A study of normalisation through subatomic logic

Andrea Aler Tubella

Proof theorists have long been interested in the study of normalisation of proofs. From cut-elimination to proof identity, finding a normal form for proofs is a valuable research goal that includes questions such as which properties we would like for the normal form, and what the size of the normal form is in relation to the original proof. However, to study normalisation procedures with some generality is very difficult: cut-elimination procedures for example are highly sensitive to variations on the form and structure of the rules of a system, where a single change in one of the rules or the addition of another warrant the need for a full new proof of cut-elimination in a new system. In this thesis we unveil a common structure behind proof systems that will allow us to generalise and understand normalisation in a simpler and more effective way. We provide a new approach within the setting of deep inference, which we call *subatomic* because we look 'inside the atoms'. It allows us to present a wide variety of propositional proof systems in such a way that *every rule* is an instance of a *single* simple linear rule scheme. We exploit this generality to study normalisation procedures and their complexity, and in particular to unveil the role played by the interactions between the rules.

Gentzen's proofs of cut-elimination [9] for classical and intuitionistic logic were only the first instance of a type of argument that has been long studied since. From that breakthrough, Gentzen-style cut elimination proofs abound in the literature, exploring on a system-by-system basis how to permute the cut-rules towards the premiss of a proof. The specificity needed for these cut-elimination arguments requires tricky case by case analyses, making it difficult to understand *how* cut-elimination works. Indeed, when designing a new proof system a complex trial and error phase is necessary to obtain cut admissibility. The fact that simple variations of a rule have so much influence on these arguments is the first hint that cut-elimination is in fact a combinatorial phenomenon, hinging mostly on the shape and interaction between the rules of a system.

In particular, in traditional Gentzen-style cut-elimination procedures cut instances are eliminated from proofs by moving upwards instances of the *mix* rule [10, 8]. This rule conflates one instance of cut and several instances of contraction and therefore by using this technique we are in fact observing two different interactions between rules: the interactions of the cut with other rules, and the interactions of contractions with other rules. This phenomenon becomes more apparent when one considers the complexity of cut-elimination in different systems: in purely linear systems such as multiplicative linear logic the procedure does not change the size of proofs significantly, whereas as soon as contractions are introduced the size of proofs can grow exponentially or more.

In what follows we aim to move towards a generalised modular normalisation theory where the different interactions between rules are dealt with separately, providing a tighter control over complexity creation. We provide a generalised procedure for cut-elimination in a generalisation of linear systems, named *splitting*. Further, we present general proof rewriting rules together with sufficient conditions for a system to be decomposable into phases containing only atomic contractions/cocontractions and a linear phase. In this way we show that this type of decomposition result holds for example for both classical logic and multiplicative additive linear logic because of shared properties in the shape of their rules. Last, we use the general reduction rules introduced in this thesis to design a local procedure to remove cycles, effectively proving the independence of decomposition and cut-elimination.

1 Subatomic logic

To reduce rules to a single shape, we will work in the setting of deep inference [13, 21]. In deep inference proofs can be composed by the logical connectives that are used to compose formulae [14]. For example,

if $\phi = \begin{array}{c} A \\ \| \\ B \end{array}$ and $\psi = \begin{array}{c} C \\ \| \\ D \end{array}$ are two proofs in propositional logic,

$$\phi \wedge \psi = \begin{array}{ccc} A & & C \\ \| & \wedge & \| \\ B & & D \end{array} \quad \text{and} \quad \phi \vee \psi = \begin{array}{ccc} A & & C \\ \| & \vee & \| \\ B & & D \end{array}$$

are two valid proofs with premisses $A \wedge C$ and $A \vee C$ and conclusions $B \wedge D$ and $B \vee D$ respectively. In deep inference, rules can be applied at any depth inside a formula and as a result every contraction and cut instance can be locally transformed into their atomic variants by a local procedure of polynomial-size complexity [5, 20, 4].

This provides a surprising regularity in the inference rule schemes: it can be observed that in most deep inference systems all rules besides the atomic ones can be expressed as

$$\frac{(A \ \alpha \ B) \ \beta \ (C \ \alpha' \ D)}{(A \ \beta \ C) \ \alpha \ (B \ \beta' \ D)} \quad ,$$

where A, B, C, D are formulae and $\alpha, \beta, \alpha', \beta'$ are connectives. We call this rule shape a *medial shape*. Following this discovery, we will achieve an even greater regularity on the inference rules by looking even further, *inside the atoms*.

The main idea of this work is to consider atoms as self-dual non-commutative relations. Subatomic formulae are built by freely composing constants by connectives and atoms. For example, $A \equiv ((\mathsf{f} \ a \ \mathsf{t}) \vee \mathsf{t}) \wedge (\mathsf{t} \ b \ \mathsf{f})$ is a subatomic formula for classical logic. The intuitive idea is to interpret $\mathsf{f} \ a \ \mathsf{t}$ as a positive occurrence of the atom a, and $\mathsf{t} \ a \ \mathsf{f}$ as a negative occurrence of the same atom, denoted by $\bar{a}$. We can therefore interpret A as $(a \vee \mathsf{t}) \wedge \bar{b}$.

