A Strictly Linear Subatomic Proof System

Victoria Barrett[1]
Department of Computer Science, University of Bath, UK
Inria Saclay, Palaiseau, France

Alessio Guglielmi
Department of Computer Science, University of Bath, UK

Benjamin Ralph
Department of Computer Science, University of Bath, UK

Abstract

We present a subatomic deep-inference proof system for a conservative extension of propositional classical logic with decision trees that is strictly linear. In a strictly linear subatomic system, a single linear rule shape subsumes not only the structural rules, such as contraction and weakening, but also the unit equality rules. An interpretation map from subatomic logic to propositional classical logic recovers the usual semantics and proof theoretic properties. By using explicit substitutions that indicate the substitution of one derivation into another, we are able to show that the unit-equality inference steps can be eliminated from a subatomic system for propositional classical logic with only a polynomial complexity cost in the size of the derivation, from which it follows that the system p-simulates Frege systems, and we show cut elimination for the resulting strictly linear system.

2012 ACM Subject Classification Theory of computation → Proof theory

Keywords and phrases Deep inference, Open deduction, Subatomic logic, Decision trees, Explicit substitutions, Cut elimination, Proof theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.39

Funding *Victoria Barrett*: This work was supported by the Engineering and Physical Sciences Research Council [EP/R513155/1] and Inria Exploratory Action IMPROOF.

1 Introduction

A change of formalism can provide us with a new lens on the proof theory of familiar logics, allowing for certain proof-theoretic properties that might be unachievable in more established systems [7, 20]. One of the main motivations behind the deep-inference [13] methodology is the pursuit of **locality**, allowing us to check the correctness of inference steps in constant time. Therefore, many deep-inference proof systems consist of inference rules that are either **atomic** or **linear** [5, 6, 8, 9]. Examples of atomic and linear rules for propositional classical logic are, respectively, the **atomic contraction** and the **medial rule**, shown here:

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

One way that locality has been used to benefit the proof theory of classical logic is via the normalisation mechanisms employing **atomic flows** [3, 14, 15], which use the fact that all rules are either atomic or linear to trace the flow of atoms in a derivation. A proof system with atomic structural rules also allows for finer control of compression mechanisms such as contraction, giving fully lazy sharing when translated into the lambda calculus through a Curry-Howard interpretation [16]. However, the notion of linearity used in such proof systems only applies to atoms. In this work, we define and pursue a more extreme form of

[1] Corresponding Author

linearity, **strict linearity**, in which we require not only linearity with respect to the units instead of the atoms in a derivation. By introducing another proof compression mechanism that has been studied in deep-inference settings [12, 19], **explicit substitutions**, into our proof system, we can define a sound and complete proof system with strictly linear rules, while keeping a handle on proof complexity.

To achieve strict linearity, we further develop the **subatomic logic** approach [1, 2], where atoms are treated as non-commutative self-dual connectives whose arguments are their truth values. Subatomic logics necessitate a deep-inference proof system, since cut elimination for logics with non-commutative self-dual connectives are not possible in Gentzen systems [20]. While the syntax of subatomic logic may seem obscure, semantically they can be understood as the integration of binary decision trees into the the standard language of propositional classical logic [4].

When translated into subatomic logic, both linear and non-linear inference rules can be encoded by a common linear shape, called the **subatomic shape**:

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

Deep-inference proof systems using the subatomic shape are able to capture a range of logics, including those that cannot be expressed in a Gentzen formalism such as BV, and characterise their normalisation in a common way across these different logics [1].

Indeed, although translating a deep-inference proof system into subatomic logic results in a larger space of proofs, normalisation is simplified because there is only a single rule shape, and so the number of possible interactions is limited. Therefore, to ensure the preservation of standard proof-theoretic results such as cut elimination we can take a standard, non-subatomic derivation, translate it to subatomic logic to perform the standardised proof theoretic procedure in a system with only a single rule shape, before projecting back to the standard level.

This paper takes the principles underlying subatomic proof theory even further. In previous work in subatomic logic, the structural rules are subsumed by the subatomic rule shape but inference rules obtained from unit equalities are left intact. However, because the interpretation map from subatomic logic to a non-subatomic logic can collapse these unit equalities itself, there is a redundancy here and a natural question arises: can the rules obtained from unit equalities be eliminated from subatomic proof systems? In this paper we show that the answer is positive.

We call a proof system **strictly linear** if it contains only rules of the subatomic shape. Inference rules based on unit equalities such as $\dfrac{A \wedge \mathsf{t}}{A}$ are not strictly linear, because they are not linear in the units. We have two main motivations for studying such systems.

The first concerns normalisation. With only a single rule shape, normalisation procedures can be simplified yet further, and interference caused by unit-equality inference steps can be eliminated.

The second concerns semantics and complexity. Factorising proofs using explicit substitutions allows for the compression of proofs and the elimination of an unnecessary form of "bureaucracy" [19]. More speculatively, this work provides a theoretical foundation for the development of a proof system with explicit substitutions that retains locality, and without having to exclude derivations containing cycles between cuts and identities [3, 18]. Having defined explicit substitutions for strictly linear proofs and begun to explore how they impact complexity and normalisation, we can in the future lift this to the standard, non-subatomic level using the interpretation from subatomic to standard logic.

In summary, the central contribution of this work is to show that a strictly linear system for the subatomic version of propositional classical logic can be obtained. We do so by showing that a subatomic proof with unit-equality rules can be transformed into one with no unit-equality rules but with explicit substitutions, with only a polynomially-bounded increase in the size of the proof. In doing so, this work furthers the tradition of using the compositional freedom offered by deep-inference formalisms to regularise and homogenise proof systems and normalisation procedures, entirely eliminating all structural variety and non-locality from inference rules.

Outline

In Section 2 we define the preliminaries necessary for the paper: we introduce explicit substitutions and describe the ways in which derivations containing them can be composed, we introduce the proof systems that we will use in this work, in particular we introduce subatomic and strictly linear systems for classical logic.

In Section 3 we introduce the technical machinery that we will use to prove the results in the later chapter, in particular, the Eversion Lemma.

In Section 4 we show that the unit-equality inference steps can be eliminated from a subatomic system for propositional logic. From this result, we can obtain a strictly linear system that is complete for classical logic. In particular, we show that, by using eversion and explicit substitutions, we can achieve this with only a polynomial complexity cost in the size of the derivation.

In Section 5 we show that the cut rule can be eliminated from the strictly linear system in a way that is preserved by interpretation to the standard, non-subatomic level.

2 Preliminaries

▶ **Definition 1.** *We have the following mutually disjoint countable sets of atoms, connectives, units and variables:*

$$\mathcal{A} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}, \quad \mathcal{C} = \{\wedge, \vee\}, \quad \mathcal{U} = \{\mathsf{0}, \mathsf{1}\}, \quad \mathcal{V} = \{x, y, z, w, \dots\}$$

The set of ***formulae****,* $\mathcal{F}$ *is defined in the following way:*

$$\mathcal{F} ::= \mathcal{V} \mid \mathcal{U} \mid \mathcal{F}\,\mathcal{A}\,\mathcal{F} \mid \mathcal{F}\,\mathcal{C}\,\mathcal{F} \mid \langle \mathcal{F} | \mathcal{V} \rangle\, \mathcal{F}$$

Note that, since we are working with subatomic logic, atoms are binary connectives rather than atomic formulae. We also can compose formulae by ***explicit substitution****, where* $\langle A|x\rangle\, B$ *denotes the explicit substitution of* A *for a variable* x *in* B*.*

Formulae containing no explicit substitutions are called ***flat formulae****. Given a formula* A*, we write* $\mathsf{fl}\, A$*, the* ***flat expansion*** *of* A*, for the formula where all the explicit substitution terms* $\langle C|x\rangle\, D$ *appearing in* A *are applied, i.e. each instance of the variable* x *in* D *is replaced by* C*, and* $\mathsf{fl}\, A$ *is the (unique) formula so obtained. We denote by* $A \equiv B$ *the syntactic identity of* A *and* B *modulo renaming of variables bound by explicit substitution.*

▶ **Definition 2.** *Let the two operations* ***down-saturation*** *and* ***up-saturation*** *on atoms and connectives be defined as* $\check{\wedge} = \check{\vee} = \vee$*,* $\hat{\wedge} = \hat{\vee} = \wedge$ *and* $\hat{\boldsymbol{a}} = \check{\boldsymbol{a}} = \boldsymbol{a}$ *for each* $\boldsymbol{a} \in \mathcal{A}$*.*

This definition can be extended to formulae by replacing each atom and connective by its up- or down-saturation respectively.

▶ **Definition 3.** *The set of* ***derivations*** $\mathcal{D}$*, denoted by* ϕ*,* ψ*,* χ*,* ω*, . . . , is defined by the grammar:*

$$
\begin{array}{lll}
\mathcal{D} ::= \mathcal{V} \mid \mathcal{U} & & \\
\quad \mid \mathcal{D}\,\mathcal{C}\,\mathcal{D} \mid \mathcal{D}\,\mathcal{A}\,\mathcal{D} & & \textit{composition by } \textbf{\textit{connective}} \textit{ or } \textbf{\textit{atom}}, \\
\quad \left| \boxed{\begin{array}{c}\mathcal{D}\\ \sim\\ \mathcal{D}\end{array}} \right. & & \textit{composition by } \textbf{\textit{expansion}}, \\
\quad \left| \boxed{r\,\dfrac{\mathcal{D}}{\mathcal{D}}} \right. & & \textit{composition by } \textbf{\textit{inference}} \textit{ or } \textbf{\textit{inference step}}, \\
\quad \left| \langle \mathcal{D} | \mathcal{V} \rangle\, \mathcal{D} \right. & & \textit{composition by } \textbf{\textit{explicit substitution}},
\end{array}
$$

We say that $\langle \phi | x \rangle\, \psi$ *is the* ***explicit substitution*** *of* ϕ *into a variable* x *of* ψ*. We define free and bound variables in the usual way: in particular every occurrence of* x *in* ψ *is* ***bound*** *in* $\langle \phi | x \rangle\, \psi$ *and if a variable occurrence is not bound it is* ***free****. We do not consider the substitution variable* x *in* $\langle \phi | x \rangle$ *to be an occurrence of the variable* x *so it is neither free nor bound. We denote by* $\underline{\phi}$ *the set of free variables appearing in the derivation* ϕ*.*

