A
 \end{array}
 ;$$

(3)

$$\begin{array}{c}
 \Pi'' \parallel \\
 K \left\{ \frac{a \vee a}{a^\epsilon} \right\} \\
 \Phi \parallel \\
 H \left\{ \frac{a^\epsilon}{t} \right\} \\
 \Psi \parallel \\
 A
 \end{array}
 \text{ becomes }
 \begin{array}{c}
 \Pi'' \parallel \\
 K \left[\frac{a}{t} \vee \frac{a}{t} \right] \\
 \Phi_{\{a^\epsilon/t\}} \parallel \\
 H\{t\} \\
 \Psi \parallel \\
 A
 \end{array}
 ;$$

(4)

$$\begin{array}{c}
 \Pi'' \parallel \\
 K \left\{ \frac{a}{a^\epsilon \wedge a} \right\} \\
 \Phi \parallel \\
 H \left\{ \frac{a^\epsilon}{t} \right\} \\
 \Psi \parallel \\
 A
 \end{array}
 \text{ becomes }
 \begin{array}{c}
 \Pi'' \parallel \\
 K\{a\} \\
 \Phi_{\{a^\epsilon/t\}} \parallel \\
 H\{t\} \\
 \Psi \parallel \\
 A
 \end{array}
 .$$

The process terminates in linear time on the size of Π' because each transformation eliminates some atom occurrences. The final proof is in aSKS. \square

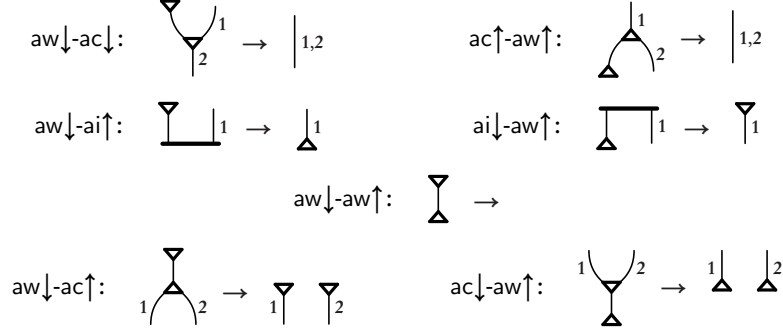
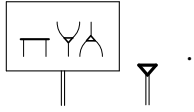


FIGURE 6. Weakening and coweakening atomic-flow reductions.

The transformations described in the proof of Theorem 27 are the minimal ones necessary to produce a proof in aSKS. However, it is possible to further reduce the proof so obtained. The transformations in the proof of Theorem 27, together with the one mentioned in Step (1) in the proof of Theorem 12, all belong to the class of weakening and coweakening reductions studied in [GG08]. In the rest of this section, we quickly outline a possible, further transformation of the analytic form produced by those reductions, and refer the reader to [GG08] for a more thorough explanation.

It is advantageous to describe the weakening and coweakening transformations directly as atomic-flow reduction rules. These are special graph rewriting rules for atomic flows, that are known to correspond to sound derivation transformations, in the following sense. If Φ is a derivation with flow ϕ , and ϕ can be transformed into ψ by one of the atomic-flow reduction rules, then there exists a derivation Ψ whose flow is ψ and such that it has the same premiss and conclusion as Φ . Moreover, Ψ can be obtained from Φ by instantiating some atoms and changing some rule instances, in linear time.

The weakening and coweakening atomic-flow reduction rules are shown in Figure 6. The reduction rule labelled $\text{aw}\downarrow\text{-ai}\uparrow$ is employed in Step (1) in the proof of Theorem 12. The reduction rules labelled $\text{ac}\uparrow\text{-aw}\uparrow$, $\text{ai}\downarrow\text{-aw}\uparrow$, $\text{aw}\downarrow\text{-aw}\uparrow$ and $\text{ac}\downarrow\text{-aw}\uparrow$ are employed in the proof of Theorem 27, respectively as Case (4), (1), (2) and (3). If we apply the full set of weakening and coweakening reductions until possible, starting from a proof in cut-free form, we obtain a proof of the same formula and whose flow has shape



Note that the graph rewriting system consisting of the reductions in Figure 6 is confluent.

8. FINAL COMMENTS

System aSKS is not a minimal complete system for propositional logic, because the atomic cocontraction rule $\text{ac}\uparrow$ is admissible (via $\text{ac}\downarrow$, $\text{ai}\uparrow$ and s). Removing $\text{ac}\uparrow$ from aSKS yields system KS. A natural question is whether quasipolynomial normalisation holds for KS as well. We do not know, and all indications and intuition point to an essential role being played by cocontraction in keeping the complexity low. Analysing Figure 5 shows how cocontraction provides for a typical ‘dag-like’ speed-up over the corresponding ‘tree-like’ expansion consisting in generating some sort of Gentzen tree. However, we are aware that in the past this kind of intuition has been fallacious.