We can view subatomic formulae as a superposition of truth values. For example, $\mathsf{f} \ a \ \mathsf{t}$ is the superposition of the two possible assignments for the atom a, and $\mathsf{t} \, a \, \mathsf{f}$ is the superposition of the possible assignments for $\bar{a}$: if we read the value on the left of the atom we assign f to a and t to $\bar{a}$, and if we read the one on the right we assign t to a and f to $\bar{a}$.

We give a broad definition of what relations are, not assuming any logical characteristics or properties such as commutativity or associativity. We therefore encompass logics with both commutative and non-commutative, associative and non-associative, dual and-self dual relations. This feature deserves to be highlighted since expressing self-dual non-commutative connectives into proof systems that enjoy cut-elimination is a challenge in Gentzen-style sequent calculi: it is impossible to have a complete analytic system with a self-dual non-commutative relation [21].

Definition 1. Let $\mathscr{U}$ be a denumerable set of *constants* whose elements are denoted by $u, v, w, \ldots$. Let $\mathscr{R}$ be a denumerable partially ordered set of *connectives* whose elements are denoted by α, β, γ, ... The set $\mathscr{F}$ of *subatomic formulae* (or SA formulae) contains terms defined by the grammar

$$\mathscr{F} ::= \mathscr{U} \mid \mathscr{F} \ \mathscr{R} \ \mathscr{F} \quad .$$

Formulae are denoted by A, B, C, ...

A *(formula) context* $K\{\ \} \cdots \{\ \}$ is a formula where some subformulae are substituted by holes; $K\{A_1\} \cdots \{A_n\}$ denotes a formula where the n holes in $K\{\ \} \cdots \{\ \}$ have been filled with A_1, ..., A_n.

In this summary we will only use classical logic as an example. Further examples featuring multiplicative linear logic, multiplicative additive linear logic, BV (showcasing a logic with a self-dual non-commutative connective) and SKV (which features a modality) can be found in the thesis.

Given a propositional logic with certain connectives and constants, its subatomic counterpart is therefore composed of an extended language of formulae, made up from the same connectives and atoms. We can translate subatomic formulae constructed in this natural way into the 'usual' formulae by defining a simple interpretation map. Further, we can easily endow subatomic formulae with an equational theory and an involutive negation, matching that of the 'usual' formulae.

Definition 2. Let $\mathscr{G}$ be the set of formulae of a propositional logic L , and let $\mathscr{F}$ be the set of subatomic formulae with constants $\mathscr{U}$ and connectives $\mathscr{R}$. A surjective partial function $I : \mathscr{F} \to \mathscr{G}$ is called *interpretation map*. The domain of definition of I is the *set of interpretable formulae* and is denoted by $\mathscr{F}^i$. If $F \equiv I(A)$, we say that F is the *interpretation* of A, and that A is a *representation* of F.

The useful properties of subatomic formulae become apparent when we extend the principle to atomic inference rules. Let us consider, for example, the usual contraction rule for an atom. We could obtain this rule subatomically by reading $\mathsf{f}\,a\,\mathsf{t}$ as a and $\mathsf{t}\,a\,\mathsf{f}$ as $\bar{a}$, as follows:

$$\text{we read } \frac{(\mathsf{f}\,a\,\mathsf{t})\vee(\mathsf{f}\,a\,\mathsf{t})}{(\mathsf{f}\vee\mathsf{f})\,a\,(\mathsf{t}\vee\mathsf{t})} \text{ as } \frac{a\vee a}{a} \text{ and we read } \frac{(\mathsf{t}\,a\,\mathsf{f})\vee(\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{t}\vee\mathsf{t})\,a\,(\mathsf{t}\vee\mathsf{t})} \text{ as } \frac{\bar{a}\vee\bar{a}}{\bar{a}}.$$

These rules are therefore generated by the linear scheme

$$\frac{(A\,a\,B)\vee(C\,a\,D)}{(A\vee C)\,a\,(B\vee D)}, \text{ where } A,B,C,D \text{ are formulae.}$$

The non-linearity of the contraction rule has been pushed from the atoms to the units.