We say that a derivation is ***open*** *if it contains no units (so that its every leaf is a free variable that can be substituted into) and that it is* ***flat*** *if it contains no explicit substitutions. Explicit substitution terms such as* $\langle \phi | x \rangle$ *can be denoted by* π*,* ρ*,* σ*,* τ*, and so on. We may drop parentheses and boxes when there is no ambiguity. We denote by* $\phi \equiv \psi$ *the syntactic identity of* ϕ *and* ψ *modulo renaming of bound variables and associativity of compositions by expansion and inference.*

▶ **Definition 4.** *The* ***size*** $|\phi|$ *of a derivation* ϕ *is the number of occurrences of variables and units appearing in it, not counting the substitution variables in explicit substitution terms, i.e.* $|\langle \phi | x \rangle\, \psi| = |\phi| + |\psi|$*.*

▶ **Definition 5.** *The two maps* ***premise*** *and* ***conclusion****,* $\mathsf{pr}, \mathsf{cn} \colon \mathcal{D} \to \mathcal{F}$ *and the two maps* ***width*** *and* ***height****,* $\mathsf{w}, \mathsf{h} \colon \mathcal{D} \to \mathbb{N}$ *are so defined:*

- *If* $\phi \in \mathcal{F}$*, then* $\mathsf{pr}\,\phi \equiv \mathsf{cn}\,\phi \equiv \phi$ *and* $\mathsf{w}\,\phi = |\phi|$ *and* $\mathsf{h}\,\psi = 0$
- *If* $\phi \equiv \psi\ \alpha\ \chi$*, then*

$$
\begin{array}{ll}
\mathsf{pr}\,\phi \equiv \mathsf{pr}\,\psi\ \alpha\ \mathsf{pr}\,\chi & \mathsf{w}\,\phi = \mathsf{w}\,\psi + \mathsf{w}\,\chi \\
\mathsf{cn}\,\phi \equiv \mathsf{cn}\,\psi\ \alpha\ \mathsf{cn}\,\chi & \mathsf{h}\,\phi = \max(\mathsf{h}\,\psi, \mathsf{h}\,\chi)
\end{array}
$$

- *If* $\phi \equiv \langle \psi | x \rangle\, \chi$*, then*

$$
\begin{array}{ll}
\mathsf{pr}\,\phi \equiv \langle \mathsf{pr}\,\psi | x \rangle\, \mathsf{pr}\,\chi & \mathsf{w}\,\phi = \mathsf{w}\,\psi + \mathsf{w}\,\chi \\
\mathsf{cn}\,\phi \equiv \langle \mathsf{cn}\,\psi | x \rangle\, \mathsf{cn}\,\chi & \mathsf{h}\,\phi = \max(\mathsf{h}\,\psi, \mathsf{h}\,\chi)
\end{array}
$$

- *if* $\phi \equiv \boxed{\dfrac{\psi}{\chi}}$ *or* $\phi \equiv \boxed{\begin{array}{c}\psi\\ \sim\\ \chi\end{array}}$*, then*

$$
\begin{array}{ll}
\mathsf{pr}\,\phi \equiv \mathsf{pr}\,\psi & \mathsf{w}\,\phi = \max(\mathsf{w}\,\psi, \mathsf{w}\,\chi) \\
\mathsf{cn}\,\phi \equiv \mathsf{cn}\,\chi & \mathsf{h}\,\phi = \mathsf{h}\,\psi + \mathsf{h}\,\chi + 1
\end{array}
$$

In Definition 3, we give the definition of a derivation abstracted from any particular proof system, with no correctness criteria given for composition by expansion or inference. However, to be able to identify correct derivations for a specific deep-inference proof system, we need to define what it means to be a correct instance of an inference rule. Furthermore, in order to define proof systems equipped with explicit substitutions, we need to show that the correctness of an instance of composition by expansion $\frac{A}{\sim\!\sim B}$ can be decided in polynomial time on the size of A and B. Proposition 7 adapts Paterson and Wegman's algorithm for linear unification [17] to this problem, in the style of [10].

▶ **Definition 6.** *An **inference rule** is a relation on formulae decidable in polynomial time on the size of its arguments. A **proof system** is a finite set of unit equality and subatomic-linear rules. Given a proof system $\mathcal{S}$, an inference step such that $(\mathsf{cn}\,\psi, \mathsf{pr}\,\chi) \in r \in \mathcal{S}$, for some rule r, is called an **instance of r** and is denoted as $r\,\frac{\psi}{\chi}$.*

*In this paper, we only discuss proof systems with two types of inference rules, **unit-equality rules** and **subatomic rules**. We can therefore specify the unit-equality and subatomic rules of a proof system $\mathcal{S}$ by $\mathcal{S}_{=}$ and $\mathcal{S}_{\mathsf{sa}}$ respectively.*

▶ **Proposition 7.** *Given formulae A and B, the identity $\mathsf{fl}\,A \equiv \mathsf{fl}\,B$ can be decided in linear time with respect to the size of the formulae, by comparing the flat expansions of A and B without actually performing the substitutions.*

Proof. Let us call a **normal representation** of A a formula

$$C \equiv \langle C_n|x_n\rangle \cdots \langle C_1|x_1\rangle\, C_0$$

such that, for $n \geq 0$, $x_1, \ldots, x_n$ are fresh, distinct variables, $C_0, \ldots, C_n$ are flat, $C_0 \notin \{x_1, \ldots, x_n\}$, each of $C_1, \ldots, C_n$ contains one and only one connective and $\mathsf{fl}\,C \equiv \mathsf{fl}\,A$; C can be obtained from A in linear time on $|A|$. Let

$$D \equiv \langle D_m|y_m\rangle \cdots \langle D_1|y_1\rangle\, D_0$$

be a normal representation of B; we can check $\mathsf{fl}\,A \equiv \mathsf{fl}\,B$ by the following recursive procedure invoked as $p(C_0, D_0)$:

Procedure $p(C_i, D_j)$.

1. if $C_i \equiv D_j$, return success;
2. otherwise, if $C_i \equiv x_{i_h} \in \{x_1, \ldots, x_n\}$ and $D_j \equiv y_{j_h} \in \{y_1, \ldots, y_m\}$ and $p(C_{i_h}, D_{j_h})$ succeeds, then return success;
3. otherwise, if $C_i \equiv C_{i_1}\,\alpha\,C_{i_2}$ and $D_j \equiv D_{j_1}\,\alpha\,D_{j_2}$ and, for $h = 1, 2$, $p(C_{i_h}, D_{j_h})$ succeeds, then return success;
4. otherwise, return failure.

To convert the formula A into its normal representation is linear: we transform every subformula $A_1\,\alpha\,A_2$ to $\langle A_1|z\rangle\,\langle A_2|z'\rangle(z\,\alpha\,z')$; the number of such transformations is bounded by the number of connectives in A. To satisfy the constraint that each $C_i, i > 1$ contains exactly one connective, explicit substitutions of the form $\langle A|x\rangle$ for $A \in \mathcal{U} \cup \mathcal{V}$ are applied without increasing the size of the formula.

Where we are comparing formulae $\langle C_i|x_i\rangle\, K\{x_i\}$ and $\langle D_j|y_j\rangle\, K\{y_j\}$, the procedure $p(C_i, D_j)$ need only be performed once, so the comparison is linear on the size of the original formulae. ◀

▶ **Definition 8.** *Given a proof system $\mathcal{S}$, we say that a derivation ϕ in $\mathcal{D}$ is a* ***derivation in $\mathcal{S}$*** *if every inference step in ϕ is an instance of some rule of $\mathcal{S}$ and for each composition by expansion $\overset{\psi}{\underset{\chi}{\sim}}$ we have $\mathsf{fl}\,\mathsf{cn}\,\psi \equiv \mathsf{fl}\,\mathsf{pr}\,\chi$. One way to denote such a derivation is $\phi \Vdash \mathcal{S}$, where $\overset{A}{B}$ A and B are the premise and conclusion of ϕ. We note that by Proposition 7 establishing the correctness of composition by expansion is decidable in linear time.*

▶ **Definition 9.** *We define a set of subatomic rules generated by the following scheme*

$$\widehat{\alpha}\beta\,\frac{(x\,\widehat{\alpha}\,y)\,\beta\,(z\,\alpha\,w)}{(x\,\beta\,z)\,\alpha\,(y\,\beta\,w)} \qquad \beta\widehat{\alpha}\,\frac{(x\,\alpha\,y)\,\beta\,(z\,\widehat{\alpha}\,w)}{(x\,\beta\,z)\,\alpha\,(y\,\beta\,w)}$$

$$\breve{\beta}\alpha\,\frac{(x\,\alpha\,y)\,\beta\,(z\,\alpha\,w)}{(x\,\breve{\beta}\,z)\,\alpha\,(y\,\beta\,w)} \qquad \alpha\breve{\beta}\,\frac{(x\,\alpha\,y)\,\beta\,(z\,\alpha\,w)}{(x\,\beta\,z)\,\alpha\,(y\,\breve{\beta}\,w)}$$

where $\alpha, \beta \in \mathcal{C} \cup \mathcal{A}$. We define the set of subatomic rules KDT *to be every rule generated by this scheme together with the mix rule* $\breve{\wedge}\,\dfrac{A \wedge B}{A \vee B}$.

We define a set of unit-equality rules Keq *as follows:*

$$=_1 \frac{x}{x \vee \mathsf{0}} \qquad =_2 \frac{x \vee \mathsf{0}}{x} \qquad =_3 \frac{x}{\mathsf{0} \vee x} \qquad =_4 \frac{\mathsf{0} \vee x}{x}$$

$$=_5 \frac{\mathsf{0}}{\mathsf{0}\,\alpha\,\mathsf{0}} \qquad =_6 \frac{\mathsf{0}\,\alpha\,\mathsf{0}}{\mathsf{0}} \qquad \textit{where} \qquad \alpha \in \{\wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$$

$$=_7 \frac{x}{x \wedge \mathsf{1}} \qquad =_8 \frac{x \wedge \mathsf{1}}{x} \qquad =_9 \frac{x}{\mathsf{1} \wedge x} \qquad =_{10} \frac{\mathsf{1} \wedge x}{x}$$

$$=_{11} \frac{\mathsf{1}}{\mathsf{1}\,\beta\,\mathsf{1}} \qquad =_{12} \frac{\mathsf{1}\,\beta\,\mathsf{1}}{\mathsf{1}} \qquad \textit{where} \qquad \beta \in \{\vee, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$$

We now define two proof systems by specifying their unit-equality and subatomic rules:

- KDTS*, where* $\mathsf{KDTS}_{=} = \emptyset$ *and* $\mathsf{KDTS}_{\mathsf{sa}} = \mathsf{KDT}$.
- KDTeq*, where* $\mathsf{KDTeq}_{=} = \mathsf{Keq}$ *and* $\mathsf{KDTeq}_{\mathsf{sa}} = \mathsf{KDT}$.