Of course, we can consider the role of cocontraction as an explanation of why quasipolynomial normalisation is not achievable 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.

As mentioned, there is reason to believe that polynomial normalisation is achievable, because it is possible to express threshold functions with polynomial formulae. However, the hardest problem seems to be obtaining polynomial Γ -like (cut-free) derivations with the property of Theorem 22. We tend to think that polynomiality ought to be possible, and deep inference should help because it has unprecedented flexibility in constructing derivations. We intend to investigate this possibility vigorously.

One might think of employing the (co)weakening atomic-flow reductions of Figure 6 to further simplify the present threshold formulae. We are sceptical that this could prove useful, and we base this judgement, again, on a careful analysis of Figure 5. Essentially, the (co)weakening reductions tend to be blocked at some point by the (co)contraction vertices.

The normalisation procedure presented here is peculiar because it achieves its result by using an external scheme, constituted by the threshold functions and the Γ derivations, which does not depend on the derivation to be normalised. It is as if the threshold construction was a (rather big) catalyser that favours the normalising reaction. It would be interesting to interpret this phenomenon computationally, in some sort of Curry-Howard correspondence, where the threshold construction implements a clever sharing mechanism. We intend to explore this path in the near future.

It seems possible, with some work, to extend the mechanism investigated here to the more general notion of normalisation that we called *streamlining* in [GG08]. Streamlining is a top-down symmetric notion, that does full justice to the additional symmetry of deep inference, compared to Gentzen formalisms. Streamlined derivations entail analytic proofs as a special case.

Finally, we are interested in the normalisation theory of modal logics in deep inference, and so we are naturally led to consider the methods presented in this paper to that purpose as well.