Similarly, we can consider the atomic identity rule. It can be obtained subatomically as follows:

$$\text{we read } \frac{(\mathsf{f}\,a\,\mathsf{t})\vee(\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{f}\vee\mathsf{t})\,a\,(\mathsf{t}\vee\mathsf{f})} \text{ as } \frac{\mathsf{t}}{a\vee\bar{a}}.$$

Similarly to the contraction rule, it is generated by the linear scheme

$$\frac{(A\vee B)\,a\,(C\vee D)}{(A\,a\,C)\vee(B\,a\,D)}, \text{ where } A,B,C,D \text{ are formulae.}$$

It is quite plain to see that both the subatomic contraction rule and the subatomic introduction rule have the same *medial* shape, typical of logical rules in deep inference. We have therefore uncovered an underlying structure behind the shape of inference rules, that we will exploit to obtain a general characterisation of rules.

To make use of the general characterisation, we will impose some restrictions on α,ν,β,γ. These conditions strike a balance between being general enough to encompass a wide variety of logics and being explicit enough to enable us to generalise procedure such as cut-elimination and decomposition. To do so, we exploit the dualities present in the inference rules, and we introduce a notion of polarity in the pairs of dual relations. This notion of polarity can be reminiscent of the polarities assigned to connectives in linear logic [11], but the idea behind it is rather to assign which of the relations in the pair is 'stronger' than the other. Intuitively, it loosely corresponds to assigning which relation of the pair will imply the other. For example, in classical logic $A\wedge B$ implies $A\vee B$, and thus we will assign $\wedge$ to be *strong* and $\vee$ to be *weak*.

Definition 3. For each pair of connectives $\{\alpha,\overline{\alpha}\}$, we give a polarity assignment: we call one connective of the pair *strong* and the other one *weak*.

If α is strong and $\overline{\alpha}$ is weak, we will write $\alpha^M{=}\overline{\alpha}^M{=}\alpha$ and $\alpha^m{=}\overline{\alpha}^m{=}\overline{\alpha}$. Self-dual connectives are both strong and weak.

Definition 4. A *subatomic proof system* SA with set of formulae $\mathscr{F}$ is

- a collection of inference rules of the shape $\dfrac{(A\;\beta\;B)\;\alpha\;(C\;\beta\;D)}{(A\;\alpha\;C)\;\beta\;(B\;\alpha^m\;D)}$, $\alpha,\beta\in\mathscr{R}$, called *down-rules*,
- a collection of inference rules of the shape $\dfrac{(A\;\beta\;B)\;\alpha\;(C\;\beta^M\;D)}{(A\;\alpha\;C)\;\beta\;(B\;\alpha\;D)}$, $\alpha,\beta\in\mathscr{R}$, called *up-rules*,
- a collection of rules $=\dfrac{A}{B}$ and $=\dfrac{\overline{A}}{\overline{B}}$, for every axiom $A=B$ of the equational theory $=$ on $\mathscr{F}$, called *equality rules*.

We can straightforwardly build deep inference derivations as is usual in the literature, by vertical composition through an inference rule and horizontal composition by logical relations, and the interpretation map is easily extended from formulae to derivations. The notion of proof is generalised as well.

Definition 5. Let $1\in\mathscr{U}$ be a distinguished constant. A *proof* of A is a derivation ϕ whose premiss is 1. We denote proofs by $\overset{\phi}{\underset{A}{\Pi}}$.

$$a{\downarrow}\,\frac{(A \vee B)\; a\; (C \vee D)}{(A\; a\; C) \vee (B\; a\; D)} \qquad a{\uparrow}\,\frac{(A\; a\; B) \wedge (C\; a\; D)}{(A \wedge C)\; a\; (B \wedge D)}$$

$$\wedge{\downarrow}\,\frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)} \qquad \vee{\uparrow}\,\frac{(A \vee B) \wedge (C \wedge D)}{(A \wedge C) \vee (B \wedge D)}$$

$$m\,\frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}$$

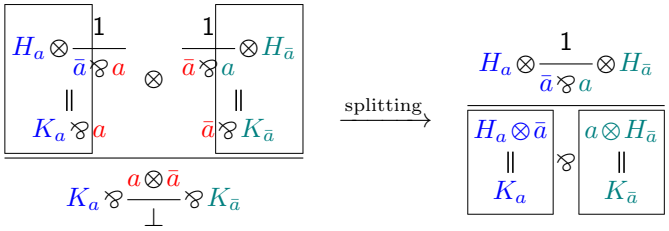
$$ac\,\frac{(A\; a\; B) \vee (C\; a\; D)}{(A \vee C)\; a\; (B \vee D)} \qquad a\bar{c}\,\frac{(A \wedge B)\; a\; (C \wedge D)}{(A\; a\; C) \wedge (B\; a\; D)}$$

Figure 1: *Subatomic system* SAKS *for classical logic*

2 Splitting

There are many different cut-elimination techniques in the deep inference literature [12, 3, 2, 20, 15], exploiting different aspects of the proof systems they work on. In this assortment, a particular methodology does however stand out for its generality: cut-elimination via *splitting* [13] can be achieved in the deep inference systems for linear logic [18], multiplicative exponential linear logic [20], the mixed commutative/non-commutative logic BV [13] and its extension with linear exponentials NEL [15], and classical predicate logic [3]. The generality of this procedure points towards the fact that it exploits some properties that are common to all these systems.