We say that a derivation in KDTS *or* KDTeq *is a* ***proof*** *if its premise is equal to* $\mathsf{1}$ *with respect to the unit equalities given in* Keq.

▶ Remark 10. Is shown in [4] that the system KDTeq employed in a formalism without explicit substitutions is sound and complete for standard non-subatomic propositional classical logic conservatively extended by decision trees.

Therefore, in this paper, we will only work with KDTeq derivations that are flat, i.e. without any explicit substitutions and we will refer to the logic this system corresponds to as subatomic propositional classical logic.

▶ **Definition 11.** *Let ψ and χ be two derivations such that $\mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi$. We define a derivation called the* ***synchronal composition*** *of ψ and χ, denoted as*

$$\boxed{\begin{matrix}\psi \\ \cdots \\ \chi\end{matrix}};$$

we do so by structural induction, as follows:

1. *if* ψ *is a formula, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \chi$*; similarly, if* χ *is a formula, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \psi$*;*
2. *if* $\psi \equiv \alpha(\psi_1, \ldots, \psi_n)$ *and* $\chi \equiv \alpha(\chi_1, \ldots, \chi_n)$*, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \alpha\left(\left[\begin{smallmatrix}\psi_1\\ \cdots\\ \chi_1\end{smallmatrix}\right], \ldots, \left[\begin{smallmatrix}\psi_n\\ \cdots\\ \chi_n\end{smallmatrix}\right]\right)$*;*
3. *if* $\psi \equiv \langle \psi_1 | x \rangle\, \psi_2$ *and* $\chi \equiv \langle \chi_1 | x \rangle\, \chi_2$*, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \left\langle \left[\begin{smallmatrix}\psi_1\\ \cdots\\ \chi_1\end{smallmatrix}\right] \middle| x \right\rangle \left[\begin{smallmatrix}\psi_2\\ \cdots\\ \chi_2\end{smallmatrix}\right]$*;*
4. *if* $\psi \equiv \left[\begin{smallmatrix}\psi_1\\ \sim\\ \psi_2\end{smallmatrix}\right]$*, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \left[\begin{smallmatrix}\psi_1\\ \sim\\ \psi_2\\ \cdots\\ \chi\end{smallmatrix}\right]$*; similarly, if* $\chi \equiv \left[\begin{smallmatrix}\chi_1\\ \sim\\ \chi_2\end{smallmatrix}\right]$*, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \left[\begin{smallmatrix}\psi\\ \cdots\\ \chi_1\\ \sim\\ \chi_2\end{smallmatrix}\right]$*;*
5. *if* $\psi \equiv \left[\begin{smallmatrix}\psi_1\\ \hline\\ \psi_2\end{smallmatrix}\right]$*, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \left[\begin{smallmatrix}\psi_1\\ \hline\\ \psi_2\\ \cdots\\ \chi\end{smallmatrix}\right]$*; similarly, if* $\chi \equiv \left[\begin{smallmatrix}\chi_1\\ \hline\\ \chi_2\end{smallmatrix}\right]$*, then* $\left[\begin{smallmatrix}\psi\\ \cdots\\ \chi\end{smallmatrix}\right] \equiv \left[\begin{smallmatrix}\psi\\ \cdots\\ \chi_1\\ \hline\\ \chi_2\end{smallmatrix}\right]$*.*

▶ **Definition 12.** *A* ***section*** *of a derivation* ϕ *is any formula* A *such that*

$$\phi \equiv \left[\begin{smallmatrix}\psi\\ \cdots\\ A\\ \cdots\\ \chi\end{smallmatrix}\right] \quad ,$$

for some derivations ψ *and* χ*; in the above derivation, each section of* ψ *is said to be* ***above*** *each section of* χ *and each section of* χ *is said to be* ***below*** *each section of* ψ*.*

▶ **Definition 13.** ***Formula contexts*** *are used to indicate formulae with one or more holes, and are denoted* $A\{\,\}\cdots\{\,\}$*, or with other letters as necessary, often* H *and* K*. The holes can be filled by derivations as well as formulae. When unambiguous, we write* $A\{B\}$ *to indicate the formula* A *where the location of its subformula* B *has been singled out.*

▶ **Definition 14.** *We denote an* ***actual substitution*** *that maps* x *to* A *and leaves all other variables unchanged by* $[A|x]$*. Actual substitutions can be applied to derivations and* $[A|x]\,\phi$ *stands for the derivation obtained by replacing every free occurrence of* x *in* ϕ *by the formula* A *and we say that this substitutes* A *for* x *in* ϕ*. In the specific case where we are substituting into a formula, we can extend this notion to allow for the actual substitution of a derivation into a formula, where* $[\psi|x]\,B$ *is obtained by replacing every free occurrence of* x *in the formula* B *by the derivation* ψ*. Note that* $[A|x]\,B\{x\} \equiv B\{A\}$ *if* x *does not appear free in* $B\{\,\}$*.*

We abbreviate the simultaneous actual substitution $[B_1|x_1, \ldots, B_n|x_n]\,A$ *as* $[B_i|x_i]_{1\ldots n}\,A$*. Given a set of variables* $S = \{x_1, \ldots, x_n\}$*, we write* $[B_i|x_i]_S\,A$ *for* $[B_1|x_1, \ldots, B_n|x_n]\,A$*; we might also write* $[B_v|v]_S\,A$ *to stand for* $[B_{x_1}|x_1, \ldots, B_{x_n}|x_n]\,A$*. We denote both individual and simultaneous actual substitutions by* π*,* ρ*,* σ *and* τ*, and so on.*

We extend the conventions on simultaneous actual substitutions to explicit substitutions and derivations. Therefore, we might indicate with $\langle B_v|v\rangle_{\underline{A}}\,\phi$ the substitution $\langle B_{x_1}|x_1, \ldots, B_{x_n}|x_n\rangle\,\phi$, where $\underline{A} = \{x_1, \ldots, x_n\}$ is the set of free variables of A (see Definition 1).

The notation $[B_v|v]_{\underline{A}}A$ is at risk of being ambiguous because the enumeration of the variables is arbitrary. For example, if $\underline{A} = \{v_1, v_2\}$ and $B_1 = v_2$ then $[B_1|v_1]\,[B_2|v_2]\,A \not\equiv [B_2|v_2]\,[B_1|v_1]$. We take care to use this notation only when it is unambiguous and there is no dependency between substitutions.

3 The Merge and Eversion Lemmas

We are now ready to first prove the Merge Lemma, and then its generalisation the Eversion Lemma, which enables the proof of the main results of this paper. As we explained in the introduction, we want to be able to eliminate all the non-linear equality rules from KDTeq to produce a strictly linear proof in KDTS. Using the Eversion Lemma, we are able to transform the derivation on the left into the derivation on the right:

$$\dfrac{\pi\phi}{\pi K\left\{\dfrac{A}{A\vee B}\right\}}\pi\psi \quad\longrightarrow\quad \dfrac{\pi[x\vee B|x]_{\underline{A}}\phi}{\pi K\left\{\begin{array}{l}[x\vee B|x]_{\underline{A}}A\\ \quad\|\\ A\vee\boxed{\begin{array}{l}[B|x]_{\underline{A}}\breve{A}\\ \quad\|\\ \left[\breve{A}^y\middle|y\right]_{\underline{B}}B\end{array}}\end{array}\right\}}\pi\left[\breve{A}^y\middle|y\right]_{\underline{B}}\psi}\,,$$

where we assume that A and B are open formulae and that $\breve{A}^y$ is obtained from A by replacing every variable by y and every connective with its down-saturation. By doing this transformation, the inference step becomes strictly linear. This propagates substitutions up and down the derivation but does not affect its structure.

Before stating and proving the full Eversion Lemma, we state and prove the Merge Lemma, a version of it restricted to a substitution with a single connective or atom.

▶ **Proposition 15** (Merge Lemma). *Let A be an open formula with variables $\{x_1, \ldots, x_n\}$ and let $\beta \in \mathcal{C} \cup \mathcal{A}$. Then there exist* KDTS *derivations:*

$$\begin{array}{c}[y_i\;\beta\;z_i|x_i]_{\underline{A}}A\\ \|\\ [y_i|x_i]_{\underline{A}}A\;\beta\;[z_i|x_i]_{\underline{A}}\breve{A}\end{array}\qquad\qquad\begin{array}{c}[y_i|x_i]_{\underline{A}}A\;\beta\;[z_i|x_i]_{\underline{A}}\widehat{A}\\ \|\\ [y_i\;\beta\;z_i|x_i]_{\underline{A}}A\end{array}\,.$$

The width and height of each derivation are bounded by $2|A|$.

Proof. We consider the derivation on the left and proceed by induction on the structure of A.

- If $A \equiv x_i$ then we take the derivation $y_i\;\beta\;z_i$.
- If $A \equiv C\;\alpha\;D$ then we build:

$$\beta\breve{\alpha}\,\dfrac{\boxed{\begin{array}{c}[y_i\;\beta\;z_i|x_i]_{\underline{C}}C\\ \|\\ [y_i|x_i]_{\underline{C}}C\;\beta\;[z_i|x_i]_{\underline{C}}\breve{C}\end{array}}\;\alpha\;\boxed{\begin{array}{c}[y_i\;\beta\;z_i|x_i]_{\underline{D}}D\\ \|\\ [y_i|x_i]_{\underline{D}}D\;\beta\;[z_i|x_i]_{\underline{D}}\breve{D}\end{array}}}{\left([y_i|x_i]_{\underline{C}}C\;\alpha\;[y_i|x_i]_{\underline{D}}D\right)\;\beta\;\left([z_i|x_i]_{\underline{C}}\breve{C}\;\breve{\alpha}\;[z_i|x_i]_{\underline{D}}\breve{D}\right)}\,.$$

- If $A \equiv \langle C|w\rangle\, D$ then we build:

$$\dfrac{\left\langle \begin{array}{c}[y_i\ \beta\ z_i|x_i]_{\underline{C}}C \\ \| \\ [y_i|x_i]_{\underline{C}}C\ \beta\ [z_i|x_i]_{\underline{C}}\breve{C}\end{array}\middle|\, w\right\rangle [y_i\ \beta\ z_i|x_i]_{\underline{D}}D}{\left\langle [y_i|x_i]_{\underline{C}}C\middle|w'\right\rangle \left\langle [z_i|x_i]_{\underline{C}}\breve{C}\middle|w''\right\rangle \boxed{\begin{array}{c}[y_i\ \beta\ z_i|x_i]_{\underline{D}}\,[w'\ \beta\ w''|w]\,D \\ \| \\ [y_i|x_i]_{\underline{D}}\,[w'|w]\,D\ \beta\ [z_i|x_i]_{\underline{D}}\,[w''|w]\,\breve{D}\end{array}}}{\left\langle [y_i|x_i]_{\underline{C}}C\middle|w\right\rangle [y_i|x_i]_{\underline{D}}D\ \beta\ \left\langle [z_i|x_i]_{\underline{C}}\breve{C}\middle|w\right\rangle [z_i|x_i]_{\underline{D}}\breve{D}}\ .$$

  That is, we perform a merge on C and then, with the first composition by expansion, we replace all occurrences w in D by $w'\ \beta\ w''$, the minimal amount of structure needed to perform the merge on D. The second composition by expansion then rearranges this to the desired structure.