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.
- [BG07] Paola Bruscoli and Alessio Guglielmi. On analytic inference rules in the calculus of structures. <http://cs.bath.ac.uk/ag/p/0nan.pdf>, 2007.
- [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/PrComp1DI.pdf>.
- [Bru02] Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference*, volume 2401 of *Lecture Notes in Computer Science*, pages 302–316. Springer-Verlag, 2002. <http://cs.bath.ac.uk/pb/bv1/bv1.pdf>.
- [Brü03] Kai Brünnler. Atomic cut elimination for classical logic. In M. Baaz and J. A. Makowsky, editors, *CSL 2003*, volume 2803 of *Lecture Notes in Computer Science*, pages 86–97. Springer-Verlag, 2003. <http://www.iam.unibe.ch/~kai/Papers/ace.pdf>.
- [Brü04] Kai Brünnler. *Deep Inference and Symmetry in Classical Proofs*. Logos Verlag, Berlin, 2004. <http://www.iam.unibe.ch/~kai/Papers/phd.pdf>.
- [Brü06a] Kai Brünnler. Cut elimination inside a deep inference system for classical predicate logic. *Studia Logica*, 82(1):51–71, 2006. <http://www.iam.unibe.ch/~kai/Papers/q.pdf>.
- [Brü06b] 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>.
- [Brü06c] Kai Brünnler. Deep sequent systems for modal logic. In Guido Governatori, Ian Hodkinson, and Yde Venema, editors, *Advances in Modal Logic*, volume 6, pages 107–119. College Publications, 2006. <http://www.aiml.net/volumes/volume6/Bruennler.ps>.
- [Brü06d] Kai Brünnler. Locality for classical logic. *Notre Dame Journal of Formal Logic*, 47(4):557–580, 2006. <http://www.iam.unibe.ch/~kai/Papers/LocalityClassical.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 Computer Science*, pages 347–361. Springer-Verlag, 2001. <http://www.iam.unibe.ch/~kai/Papers/lc1-lpar.pdf>.
- [Bus91] Samuel R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53(1):75–102, 1991.
- [DG04] Pietro Di Gianantonio. Structures for multiplicative cyclic linear logic: Deepness vs cyclicity. In J. Marcinkowski and A. Tarlecki, editors, *CSL 2004*, volume 3210 of *Lecture Notes in Computer Science*, pages 130–144. Springer-Verlag, 2004. <http://www.dimi.uniud.it/~pietro/papers/Soft-copy-ps/sc11.ps.gz>.
- [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>.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- [GS01] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In L. Fribourg, editor, *CSL 2001*, volume 2142 of *Lecture Notes in Computer Science*, pages 54–68. Springer-Verlag, September 2001. <http://cs.bath.ac.uk/ag/p/NoncMELLCoS.pdf>.
- [GS02] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *Lecture Notes in Computer Science*, pages 231–246. Springer-Verlag, October 2002. <http://www.lix.polytechnique.fr/~lutz/papers/NEL.pdf>.
- [GS09] Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure V: The exponentials and splitting. Submitted. <http://www.lix.polytechnique.fr/~lutz/papers/NEL-splitting.pdf>, 2009.
- [GT07] Rajeev Goré and Alwen Tiu. Classical modal display logic in the calculus of structures and minimal cut-free deep inference calculi for S5. *Journal of Logic and Computation*, 17(4):767–794, 2007. <http://users.rsise.anu.edu.au/~tiu/papers/cmdl.pdf>.
- [Gug] Alessio Guglielmi. Deep inference. 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>.
- [Gug09] Alessio Guglielmi. Th.pl, 2009. Prolog program. <http://cs.bath.ac.uk/ag/p/th.pl>.
- [HS05] Robert Hein and Charles Stewart. Purity through unravelling. In Paola Bruscoli, François Lamarche, and Charles Stewart, editors, *Structures and Deduction*, pages 126–143. Technische Universität Dresden, 2005. ICALP Workshop. ISSN 1430-211X. <http://cs.bath.ac.uk/pb/SD05/SD05-Proc.pdf>.
- [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>.
- [Kah06] Ozan Kahramanoğulları. Reducing nondeterminism in the calculus of structures. In M. Hermann and A. Voronkov, editors, *LPAR 2006*, volume 4246 of *Lecture Notes in Computer Science*, pages 272–286. Springer-Verlag, 2006. http://dx.doi.org/10.1007/11916277_19.
- [Kah07] Ozan Kahramanoğulları. System BV is NP-complete. *Annals of Pure and Applied Logic*, 152(1–3):107–121, 2007. <http://dx.doi.org/10.1016/j.apal.2007.11.005>.
- [SG09] Lutz Straßburger and Alessio Guglielmi. A system of interaction and structure IV: The exponentials and decomposition. Submitted. <http://www.lix.polytechnique.fr/~lutz/papers/NEL-decomposition.pdf>, 2009.
- [SS05] Charles Stewart and Phiniki Stouppa. A systematic proof theory for several modal logics. In Renate Schmidt, Ian Pratt-Hartmann, Mark Reynolds, and Heinrich Wansing, editors, *Advances in Modal Logic*, volume 5, pages 309–333. King’s College Publications, 2005. <http://www.aiml.net/volumes/volume5/Stewart.ps>.
- [Sta78] Richard Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic*, 15:225–287, 1978.
- [Sto07] Phiniki Stouppa. A deep inference system for the modal logic S5. *Studia Logica*, 85(2):199–214, 2007. <http://www.iam.unibe.ch/til/publications/pubitems/pdfs/sto07.pdf>.
- [Str02] Lutz Straßburger. A local system for linear logic. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *Lecture Notes in Computer Science*, pages 388–402. Springer-Verlag, 2002. <http://www.lix.polytechnique.fr/~lutz/papers/l1s-lpar.pdf>.
- [Str03a] Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden, 2003. <http://www.lix.polytechnique.fr/~lutz/papers/dissvonlutz.pdf>.
- [Str03b] Lutz Straßburger. MELL in the calculus of structures. *Theoretical Computer Science*, 309:213–285, 2003. <http://www.lix.polytechnique.fr/~lutz/papers/e1s.pdf>.
- [Tiu06a] Alwen Tiu. A local system for intuitionistic logic. In M. Hermann and A. Voronkov, editors, *LPAR 2006*, volume 4246 of *Lecture Notes in Computer Science*, pages 242–256. Springer-Verlag, 2006. <http://users.rsise.anu.edu.au/~tiu/localint.pdf>.

- [Tiu06b] Alwen Tiu. A system of interaction and structure II: The need for deep inference. *Logical Methods in Computer Science*, 2(2:4):1–24, 2006. <http://arxiv.org/pdf/cs.L0/0512036>.
- [Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons Ltd and B. G. Teubner, Stuttgart, 1987.

BRUSCOLI, GUGLIELMI, GUNDERSEN: UNIVERSITY OF BATH (UK) AND LORIA & INRIA NANCY-GRAND EST (FRANCE).

PARIGOT: LABORATOIRE PPS, UMR 7126, CNRS & UNIVERSITÉ PARIS 7 (FRANCE).