Splitting is based on a simple idea: to show that an atomic cut involving a and $\bar{a}$ is admissible, we trace a and $\bar{a}$ to the top of the proof to find two independent subproofs, the premiss of one containing the dual of a and the other one containing the dual of $\bar{a}$. In this way we obtain two independent 'pieces' that we can rearrange to get a new cut-free proof.



Proofs of cut-elimination by splitting therefore rely on two main properties of a proof system: the *dualities* present in it to ensure that each of the independent subproofs contains the dual of an atom involved in the cut, and the *shape* of the linear rules ensuring that the two proofs remain independent above the cut. It is precisely a formal characterisation of these properties that we will provide, enabling us to understand why they are enough to guarantee cut-elimination.

To trace a connective through the proof from the bottom to the top, we need its scope to widen. Accordingly, we will consider systems where the shape of the rules ensures the widening of the scope. In what follows, we will characterise *splittable systems*, *i.e.*, systems with sufficient conditions to ensure cut-elimination through a splitting procedure.

Definition 6. A system $\mathsf{SA}^{\downarrow}$ is *splittable* if:

1. There is a distinguished associative and commutative strong connective $\times$ with unit 1 and dual $+$ with unit 0,
2. SA is uniquely composed of down-rules of the form

$$\alpha{\downarrow}\,\frac{(A + B)\; \alpha\; (C + D)}{(A\; \alpha\; C) + (B\; \alpha^{m}\; D)} \quad ,$$

for every connective $\alpha \in \mathscr{R}$.

3. There is a constant assignment $u + \bar{u} = 1$ for every unit $u \in \mathcal{U}$,

4. $1\ \alpha^M\ 1 = 1$ for every $\alpha \in \mathcal{R}$.

The proof of the splitting result is done in two steps for ease of reading: shallow splitting and context reduction, just as is standard in the literature. As noted in [13] and in [19], the main difficulty of splitting is finding the right induction measure for every system. In the literature, each splitting theorem for each proof system uses a different induction measure tailored specifically for it. By providing a general splitting theorem, we not only give a formal definition of what a splitting theorem is, but also give a new one-size-fits-all induction measure based on the length of the proof that works for every splittable system, taking the search for an induction measure out of the process for designing a proof system.

Theorem 7 (Shallow Splitting)**.** *If* $\mathsf{SA}^{\downarrow}$ *is splittable, for every formulae A, B, C, for every connective* $\alpha \neq +$*, for every proof*

$$\overset{\phi \,\P\, \mathsf{SA}^{\downarrow}}{(A\ \alpha\ B) + C}$$

there exist formulae Q_1*,* Q_2 *and derivations*

$$\begin{array}{c} Q_1\ \overline{\alpha}\ Q_2 \\ \psi \,\|\, \mathsf{SA}^{\downarrow} \\ C \end{array} \quad , \quad \begin{array}{c} \phi_1 \,\P\, \mathsf{SA}^{\downarrow} \\ A + Q_1 \end{array} \quad \textit{and} \quad \begin{array}{c} \phi_2 \,\P\, \mathsf{SA}^{\downarrow} \\ B + Q_2 \end{array} \quad .$$

Theorem 8 (Context Reduction)**.** *Let* $\mathsf{SA}^{\downarrow}$ *be a splittable system. For any formula A and for any context* $S\{\ \}$*, given a proof* $\overset{\phi \,\P\, \mathsf{SA}^{\downarrow}}{S\{A\}}$*, there exist a formula K, a provable context* $H\{\ \}$ *and derivations*

$$\begin{array}{c} \zeta \,\P\, \mathsf{SA}^{\downarrow} \\ A + K \end{array} \quad \textit{and} \quad \begin{array}{c} H\{\{\ \} + K\} \\ \chi \,\|\, \mathsf{SA}^{\downarrow} \\ S\{\ \} \end{array} \quad .$$

As a corollary of shallow splitting and context reduction we can show the admissibility of a class of up-rules. The main idea is that through context reduction followed by shallow splitting we can separate a proof into 'building blocks' that are independently provable. We then combine these building blocks differently to obtain a new proof with the same conclusion.

Definition 9. Rules of the form $\alpha\uparrow \dfrac{(A\ \alpha\ B) \times \big(C\ \alpha^M\ D\big)}{(A \times C)\ \alpha\ (B \times D)}$ are *cuts.*

Corollary 10 (Admissibility of cuts)**.** *Let* SA *be a splittable proof system.*

For any formulae A, B, C, D*, any context* S*, any connective* $\alpha \neq +$*, given a proof*

$$\phi \equiv S\left\{ \alpha\uparrow \frac{(A\ \alpha\ B) \times \big(C\ \alpha^M\ D\big)}{(A \times C)\ \alpha\ (B \times D)} \right\} \quad ,$$

there is a proof

$$\overset{\pi \,\P\, \mathsf{SA}^{\downarrow}}{S\{(A \times C)\ \alpha\ (B \times D)\}} \quad ,$$

i.e., cuts are admissible.