The width of the derivation is at most $2|A|$; note that $\left|[y_i\ \beta\ z_i|x_i]_{\underline{A}}A\right| < 2|A|$ if there are explicit substitutions in A.

For each connective in A there is a corresponding instance of composition by rule in the constructed derivation; and for each explicit substitution in A, there are two corresponding instances of composition by expansion. Therefore the height of the derivation is at most $2|A|$, the worst scenario being a formula only composed of explicit substitutions, each of which is of size 1. ◀

We call the derivations on the left in Proposition 15 **down-merges**, and the derivations on the right **up-merges**.

We can use the Merge Lemma to simulate unit-equality inference steps, without affecting the value or the structure of the rest of the derivation. For example, the unit-equality inference step $=\dfrac{A}{A \wedge 1}$ becomes $\begin{array}{c}[w \wedge 1|w]_{\underline{A}}A \\ \| \\ A \wedge [1|x]_{\underline{A}}\breve{A}\end{array}$. This propagates $w \wedge 1$ upwards through the derivation in place of w, for each variable w occurring in A, and propagates $[1|w]_{\underline{A}}\breve{A}$, which is equal to 1 for any formula A, downwards through the derivation in place of this occurrence of 1.

A naïve approach to eliminating the unit-equality inference steps in this way will blow up the size of the derivation exponentially. To see this, we can consider the following transformation, in which π substitutes a unit onto the variable x and we assume that in each section of ψ, x occurs exactly once, so that the two inference steps shown form a pair:

$$\pi\left[\begin{array}{c}\phi \\ \hdashrule{3em}{0.4pt}{} \\ K\left\{=\dfrac{A}{A\ \alpha\ x}\right\} \\ \hdashrule{3em}{0.4pt}{} \\ \psi \\ \hdashrule{3em}{0.4pt}{} \\ H\left\{=\dfrac{B\ \beta\ x}{B}\right\} \\ \hdashrule{3em}{0.4pt}{} \\ \chi\end{array}\right] \longrightarrow \pi\left[\begin{array}{c}[v\ \beta\ x|v]_{\underline{H\{\ \}}}[w\ \alpha\ x|w]_{\underline{A}}\phi \\ \hdashrule{3em}{0.4pt}{} \\ [v\ \beta\ x|v]_{\underline{H\{\ \}}}K\left\{\begin{array}{c}[w\ \alpha\ x|w]_{\underline{A}}A \\ \| \\ A\ \alpha\ \breve{A}^x\end{array}\right\} \\ \hdashrule{3em}{0.4pt}{} \\ [v\ \beta\ x|v]_{\underline{H\{\ \}}}\left[\breve{A}^x\middle|x\right]\psi \\ \hdashrule{3em}{0.4pt}{} \\ [v\ \beta\ x|v]_{\underline{H\{\ \}}}H\left\{B\ \beta\ \breve{A}^x\right\} \\ \| \\ \boxed{\begin{array}{c}H\{B\} \\ \hdashrule{1em}{0.4pt}{} \\ \chi\end{array}}\ \beta\ [x|v]_{\underline{\breve{H}\{\ \}}}\breve{H}\{\breve{A}^x\}\end{array}\right]$$

Eliminating the two unit-equality inference steps in the way described above would result in a substitution $\left[\hat{B}^x\middle|x\right]$ being propagated up, and a substitution $\left[\check{A}^x\middle|x\right]$ being propagated down (where A^x and B^x stand for the result of substituting the variable x onto every leaf of A and B respectively). Therefore, in order to eliminate both unit-equality inference steps, the entire context around one of them must be duplicated, resulting in something like the derivation above. This doubles the width of the derivation, leading to an exponential blow-up in the size when eliminating all unit-equality inference steps in succession.

By iterating the Merge Lemma in a certain way, we derive the Eversion Lemma and can use this to avoid this exponential blow-up.

▶ **Lemma 16** (Eversion Lemma)**.** *Let A and B be open formulae with their free variables denoted $\underline{A} = \{w_1, \ldots, w_n\}$ and $\underline{B} = \{y_1, \ldots, y_m\}$ respectively. Then there exist* KDTS *derivations:*

$$\begin{array}{c}\left[\hat{B}\middle|w_i\right]_{\underline{A}}\check{A}\\ \| \\ \left[\check{A}^{y_j}\middle|y_j\right]_{\underline{B}}\hat{B}\end{array} \qquad\qquad \begin{array}{c}\left[\hat{B}^{w_i}\middle|w_i\right]_{\underline{A}}\check{A}\\ \| \\ \left[\check{A}\middle|y_j\right]_{\underline{B}}\hat{B}\end{array} ,$$

where $B^{w_i} \equiv [w_i|y_j]_{\underline{B}}B$ and $A^{y_j} \equiv [y_j|w_i]_{\underline{A}}A$. Both the width and the height of these derivations are $O(|A||B|)$.

Proof. We consider the derivation on the left and proceed by induction on the structure of B.

- If $B \equiv y_j$ then we take the derivation $\check{A}^{y_j}$.
- If $B \equiv E\,\beta\,F$ then we build:

$$\boxed{\begin{array}{c}\left[\hat{E}\,\hat{\beta}\,\hat{F}\middle|w_i\right]_{\underline{A}}\check{A}\\ \chi\,\| \\ \boxed{\begin{array}{c}\left[\hat{E}\middle|w_i\right]_{\underline{A}}\check{A}\\ \phi\,\|\\ \left[\check{A}^{y_j}\middle|y_j\right]_{\underline{E}}\hat{E}\end{array}}\;\hat{\beta}\;\boxed{\begin{array}{c}\left[\hat{F}\middle|w_i\right]_{\underline{A}}\check{A}\\ \psi\,\|\\ \left[\check{A}^{y_j}\middle|y_j\right]_{\underline{F}}\hat{F}\end{array}}\end{array}} ,$$

  where the derivations ϕ and ψ are obtained by induction and the derivation χ is obtained via the Merge Lemma 15.
- If $B \equiv \langle E|z\rangle\,F$ then we build:

$$\boxed{\begin{array}{c}\left[\left\langle\hat{E}\middle|z\right\rangle\hat{F}\middle|w_i\right]_{\underline{A}}\check{A}\\ \sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim \\ \left\langle\hat{E}\middle|z\right\rangle\boxed{\begin{array}{c}\left[\hat{F}\middle|w_i\right]_{\underline{A}}\check{A}\\ \phi\,\|\\ \left[\check{A}^{y_j}\middle|y_j\right]_{\underline{F}\setminus\{z\}}\left[\check{A}^{z}\middle|z\right]\hat{F}\end{array}}\\ \sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim \\ \left\langle\boxed{\begin{array}{c}\left[\hat{E}\middle|z\right]\check{A}^z\\ \psi\,\|\\ \left[\check{A}^{z}\middle|y_j\right]_{\underline{E}}\hat{E}\end{array}}\middle|z\right\rangle\left[\check{A}^{y_j}\middle|y_j\right]_{\underline{F}\setminus\{z\}}\hat{F}\end{array}} ,$$

  where the derivations ϕ and ψ are obtained by induction.

The width of the derivations generated in each case are all $O(|A||B|)$. The worst-case scenario for the height is when B is composed by connective, in which case the height increases by at most $2|A|$ in the merge χ. The number of iterations is $O(|B|)$, so the height is $O(|A||B|)$. ◀

We call the derivations in Lemma 16 **eversions**.

▶ **Example 17.** Returning to the example of exponential blow-up above, the pair of unit-equality inference steps in the above example can then be eliminated by the following transformation:

$$\pi\left[\begin{array}{c}\phi\\ \hdashrule{2cm}{0.4pt}{1pt}\\ K\left\{\begin{array}{c}=\\ A\end{array}\frac{A}{A\;\alpha\;x}\right\}\\ \hdashrule{2cm}{0.4pt}{1pt}\\ \psi\\ \hdashrule{2cm}{0.4pt}{1pt}\\ H\left\{\begin{array}{c}=\\ B\end{array}\frac{B\;\beta\;x}{B}\right\}\\ \hdashrule{2cm}{0.4pt}{1pt}\\ \chi\end{array}\right] \longrightarrow \pi\left[\begin{array}{c}\left\langle\widehat{B}^x\middle|x\right\rangle\,[w\;\alpha\;x|w]_{\underline{A}}\phi\\ \hdashrule{3cm}{0.4pt}{1pt}\\ \left\langle\widehat{B}^x\middle|x\right\rangle\,K\left\{\begin{array}{c}[w\;\alpha\;x|w]_{\underline{A}}A\\ \|\\ A\;\beta\;\breve{A}^x\end{array}\right\}\\ \hdashrule{3cm}{0.4pt}{1pt}\\ \left\langle\begin{array}{c|c}\left[\widehat{B}^x\middle|x\right]&\breve{A}\\ \|&\\ \left[\breve{A}^x\middle|x\right]&\widehat{B}\end{array}\middle|x\right\rangle\,\psi\\ \hdashrule{3cm}{0.4pt}{1pt}\\ \left\langle\breve{A}^x\middle|x\right\rangle\,H\left\{\begin{array}{c}B\;\beta\;\widehat{B}^x\\ \|\\ [y\;\beta\;x|y]_{\underline{B}}B\end{array}\right\}\\ \hdashrule{3cm}{0.4pt}{1pt}\\ \left\langle\breve{A}^x\middle|x\right\rangle\,[y\;\beta\;x|y]_{\underline{B}}\chi\end{array}\right].$$

This increases the width of the derivation by $O(|A||B|)$ at the widest point, which is the eversion in the explicit substitution, and increases the height by at least $O(|A|)+O(|B|)$ due to the merges, and in the worst case by $O(|A||B|)$, again due to the eversion.