All of the theorems above preserve interpretability, i.e., if a subatomic proof is interpretable into an 'ordinary' proof, then so is its cut-free form obtained via splitting.

Last, we have shown that splitting hinges only on the shape of rules and on dualities. In the general splitting theorem that we presented we considered only binary relations, but it will be the focus of future research to extend this result to include relations of different arities: splitting can be applied to different types of unary operators, as is shown by the splitting theorems for exponentials in [19] or for a self-dual binder in [17]. In the thesis we show a starting point in the direction of such a generalisation, by extending the general procedure to a system with a self-dual modality. The fact that it is possible to do so shows the robustness of the general splitting methodology: it is based on properties that are present in systems with very different expressiveness and therefore it can be expanded to include an extremely wide variety of relations as long as they are introduced by rules of non-contractive shape.

3 Decomposition

Splitting allows us to understand the interactions of the cut with linear rules, but how about contractions? It is known that we can decompose classical logic [16] and multiplicative additive linear logic [18] proofs into a linear phase and a phase made-up only of contractions. We study this phenomenon, providing general rewriting rules that encompass the reductions presented in both systems. We thus show that both decomposition results are a consequence of precisely the same properties.

Additionally, it has long been conjectured [5] that it is possible to achieve a further decomposition of these systems, permuting not only the atomic contraction but a whole family of *contractive* rules towards the bottom of a derivation. The generalised rewriting rules that we present allow us to permute contractive rules with linear rules, including cuts. The regularity provided by subatomic systems is a big simplification for the study of these interactions: by having a single shape we only have to consider two non-trivial permutation cases.

The first step in the generalisation is to characterise the *contractions*, the family of rules that will be permuted. Unsurprisingly, the rules that we will be able to permute downwards/upwards in a derivation correspond to the rules involved in making contraction atomic. We will call them *contractions* as well.

We define *ν-contractive systems*, which correspond to those systems where we can recover *general contractions*

$$\frac{A \;\nu\; A}{A} \quad .$$

Definition 11. Let ν be a distinguished relation with unit $\triangledown$, and $\overline{\nu}$ its dual with unit $\triangle$. A *ν-contractive system* SA is a subatomic proof system where:

- For every relation α there is a down rule of the form

$$\alpha c\,\frac{(A \;\alpha\; B) \;\nu\; (C \;\alpha\; D)}{(A \;\nu\; C) \;\alpha\; (B \;\nu\; D)} \quad ,$$

  that we call *contraction for* α and its dual up rule that we call *cocontraction for* $\overline{\alpha}$.

- For every constant $u \in \mathscr{U}$ there are equalities of the form $u \;\nu\; u = u$ and $\bar{u} \;\overline{\nu}\; \bar{u} = \bar{u}$. We call the equality rules $=\dfrac{u \;\nu\; u}{u}$ the *contraction equality rule for* u and $=\dfrac{\bar{u}}{\bar{u} \;\overline{\nu}\; \bar{u}}$ the *cocontraction equality rule for* $\bar{u}$.

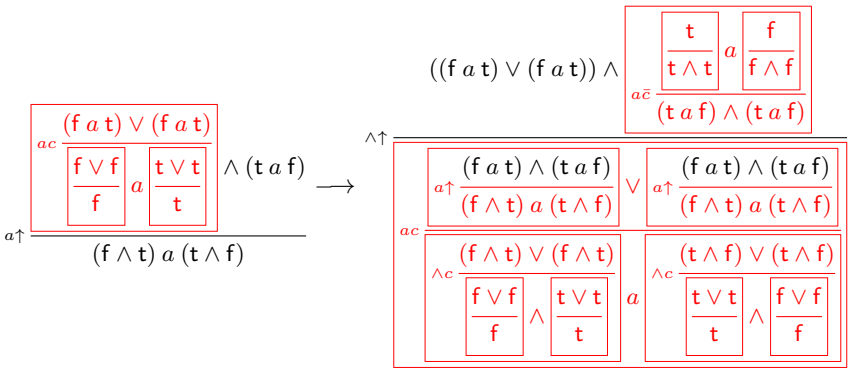
- For every constant $u \in \mathscr{U}$, $w\dfrac{\triangledown}{u}$ and its dual $\bar{w}\dfrac{\bar{u}}{\triangle}$ are derivable in SA. We call these unitary instances of (co)contraction rules *weakening* and *coweakening* respectively.