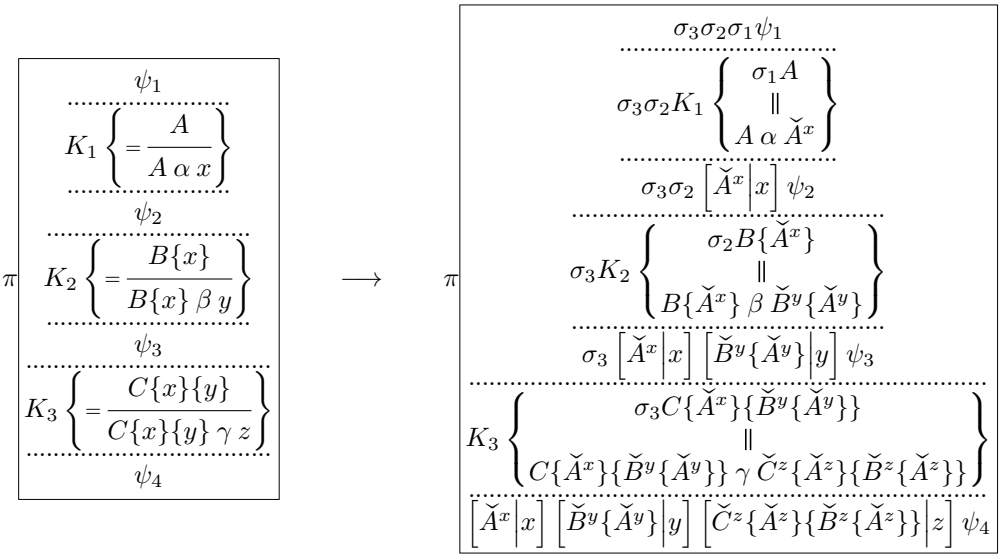
The premise of the transformed derivation is $\pi\left\langle\widehat{B}^x\middle|x\right\rangle\,[w\;\alpha\;x|w]_{\underline{A}}\;\mathsf{pr}\;\phi$. This is equal to the premise $\pi\;\mathsf{pr}\;\phi$ of the original derivation because $x=\widehat{B}^x$ for any formula B and $\pi(w\,\alpha\,x)=\pi w$ for every variable w appearing in A, since $\pi\left[\frac{A}{A\;\alpha\;x}\right]$ is a unit-equality inference step and therefore either $\alpha\in\mathcal{C}$ and πx is the unit of α or $\alpha\in\mathcal{A}$ and $\pi A\equiv\pi x$.

Similarly, the conclusion becomes $\pi\left\langle\breve{A}^x\middle|x\right\rangle\,[y\;\beta\;x|y]_{\underline{B}}\;\mathsf{cn}\;\chi$, and this is equal to the original conclusion $\pi\;\mathsf{cn}\;\chi$.

4 Strict Linearity

Using the Eversion Lemma, we can design a procedure that avoids most causes of exponential blow-up when eliminating the unit-equality rules. However, when performing this elimination we must pay attention to the size of the substitutions that are propagated up or down the derivation. These accumulated substitutions can also lead to an exponential blow-up in the size of the derivation when the elimination is iterated. To observe this, consider the following example:

▶ **Example 18.**



This shows a case where we have three unit-equality inference steps, and the substitutions accumulate as we eliminate them, because the units introduced get into other unit-equality inference steps. Crucially, we see that in the conclusion, z is replaced by $\breve{C}\{\breve{A}^z\}\{\breve{B}^z\{\breve{A}^z\}\}$, which contains two copies of $\breve{A}^z$: one inherited from x occurring in $B\{x\}$ and one from x occurring in $C\{x\}\{y\}$. This pattern will lead to exponential blow-up for the size of the derivation, but it can be controlled with explicit substitutions. In the elimination procedure we describe below, we will factor out the repeated instances of $\breve{A}^z$ and instead replace z by $\left\langle \breve{A}^z \middle| x \right\rangle \left\langle \breve{B}^z\{x\} \middle| y \right\rangle \breve{C}^z\{x\}\{y\}$, which has size $|A| + |B| + |C|$.

We are now ready to state the main theorem of the paper, that we can convert a flat derivation in KDTeq to a derivation in KDTS, i.e. we convert a derivation with unit-equality rules but no explicit substitutions to one that is strictly linear with explicit substitutions, without exponential blow-up in the size of the proof.

▶ **Definition 19.** *Let* $\phi \begin{Vmatrix} A' \\ B' \end{Vmatrix}$ *be a derivation in* KDTeq. *We call* $\pi \left[\begin{smallmatrix} A \\ \| \\ B \end{smallmatrix}\right]$ *a* ***unit factorisation*** *of* ϕ, *if* $A' = \pi A$ *and* $B' = \pi B$, A *and* B *are open formulae in which no variable occurs more than once, and* π *is a simultaneous actual substitution of all the units that occur in* A' *and* B'.

Note that a derivation can have multiple unit factorisations, but for the purposes of this paper it does not matter which we select.

▶ **Theorem 20.** *Let* $\phi \begin{Vmatrix} A' \\ B' \end{Vmatrix}$ *be a flat derivation in* KDTeq *and* $\pi \left[\begin{smallmatrix} A \\ \| \\ B \end{smallmatrix}\right]$ *a unit factorisation of* ϕ. *We can build a derivation* $\phi' \equiv \pi \left[\begin{smallmatrix} \sigma A \\ \| \\ \tau B \end{smallmatrix}\right]$ *in* KDTS *such that the size of* ϕ' *is polynomial in the size of* ϕ *and* $\pi A = \pi\sigma A$ *and* $\pi B = \pi\tau B$.

Proof (Sketch, full proof is in Appendix A). We eliminate the unit-equality inference steps in two phases. In the first phase, we eliminate all those unit-equality inference steps that propagate a unit downwards through a derivation, those that are labelled by $=_1$, $=_3$, $=_5$, $=_7$, $=_9$ or $=_{11}$; in the second we eliminate the others, which propagate a unit upwards through a derivation.

In the first phase, we replace the inference steps by down-merges, taking care to factor out the accumulating substitutions as described in Example 18, and propagating the resulting substitutions through the derivation.

In the second phase, we replace the remaining inference steps by up-merges, again factoring out the accumulating substitutions, and we reconcile the propagated structures via eversions, as described in Example 17, to obtain the strictly linear derivation ϕ'.

Let w be the width of ϕ and let h be its height. The number of unit-equality inference steps eliminated in each phase is bounded by wh. After the first phase, the maximum width occurs before factoring out the accumulated substitutions and is $O(w^3h^2)$; the height is similarly $O(w^3h^2)$ as this is the most that it increases by for any step. After the second phase, the maximum width again occurs before factoring out the accumulated substitutions and is $O(w^7h^5)$; again this represents the greatest increase of height in any step. Therefore the height and width of ϕ' are each $O(w^7h^5)$. ◀

This result shows that the strictly linear system KDTS is complete for propositional classical logic. In addition, it follows from results shown in [2] and [5] that KDTS p-simulates Frege systems [11].

5 Cut Elimination

We now consider the normalisation of strictly linear proofs by showing a cut elimination procedure. As we mention in the introduction, we are motivated to develop a theory of strict linearity because this combines a theoretical foundation for explicit substitutions with simple normalisation procedures. We would like to be able to take a proof in any standard system (not necessarily subatomic), translate it to a strictly linear system, normalise inside that system, and then project back to the original system without too much difficulty.

We do this by applying the method from [4] of eliminating cuts in subatomic logic via projections, adapting it to be strictly linear.

▶ **Definition 21.** *A **cut on** $\mathbf{a}$ in* KDTeq *is any instance of the rule*

$$\wedge\hat{\mathbf{a}}\,\frac{(A\,\mathbf{a}\,B)\wedge(C\,\mathbf{a}\,D)}{(A\wedge C)\,\mathbf{a}\,(B\wedge D)}$$

such that $A = \mathsf{0} = D$ and $B = \mathsf{1} = C$, or $A = \mathsf{1} = D$ and $B = \mathsf{0} = C$. In the system KDTS*, we take explicit substitutions in the context into account, and so a **cut on** $\mathbf{a}$ is any subderivation $K\left\{\wedge\hat{\mathbf{a}}\,\dfrac{A}{B}\right\}$ inside a derivation such that $\mathsf{fl}\,K\{A\}$ and $\mathsf{fl}\,K\{B\}$ when vertically composed form a cut on $\mathbf{a}$.*

We restrict the procedure that we define here to those proofs that do not exhibit too much nesting of atoms inside themselves; this is sufficient to capture a translation of any proof in the standard deep inference system for propositional classical logic, SKS.