- For every relation α there are equalities $\triangledown \;\alpha\; \triangledown = \triangledown$ and $\triangle\alpha\triangle{=}\triangle$.

Complexity is created by the duplication of atomic contractions when they are permuted through other rules [16]. It is our goal to understand this phenomenon in the best possible generality, i.e. to keep track of the creation and duplication of atoms when any contraction rule is permuted downwards in a derivation. However, when permuting contraction rules we may create an unbounded number of other contractive and cocontractive rules.

By observing the subatomic form of known rewriting rules that permute atomic contractions downwards in derivations, a novel way of controlling this phenomenon arises: we will show that it is possible to move 'blocks' of nested contraction rules together, in such a way that we are no longer concerned by the number of (co)contraction rules created by the procedure.

Consider this reduction, corresponding to permuting an atomic contraction through an atomic cut:



In this reduction, we move a block of nested contractions (in red) by creating another block of nested contractions lower in the proof and a block of nested cocontractions (in red as well). The structure that we are therefore interested in studying is that of recursive nestings of contraction rules. For convenience and readability, we will represent these nestings in the form of a hyper-rule named *merge contraction*, which will be defined recursively in order to capture the nested structure.

Definition 12. In a ν-contractive system SA, *a nesting of contractions* is an SA derivation defined recursively as follows:

- A formula $A \nu B$ is a nesting of contractions ;
- A contraction equality rule is a nesting of contractions ;
- A derivation

$$c\,\frac{(A\ \alpha\ B)\ \nu\ (C\ \alpha\ D)}{\boxed{\begin{array}{c}A\ \nu\ C\\ \phi_1 \parallel\\ R\end{array}}\ \alpha\ \boxed{\begin{array}{c}B\ \nu\ D\\ \phi_2 \parallel\\ S\end{array}}}$$

  is a nesting of contractions if c is a contraction and ϕ_1 and ϕ_2 are nestings of contractions.

Definition 13. A *ν-merge* of two formulae is defined as follows:

- $A \nu B$ is a ν-merge of A and B that we call a *trivial merge*;
- u is a ν-merge of u and u, where $u \in \mathscr{U}$ is a constant;
- $C_1\ \alpha\ C_2$ is a ν-merge of $A_1\ \alpha\ A_2$ and $B_1\ \alpha\ B_2$ for $\alpha \in \mathscr{R}$ if C_1 is a ν-merge of A_1 and B_1 and C_2 is a ν-merge of A_2 and B_2. In this case we say that α is the *main relation* of the merge.

If C is a ν-merge of A and B, by an abuse of language we will sometimes refer to the triple (A, B, C) as a ν-merge.

$\overline{\nu}$-merges of two formulae are defined dually.

It can be easily seen that each nesting of contractions corresponds to a ν-merge, and each ν-merge corresponds to a nesting of contractions.

Proposition 14. *Given a nesting of contractions* $\begin{array}{c}A\ \nu\ B\\ \phi \parallel\\ C\end{array}$*, C is a ν-merge of A and B.*

Proposition 15. *If C is a ν-merge of A and B, there is a nesting of contractions* $\begin{array}{c}A\ \nu\ B\\ \parallel\\ C\end{array}$*.*

The duals of the above propositions clearly hold for $\overline{\nu}$-contractions and nestings of cocontractions. Given the above characterisation of nestings as derivations whose conclusion is a ν-merge of its premiss, for ease of notation we will represent nestings as a hyper-rule, that we call merge contraction.

Definition 16. $\mathsf{mc}{\downarrow}\dfrac{A\ \nu\ B}{C}$ is a *merge contraction* if C is a non-trivial ν-merge of A and B.

$\mathsf{mc}{\uparrow}\dfrac{C}{A\ \overline{\nu}\ B}$ is a *merge cocontraction* if C is a non-trivial $\overline{\nu}$-merge of A and B.

For each nesting, we have a merge contraction, and for each merge contraction we have a nesting. We will permute nestings downwards in a derivation by creating other nestings lower in the derivation by, equivalently, permuting merge contractions downwards by creating other merge contractions lower in the derivation.

The main property allowing us to permute merge contractions through other rules is the given in the following proposition:

Proposition 17. *If C is a ν-merge of A and B, we can define* projections $\pi_A \Vdash^{A}_{C}\{=,w\}$ and $\pi_B \Vdash^{B}_{C}\{=,w\}$ associated to the merge.

With the projections associated to a merge as a tool, we will now show reduction rules allowing us to permute merge (co)contractions downwards (upwards) in a proof.

Definition 18 (Reduction rule s)**.** We define the following class of reduction rules:

$$s_\rho:\quad \mathsf{mc}{\downarrow}\dfrac{A\ \nu\ B}{C\left\{\rho\dfrac{M}{N}\right\}}\quad\longrightarrow\quad \mathsf{mc}{\downarrow}\dfrac{\boxed{\overset{A}{\underset{C\left\{\rho\frac{M}{N}\right\}}{\pi_A\Vdash}}}\ \nu\ \boxed{\overset{B}{\underset{C\left\{\rho\frac{M}{N}\right\}}{\pi_B\Vdash}}}}{C\{N\}}$$

where π_A and π_B are the projections associated to the merge (A, B, C).