▶ **Definition 22.** *Given a derivation ϕ and an atom $\mathbf{a}$, we say that $\mathbf{a}$ is **unnested** in ϕ if there is no section whose flat expansion is of the form $K\{H\{A\,\mathbf{a}\,B\}\,\mathbf{a}\,C\}$ or $K\{A\,\mathbf{a}\,\{H\{B\,\mathbf{a}\,C\}\}$.*

▶ **Definition 23.** *For a derivation ϕ in* KDTS *and an atom $\boldsymbol{a}$ that is unnested in ϕ, we define the* ***left-projection on $\boldsymbol{a}$*** *of ϕ, written $\mathsf{l}_{\boldsymbol{a}}\,\phi$, as follows:*

- *If $\phi \in \mathcal{V} \cup \mathcal{U}$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \phi$.*
- *If $\phi \equiv \psi\ \boldsymbol{a}\ \chi$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \psi$.*
- *If $\phi \equiv \psi\ \beta\ \chi$ for $\beta \not\equiv \boldsymbol{a}$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \mathsf{l}_{\boldsymbol{a}}\,\psi\ \beta\ \mathsf{l}_{\boldsymbol{a}}\,\chi$.*
- *If $\phi \equiv \langle \psi | x \rangle\ \chi$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \langle \mathsf{l}_{\boldsymbol{a}}\,\psi | x \rangle\ \mathsf{l}_{\boldsymbol{a}}\,\chi$.*
- *If $\phi \equiv \boxed{\dfrac{\psi}{\sim\!\!\sim\ \chi}}$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \boxed{\dfrac{\mathsf{l}_{\boldsymbol{a}}\,\psi}{\sim\!\!\sim\ \mathsf{l}_{\boldsymbol{a}}\,\chi}}$.*
- *If $\phi \equiv \boxed{\breve{\wedge}_{\boldsymbol{a}}\dfrac{\dfrac{\psi}{\vdots}}{\dfrac{(A\ \boldsymbol{a}\ B) \wedge (C\ \boldsymbol{a}\ D)}{\vdots}}}{\chi}$ or $\phi \equiv \boxed{\hat{\vee}_{\boldsymbol{a}}\dfrac{\dfrac{\psi}{\vdots}}{\dfrac{(A \wedge C)\ \boldsymbol{a}\ (B \vee D)}{\vdots}}}{\chi}$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \boxed{\breve{\wedge}\dfrac{\dfrac{\mathsf{l}_{\boldsymbol{a}}\,\psi}{\vdots}}{\dfrac{\mathsf{l}_{\boldsymbol{a}}\,A \wedge \mathsf{l}_{\boldsymbol{a}}\,C}{\vdots}}}{\mathsf{l}_{\boldsymbol{a}}\,\chi}$.*
- *If $\phi \equiv \boxed{\breve{\boldsymbol{a}}\beta\dfrac{\dfrac{\psi}{\vdots}}{\dfrac{(A\ \beta\ B)\ \boldsymbol{a}\ (C\ \beta\ D)}{\vdots}}}{\chi}$ or $\phi \equiv \boxed{\hat{\boldsymbol{a}}\beta\dfrac{\dfrac{\psi}{\vdots}}{\dfrac{(A\ \boldsymbol{a}\ C)\ \beta\ (B\ \boldsymbol{a}\ D)}{\vdots}}}{\chi}$ then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \boxed{\dfrac{\mathsf{l}_{\boldsymbol{a}}\,\psi}{\vdots}}{\mathsf{l}_{\boldsymbol{a}}\,\chi}$, and similarly for $\boldsymbol{a}\breve{\wedge}$ and $\boldsymbol{a}\hat{\vee}$. Note that here β cannot be $\boldsymbol{a}$ due to the assumption that $\boldsymbol{a}$ is unnested in ϕ.*
- *If $\phi \equiv \boxed{r\dfrac{\psi}{\chi}}$ in any other case, then $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \boxed{r\dfrac{\mathsf{l}_{\boldsymbol{a}}\,\psi}{\mathsf{l}_{\boldsymbol{a}}\,\chi}}$; note here again that r cannot be $\boldsymbol{a}\breve{\boldsymbol{a}}$ due to the assumption that $\boldsymbol{a}$ is unnested in ϕ.*

The ***right-projection on $\boldsymbol{a}$*** *is denoted by $\mathsf{r}_{\boldsymbol{a}}\,\phi$ and defined in exactly the same way, except for the following case:*

- *If $\phi \equiv \psi\ \boldsymbol{a}\ \chi$ then $\mathsf{r}_{\boldsymbol{a}}\,\phi \equiv \chi$.*

▶ Remark 24. For any atom $\boldsymbol{a}$ that is unnested in a derivation $\phi \in$ KDTS, $\mathsf{l}_{\boldsymbol{a}}\,\phi$ and $\mathsf{r}_{\boldsymbol{a}}\,\phi$ are uniquely determined. Note that it is not the case that $\mathsf{l}_{\boldsymbol{a}}$ and $\mathsf{r}_{\boldsymbol{a}}$ commute: for example $\mathsf{l}_{\boldsymbol{a}}\,\mathsf{r}_{\boldsymbol{a}}(0\ \boldsymbol{a}\ 1) \equiv \mathsf{l}_{\boldsymbol{a}}\,1 \equiv 1$ and $\mathsf{r}_{\boldsymbol{a}}\,\mathsf{l}_{\boldsymbol{a}}(0\ \boldsymbol{a}\ 1) \equiv \mathsf{r}_{\boldsymbol{a}}\,0 \equiv 0$.

▶ Remark 25. For any derivation ϕ in KDTS and any atom $\boldsymbol{a}$ that is unnested in ϕ, the projected derivations $\mathsf{l}_{\boldsymbol{a}}\,\phi$ and $\mathsf{r}_{\boldsymbol{a}}\,\phi$ contain no occurrences of $\boldsymbol{a}$, and so neither contains any cuts on $\boldsymbol{a}$.

It can be the case that eliminating the unit-equality inference steps from a derivation ψ in KDTeq in which $\boldsymbol{a}$ is unnested can create nesting of this atom. This occurs when a derivation contains a pair of unit-equality inference steps $=\dfrac{A\{w\ \boldsymbol{a}\ x\}}{A\{w\ \boldsymbol{a}\ x\}\ \alpha\ u}$ and $=\dfrac{B\{y\ \boldsymbol{a}\ z\}\ \beta\ u}{B\{y\ \boldsymbol{a}\ z\}}$, so that $A\{w\ \boldsymbol{a}\ x\}$ and $B\{y\ \boldsymbol{a}\ z\}$ both contain the atom $\boldsymbol{a}$.

The merge constructions by which we simulate the unit-equality inference steps propagate upwards a substitution $\left\langle \breve{A}^{u}\{u\ \boldsymbol{a}\ u\} \middle| u \right\rangle$ and downwards a substitution $\left\langle \breve{B}^{u}\{u\ \boldsymbol{a}\ u\} \middle| u \right\rangle$. These are resolved by an eversion, which produces an inference step $\boldsymbol{a}\breve{\boldsymbol{a}}\dfrac{(u\ \boldsymbol{a}\ u)\ \boldsymbol{a}\ (u\ \boldsymbol{a}\ u)}{(u\ \boldsymbol{a}\ u)\ \boldsymbol{a}\ (u\ \boldsymbol{a}\ u)}$. That is to say, we create the logical material of $(u\ \boldsymbol{a}\ u)$ twice and substitute one copy into the other. Therefore we define a slightly looser notion of nestedness that captures derivations produced in this way.

▶ **Definition 26.** *Given a derivation ϕ and an atom $\boldsymbol{a}$, we say that $\boldsymbol{a}$ is **shallowly nested** in ϕ if there is no section whose flat expansion is of the form $K\{H\{L\{A\,\boldsymbol{a}\,B\}\,\boldsymbol{a}\,C\}\,\boldsymbol{a}\,D\}$ or similar, and every instance of the inference rule $\boldsymbol{a}\breve{\boldsymbol{a}}$ is of the form $\boldsymbol{a}\breve{\boldsymbol{a}}\dfrac{(A\,\boldsymbol{a}\,A)\,\boldsymbol{a}\,(A\,\boldsymbol{a}\,A)}{(A\,\boldsymbol{a}\,A)\,\boldsymbol{a}\,(A\,\boldsymbol{a}\,A)}$, for every atom $\boldsymbol{a}\in\mathcal{A}$.*

We can extend the definition of left- and right-projection on $\boldsymbol{a}$ to derivations in which $\boldsymbol{a}$ is shallowly nested as follows:

- *If* $\phi \equiv \breve{\boldsymbol{a}}\beta\dfrac{\dfrac{\psi}{(A\,\boldsymbol{a}\,A)\,\boldsymbol{a}\,(A\,\boldsymbol{a}\,A)}}{\chi}$ *then* $\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \dfrac{\mathsf{l}_{\boldsymbol{a}}\,\psi}{A\,\boldsymbol{a}\,A}\Big/\mathsf{l}_{\boldsymbol{a}}\,\chi$ *and* $\mathsf{r}_{\boldsymbol{a}}\,\phi \equiv \dfrac{\mathsf{r}_{\boldsymbol{a}}\,\psi}{A\,\boldsymbol{a}\,A}\Big/\mathsf{r}_{\boldsymbol{a}}\,\chi$.

If an atom is unnested in a derivation ϕ, then that atom will be either unnested or shallowly nested in a derivation produced by eliminating the unit-equality steps from ϕ via the construction given in Section 4.

▶ Remark 27. For any derivation ϕ and any atom $\boldsymbol{a}$ that is shallowly nested in ϕ, the projected derivations $\mathsf{l}_{\boldsymbol{a}}\,\mathsf{l}_{\boldsymbol{a}}\,\phi$, $\mathsf{r}_{\boldsymbol{a}}\,\mathsf{l}_{\boldsymbol{a}}\,\phi$, $\mathsf{l}_{\boldsymbol{a}}\,\mathsf{r}_{\boldsymbol{a}}\,\phi$, and $\mathsf{r}_{\boldsymbol{a}}\,\mathsf{r}_{\boldsymbol{a}}\,\phi$ contain no occurrences of $\boldsymbol{a}$, and so none contains any cuts on $\boldsymbol{a}$. If $A = 1$ then $\mathsf{l}_{\boldsymbol{a}}\,A\,\boldsymbol{a}\,\mathsf{r}_{\boldsymbol{a}}\,A = 1$ for any formula A and any atom $\boldsymbol{a}$.

▶ **Proposition 28.** *For every open formula A in which every variable occurs exactly once, and every atom $\boldsymbol{a}$, there exists a cut-free derivation in* KDTS

$$\chi \equiv \dfrac{\mathsf{l}_{\boldsymbol{a}}\,A\,\boldsymbol{a}\,\mathsf{r}_{\boldsymbol{a}}\,A}{[v\,\boldsymbol{a}\,v|v]_V\,A}\quad,$$

for the set of variables $V = \underline{A} \setminus (\underline{\mathsf{l}_{\boldsymbol{a}}\,A} \cup \underline{\mathsf{r}_{\boldsymbol{a}}\,A})$.

Proof. The construction follows the same structure as the Merge Lemma. Cut-freeness follows from the fact that all inference rules will be of the form $\alpha\breve{\boldsymbol{a}}$, for each connective α in A. ◀

▶ **Theorem 29** (Cut Elimination). *For every* KDTS *proof ϕ in which each atom is either unnested or shallowly nested and whose unit factorisation is $\pi\dfrac{A}{B}$, we can build a cut-free proof of $\pi\sigma B$ such that $\pi\sigma B = \pi B$.*

Proof. We enumerate the atoms in ϕ on which there is a cut $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, let $A_0 \equiv A$, $B_0 \equiv B$, and $\phi_0 \equiv \dfrac{A}{B}$. We then build $\dfrac{A_i}{\phi_i\,\|\,B_i}$ from $\dfrac{A_{i-1}}{\phi_{i-1}\,\|\,B_{i-1}}$ by eliminating any cuts on $\boldsymbol{a}_i$ via the following constructions:

If $\boldsymbol{a}_i$ is unnested in ϕ_{i-1} then we build:

$$\phi_i \equiv \dfrac{\mathsf{l}_{\boldsymbol{a}_i}\,\phi_{i-1}\,\boldsymbol{a}_i\,\mathsf{r}_{\boldsymbol{a}_i}\,\phi_{i-1}}{[v\,\boldsymbol{a}_i\,v|v]_{V_i}\,B_{i-1}}$$

where χ_i is the cut-free derivation given by Proposition 28.

If $\boldsymbol{a}_i$ is shallowly nested in ϕ_{i-1} then we build:

$$\phi_i \equiv \dfrac{\dfrac{\mathsf{l}_{\boldsymbol{a}_i}\,\mathsf{l}_{\boldsymbol{a}_i}\,\phi_{i-1}\;\boldsymbol{a}_i\;\mathsf{r}_{\boldsymbol{a}_i}\,\mathsf{l}_{\boldsymbol{a}_i}\,\phi_{i-1}}{\begin{array}{c}\chi_i' \,\|\\ [v\;\boldsymbol{a}_i\;v|v]_{V_i'}\;\mathsf{l}_{\boldsymbol{a}_i}\,B_{i-1}\end{array}}\,\boldsymbol{a}_i\,\dfrac{\mathsf{l}_{\boldsymbol{a}_i}\,\mathsf{r}_{\boldsymbol{a}_i}\,\phi_{i-1}\;\boldsymbol{a}_i\;\mathsf{r}_{\boldsymbol{a}_i}\,\mathsf{r}_{\boldsymbol{a}_i}\,\phi_{i-1}}{\begin{array}{c}\chi_i'' \,\|\\ [v\;\boldsymbol{a}_i\;v|v]_{V_i''}\;\mathsf{r}_{\boldsymbol{a}_i}\,B_{i-1}\end{array}}}{\begin{array}{c}\chi_i \,\|\\ [v\;\boldsymbol{a}_i\;v|v]_{V_i}\;[v\;\boldsymbol{a}_i\;v|v]_{V_i'}\;[v\;\boldsymbol{a}_i\;v|v]_{V_i''}\;B_{i-1}\end{array}}$$

where χ_i, χ_i', and χ_i'' are the cut-free derivations given by Proposition 28.

Then $\pi\phi_n$ contains no cuts on any atom and $\pi\,\mathsf{cn}\,\phi_n \equiv \pi\sigma B$ where σ is a substitution that does not change the value of the formula.

If a free variable appears anywhere in a proof it must also appear in its premise. However, a formula with a free variable cannot be equal to $\mathsf{1}$, therefore a proof in KDTS cannot contain any free variable and we have that $\pi\,\mathsf{pr}\,\phi_n = \mathsf{1}$. ◀

6 Conclusion

We have shown that a strictly linear subatomic system for propositional classical logic can be obtained by eliminating all unit-equality rules and controlling the complexity using explicit substitutions. Furthermore, we have shown that this strictly linear systems allows for a straightforward cut elimination procedure.

Although we do not define a non-subatomic proof system for propositional classical logic, using the interpretation map given in [2], we can construct a corresponding proof in KDTS from a non-subatomic proof and then by Theorem 29, we can eliminate the cuts from that proof to obtain a cut-free proof that can then be translated, preserving cut-freeness, back into the non-subatomic system.

One of the strengths of subatomic logic is that it can describe normalisation procedures that apply to a wide range of logics. However, in this paper we focus almost entirely on propositional classical logic. This is because our primary investigation is into the complexity of strictly linear proof systems with explicit substitutions, and by working in classical logic we are able to compare against the benchmark systems of Frege and substitution Frege [11]. It is nevertheless our intention that these ideas be extended to a wider range of logics, including first- and higher-order logics.

References

1 Andrea Aler Tubella. *A Study of Normalisation Through Subatomic Logic*. PhD thesis, University of Bath, 2017. URL: `https://people.bath.ac.uk/ag248/aat/phd.pdf`.

2 Andrea Aler Tubella and Alessio Guglielmi. Subatomic proof systems: Splittable systems. *ACM Transactions on Computational Logic*, 19(1):5:1–33, 2018. `doi:10.1145/3173544`.

3 Andrea Aler Tubella, Alessio Guglielmi, and Benjamin Ralph. Removing cycles from proofs. In Valentin Goranko and Mads Dam, editors, *26th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 82 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPIcs.CSL.2017.9`.

4 Chris Barrett and Alessio Guglielmi. A subatomic proof system for decision trees. *ACM Transactions on Computational Logic*, 23(4):26:1–25, 2022. `doi:10.1145/3545116`.

5 Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):14:1–34, 2009. `doi:10.1145/1462179.1462186`.

6 Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. Quasipolynomial normalisation in deep inference via atomic flows and threshold formulae. *Logical Methods in Computer Science*, 12(1):5:1–30, 2016. doi:10.2168/LMCS-12(2:5)2016.

7 Kai Brünnler. Two restrictions on contraction. *Logic Journal of the IGPL*, 11(5):525–529, 2003. doi:10.1093/jigpal/11.5.525.

8 Kai Brünnler. Locality for classical logic. *Notre Dame Journal of Formal Logic*, 47(4):557–580, 2006. doi:10.1305/ndjfl/1168352668.

9 Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer, 2001. doi:10.1007/3-540-45653-8_24.

10 Andrea Condoluci, Beniamino Accattoli, and Claudio Sacerdoti Coen. Sharing equality is linear. In *Proceedings of the 21st International Symposium on Principles and Practice of Declarative Programming*, PPDP '19, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3354166.3354174.

11 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979. doi:10.2307/2273702.

12 Alessio Guglielmi. Formalism B, 2004. URL: https://people.bath.ac.uk/ag248/p/AG13.pdf.

13 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1:1–64, 2007. doi:10.1145/1182613.1182614.

14 Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1):9:1–36, 2008. doi:10.2168/LMCS-4(1:9)2008.

15 Alessio Guglielmi, Tom Gundersen, and Lutz Straßburger. Breaking paths in atomic flows for classical logic. In Jean-Pierre Jouannaud, editor, *25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 284–293. IEEE, 2010. doi:10.1109/LICS.2010.12.

16 Tom Gundersen, Willem Heijltjes, and Michel Parigot. Atomic lambda calculus: A typed lambda-calculus with explicit sharing. In Orna Kupferman, editor, *28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 311–320. IEEE, 2013. doi:10.1109/LICS.2013.37.

17 M.S. Paterson and M.N. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16(2):158–167, 1978. doi:10.1016/0022-0000(78)90043-0.

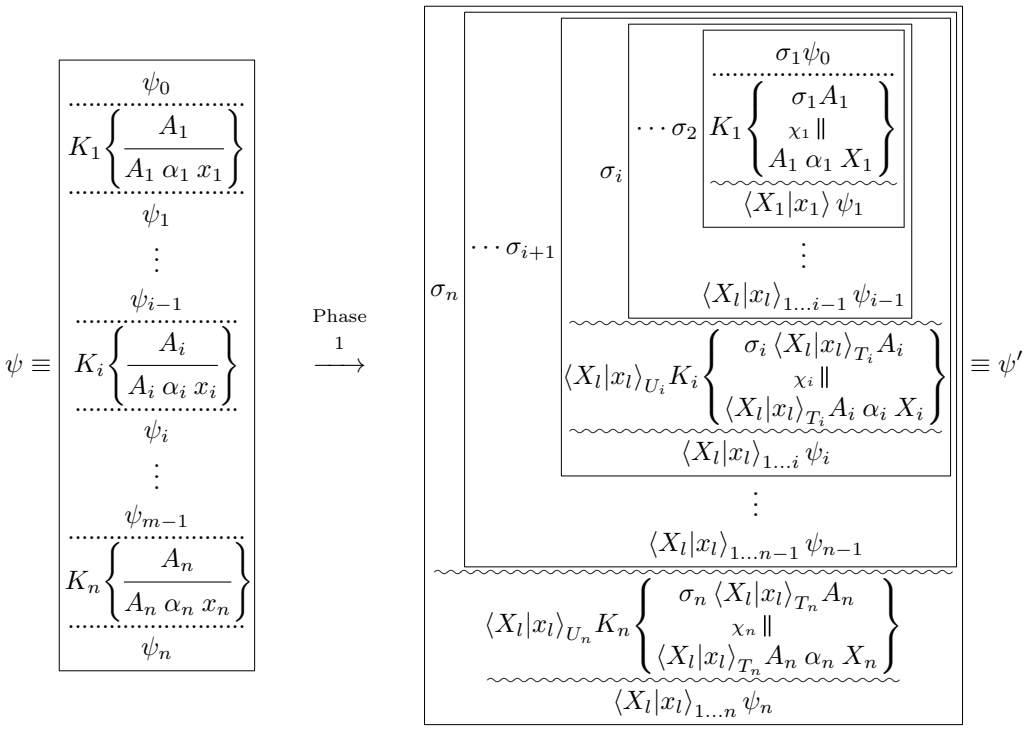
18 Alessio Santamaria. *Towards a Godement calculus for dinatural transformations*. PhD thesis, University of Bath, Somerset, UK, 2019. URL: https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.787523.

19 Lutz Straßburger. From deep inference to proof nets via cut elimination. *Journal of Logic and Computation*, 21(4):589–624, 2011. doi:10.1093/logcom/exp047.

20 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. doi:10.2168/LMCS-2(2:4)2006.

A Omitted proofs

Proof of Theorem 20. We refer to Figures 1 and 2. Given a derivation ϕ that contains inference steps in System Keq, we extract all the units into a substitution π, i.e. we obtain a derivation ψ such that $\phi \equiv \pi\psi$, where π is an actual substitution and ψ is open, i.e. it does not contain units. We assume that different occurrences of a unit or variable in each section of ϕ are assigned by π to different variables, and all variables so created are fresh. Moreover, π is such that all the inference steps in ψ except for those in System Keq remain valid, i.e. corresponding units and variables in the premise and the conclusion of a step are assigned the same variable. To be valid, each equality step of ϕ in Keq needs at least one unit that does not appear either in the premise or the conclusion, therefore ψ is not necessarily a derivation in KDTeq.