Definition 19 (Reduction rule t)**.** If the rule $\mu\dfrac{(A\ \nu\ B)\ \beta\ (C\ \overline{\nu}\ D)}{(A\ \beta\ C)\ \nu\ (B\ \beta\ D)}$ is derivable in SA we define the following family of rewriting rules:

$$t_\rho:\quad \rho\dfrac{\boxed{\mathsf{mc}{\downarrow}\dfrac{(A_1\ \alpha\ A_2)\ \nu\ (B_1\ \alpha\ B_2)}{C\ \alpha\ D}}\ \beta\ (E\ \alpha'\ F)}{(C\ \beta\ E)\ \alpha\ (D\ \beta'\ F)}\quad\longrightarrow\quad \mu\dfrac{((A_1\ \alpha\ A_2)\ \nu\ (B_1\ \alpha\ B_2))\ \beta\ \boxed{\mathsf{mc}{\uparrow}\dfrac{E\ \alpha'\ F}{(E\ \alpha'\ F)\ \overline{\nu}\ (E\ \alpha'\ F)}}}{\mathsf{mc}{\downarrow}\dfrac{\boxed{\rho\dfrac{(A_1\ \alpha\ A_2)\ \beta\ (E\ \alpha'\ F)}{(A_1\ \beta\ E)\ \alpha\ (A_2\ \beta'\ F)}}\ \nu\ \boxed{\rho\dfrac{(B_1\ \alpha\ B_2)\ \beta\ (E\ \alpha'\ F)}{(B_1\ \beta\ E)\ \alpha\ (B_2\ \beta'\ F)}}}{(C\ \beta\ E)\ \alpha\ (D\ \beta'\ F)}}$$

where C is a ν-merge of A_1 and B_1, and D is a ν-merge of A_2 and B_2.

In fact, the rewriting systems for classical logic [16] and for multiplicative additive linear logic [18] that allow us to permute *atomic* (co)contractions through other rules are particular instances of the generalised rewriting rules defined above. We have therefore shown that these results are a consequence of a wider phenomenon: both rewriting systems exploit the shape of atomic contractions to be able to permute them with other rules. Furthermore, the termination of these rewriting systems holds for the subatomic versions too: identically to the atomic versions, the subatomic rewriting systems will terminate in the absence of a particular construction, called *ai*-cycle. We will rigorously define and tackle *ai*-cycles in the next section.

Theorem 20. *Rewriting system C' for SAKS (Figure 1) is given by the reduction rules s and t where the merge contraction being permuted has main relation a, and by the dual reductions. C' is terminating on the set of ai-cycle-free derivations.*

Furthermore, by being able to permute generic contractions together, we advance towards proving a full decomposition theorem for classical logic and multiplicative additive linear logic, by being able to confine all contraction rules to the bottom of a proof.

Conjecture 21. *We define rewriting system* D *for* SAKS *as the system given by the general reductions* s*,* t*, and the dual reductions for merge cocontractions. System* D *is weakly normalising.*

4 Cycle elimination

Atomic contractions and atomic cocontractions can be permuted downwards/upwards in a classical logic derivation only in the absence of a particular construction called *ai*-cycle [16]. Cycles are created when two atom occurrences created in the same identity rule are eliminated by the same cut rule. We call these atoms occurrences the *edges* of a cycle. Identically, this result holds for multiplicative additive linear logic [18].

Cycles are straightforwardly removed by cut-elimination. Our goal in this chapter however is to take advantage of the reductions presented in the previous chapter to show that we can remove *ai*-cycles without recurring to cut-elimination, therefore proving the independence of the decomposition and the cut-elimination procedures.

Furthermore, the phenomenon of cycles has been studied in the sequent calculus, where it has been shown that it is possible to remove them through a procedure of quadratic-time complexity [7]. With the procedure we present we hope to be able to study the complexity cost of cycle-elimination in deep inference in future research.

In the sequent calculus, cycles can only occur due to the presence of contractions [6]. Likewise, in our case they exist due to the presence of contraction rules. For an *ai*-cycle to occur in classical logic, two atom occurrences coming from the same introduction rule and therefore related by $\vee$ at the top of the flow have to be connected by $\wedge$ at the bottom of the flow to be eliminated by the same cut rule. Therefore, an instance of a rule that changes the relation between formulae from $\vee$ to $\wedge$ needs to occur, and it must contain the atoms involved in the cycle. The only rule that does so in subatomic system SAKS for classical logic (Figure 1) is the contraction rule m. Likewise, in multiplicative additive linear logic cycles can only occur if there is a contraction rule between the introduction and the cut of the cycle. We call these instances of contraction rules *critical*.

Definition 22. Let ϕ be a derivation containing a cycle. The *critical medial* for this cycle is the lowest instance of a rule

$$\mathsf{m}\,\frac{(A\{a\} \wedge B) \vee (C \wedge D\{\bar{a}\})}{(A\{a\} \vee C) \wedge (B \vee D\{\bar{a}\})}$$

in ϕ where the occurrences of a and $\bar{a}$ are the edes of the cycle.

A *critical merge contraction* is a maximal merge contraction that contains a critical medial.

This novel idea of removing cycles by starting from the 'critical medial' has in fact yielded two methods for the elimination of cycles: the one presented in what follows, and the one presented in [1], that will both be studied to ascertain the complexity cost of each procedure.

The intuition behind our procedure is simple: by using the rewriting rules defined in the previous section we can permute a critical contraction rules downward until it is below the cut of its cycle. In this process derivations are significantly altered: cycles are removed and edges are bifurcated. Termination of the procedure is easy to check: we show that when permuting critical contractions downwards we do not create any additional critical contractions.

Theorem 23. *Let* ϕ *be a derivation with* n *critical merge contractions. Then there exists a derivation* ψ *with the same premiss and conclusion with* $n-1$ *critical merge contractions.*

To eliminate all cycles from a derivation, one simply performs the procedure n times, once for each critical merge contraction.

Corollary 24. *Given a derivation* ϕ*, there exists a derivation* ψ *with the same premiss and conclusion and without cycles.*

References

[1] A. Aler Tubella, A. Guglielmi, and B. Ralph, *Removing cycles from proofs*, in Leibniz International Proceedings in Informatics, vol. 82, 2017, pp. 1–18.

[2] K. Brünnler, *Atomic cut elimination for classical logic*, in Computer Science Logic (CSL), M. Baaz and J. A. Makowsky, eds., vol. 2803 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 86–97.

[3] ——, *Cut elimination inside a deep inference system for classical predicate logic*, Studia Logica, 82 (2006), pp. 51–71.

[4] ——, *Locality for classical logic*, Notre Dame Journal of Formal Logic, 47 (2006), pp. 557–580.

[5] K. Brünnler and A. F. Tiu, *A local system for classical logic*, in Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), R. Nieuwenhuis and A. Voronkov, eds., vol. 2250 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 347–361.

[6] A. Carbone, *Interpolants, cut elimination and flow graphs for the propositional calculus*, Annals of Pure and Applied Logic, 83 (1997), pp. 249–299.

[7] ——, *The cost of a cycle is a square*, The Journal of Symbolic Logic, 67 (2002), pp. 35–61.

[8] J. Gallier, *Constructive logics. Part I: A tutorial on proof systems and typed λ-calculi*, Theoretical Computer Science, 110 (1993), pp. 249–339.

[9] G. Gentzen, *The Collected Papers of Gerhard Gentzen*, North-Holland, 1969.

[10] ——, *Investigations into logical deduction*, in The Collected Papers of Gerhard Gentzen, M. Szabo, ed., North-Holland, Amsterdam, 1969, pp. 68–131.

[11] J.-Y. Girard, *A new constructive logic: classical logic.*, Mathematical Structures in Computer Science, 1 (1991), pp. 255–296.

[12] N. Guenot and L. Strassburger, *Symmetric normalisation for intuitionistic logic*, in Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2014. To appear.

[13] A. Guglielmi, *A system of interaction and structure*, ACM Transactions on Computational Logic, 8 (2007), pp. 1:1–64.

[14] A. Guglielmi, T. Gundersen, and M. Parigot, *A proof calculus which reduces syntactic bureaucracy*, in 21st International Conference on Rewriting Techniques and Applications (RTA), C. Lynch, ed., vol. 6 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010, pp. 135–150.

[15] A. Guglielmi and L. Strassburger, *A system of interaction and structure V: The exponentials and splitting*, Mathematical Structures in Computer Science, 21 (2011), pp. 563–584.

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

[17] L. Roversi, *A deep inference system with a self-dual binder which is complete for linear lambda calculus*, Journal of Logic and Computation, 26 (2016), pp. 677–698.

[18] L. Strassburger, *A local system for linear logic*, in Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), M. Baaz and A. Voronkov, eds., vol. 2514 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 388–402.

[19] ——, *Linear Logic and Noncommutativity in the Calculus of Structures*, PhD thesis, Technische Universität Dresden, 2003.

[20] ——, *MELL in the calculus of structures*, Theoretical Computer Science, 309 (2003), pp. 213–285.

[21] A. Tiu, *A system of interaction and structure II: The need for deep inference*, Logical Methods in Computer Science, 2 (2006), pp. 4:1–24.