Where:

$$\sigma_i = [v\ \alpha_i\ x_i|v]_{\underline{A_i}}$$

$$\breve{A}_i^C \equiv [C|v]_{\underline{A_i}\setminus\{x_1,\ldots,x_{i-1}\}}\ \breve{A}_i$$

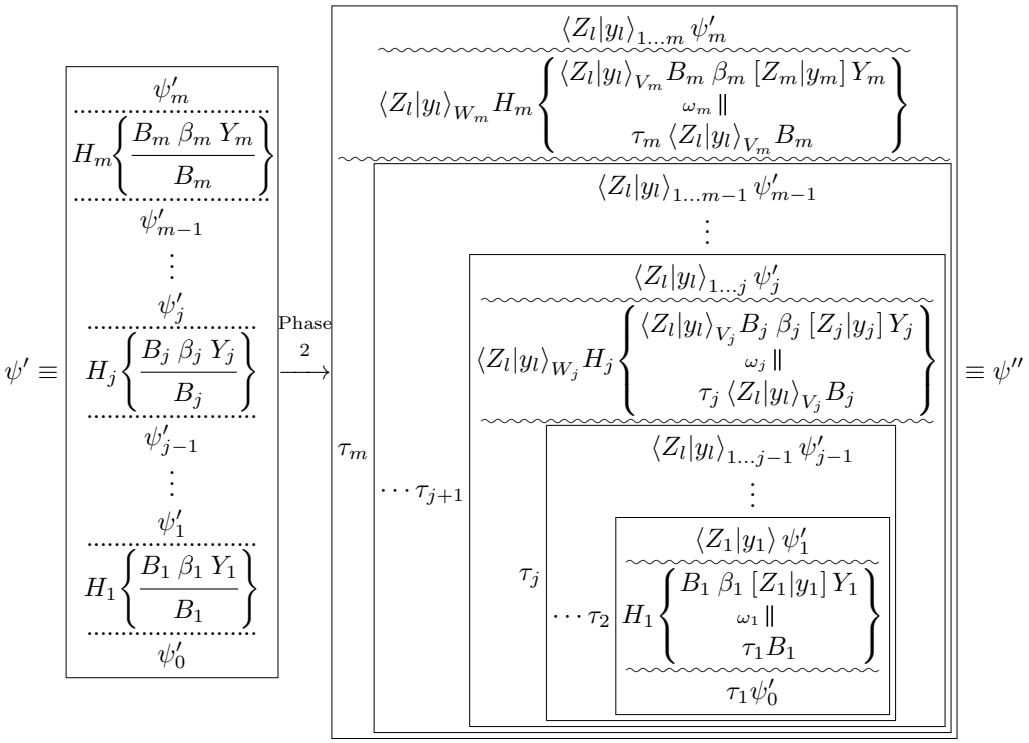
$$X_i \equiv \left\langle \breve{A}_1^{x_i} \middle| x_1 \right\rangle \cdots \left\langle \breve{A}_{i-1}^{x_i} \middle| x_{i-1} \right\rangle \breve{A}_i^{x_i}$$

$$T_i = \{x_1,\ldots,x_{i-1}\} \cap \underline{A_i}$$

$$U_i = \{x_1,\ldots,x_{i-1}\} \setminus \underline{A_i}$$

χ_i is given in Figure 3.

Figure 1 Phase 1 of the construction in Theorem 20.



Where:

$$
\begin{array}{lcl}
\underline{Y_j} & = & \{y_j\} \\
\tau_j & = & \langle Y_j|y_j\rangle\ [v\ \beta_j\ y_j|v]_{\underline{B_j}} \\
\widehat{B}_j^C & \equiv & [C|v]_{\underline{B_j}\setminus\{y_1,\ldots,y_{j-1}\}}\ \widehat{B}_j \\
Z_j & \equiv & \left\langle \widehat{B}_1^{y_j} \middle| y_1 \right\rangle \cdots \left\langle \widehat{B}_{j-1}^{y_j} \middle| y_{j-1} \right\rangle \widehat{B}_j^{y_j} \\
V_j & = & \{y_1, \ldots, y_{j-1}\} \cap \underline{B_j} \\
W_j & = & \{y_1, \ldots, y_{j-1}\} \setminus \underline{B_j} \\
\omega_j & & \text{is given in Figure 3.}
\end{array}
$$

Figure 2 Phase 2 of the construction in Theorem 20.

We first consider the equality steps that are instances of $=_1$, $=_3$, $=_5$, $=_7$, $=_9$ or $=_{11}$, as given in the definition of Keq; that is, those unit-equality inference steps which create a unit travelling downwards in the derivation. Let $x_1, \ldots, x_n$ be the variables in ψ that correspond to one of the units in those steps, via π. For rules $=_5$ and $=_{11}$, there are two choices and we pick one at random. Figure 1 shows $x_1, \ldots, x_n$ to the right of the α_is but we assume that they might be to the left, without prejudice to this proof. Without loss of generality, we assume that the sections of ψ containing the invalid inference steps are arranged as in the figure. Under the assumptions on π mentioned above, no variable x_i appears in formulae A_1, $\ldots$, A_i, for $1 \leq i \leq n$; on the other hand, x_i might appear in $A_{i+1}, \ldots, A_n$.

We build $\phi' \equiv \pi\psi''$, where ψ'' is obtained from ψ in two phases. Phase 1 and Phase 2 perform similar operations on all the invalid inference steps of ψ: in Phase 1 we fix some of them via down-merges and in Phase 2 we fix the remaining ones via up-merges. Both phases produce substitutions that are propagated through the derivation. Some of these substitutions might conflict; indeed, consider the following situation:

$$\psi \equiv \pi \left[\begin{array}{c} \psi_1 \\ \hdashline K\left\{=\dfrac{A_i}{A_i\ \alpha_i\ x_i}\right\} \\ \hdashline \psi_2 \\ \hdashline H\left\{=\dfrac{B_j\ \beta_j\ x_i}{B_j}\right\} \\ \hdashline \psi_3 \end{array}\right] .$$

Here, x_i would be assigned an instance of $\breve{A}_i$ for a down-merge at the top and an instance of $\widehat{B}_j$ for an up-merge at the bottom. By Lemma 16, these conflicting substitutions can be reconciled via the eversion construction

$$\begin{array}{l} \left[\widehat{B}_j \middle| v\right]_{\underline{A_i}} \breve{A}_i \\ \qquad \| \\ \left[\breve{A}_i \middle| v\right]_{\underline{B_j}} \widehat{B}_j \end{array} .$$

This eversion is implemented in Phase 2 (although it could have been implemented in Phase 1).

Phase 1. Each invalid inference step is replaced by an KDTS derivation χ_i, for $1 \leq i \leq n$, shown in Figure 3. Each variable x_i is replaced by a formula X_i, whose purpose is to make a down-merge of A_i via A_i and α_i possible. X_i is constituted by the formula $\breve{A}_i$ whose variables are to be replaced by formulae only containing the variable x_i. The idea is that the original variable x_i is expanded into a formula, X_i, whose structure matches the surroundings (to be amenable to a merge) but whose value remains that of x_i. Those variables of $\breve{A}_i$ that are not in $\{x_1, \ldots, x_{i-1}\}$ are set to x_i, in $\breve{A}_i^{x_i}$. The other variables of $\breve{A}_i$ must be replaced by substitutions that could match the formulae generated by the $\chi_1, \ldots, \chi_{i-1}$ above χ_1 in the derivation; those formulae are $\breve{A}_1^{x_1}, \ldots, \breve{A}_{i-1}^{x_{i-1}}$ and are matched by $\breve{A}_1^{x_i}, \ldots, \breve{A}_{i-1}^{x_i}$. At its top, χ_i generates the substitution σ_i, which does not change the value of the variables it applies to, and is propagated upwards in the derivation. At its bottom, χ_i generates the substitution $\langle X_i | x_i \rangle$, which is propagated downwards in the derivation and also does not change values because $\pi X_i = \pi x_i$. The rest of the construction in Figures 1 and 2 is bookkeeping, mainly relying on having maximally renamed apart all variables so that we can move substitutions without capturing any.

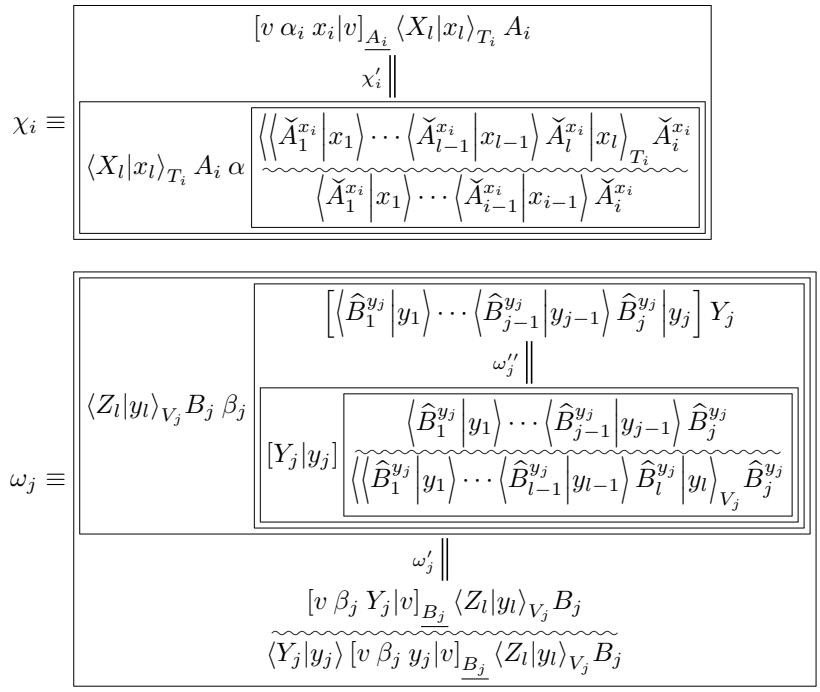


Figure 3 Auxiliary derivations for Phases 1 and 2 in Theorem 20.

Phase 2. Let us call ψ' the derivation produced in Phase 1. We operate on it in a similar way to Phase 1 but in the other direction. The equality steps to fix are those labelled $=_2$, $=_4$, $=_6$, $=_8$, $=_{10}$ and $=_{12}$ in the definition of Keq; that is, those unit-equality inference steps that create a unit travelling upwards in the derivation. For $1 \leq j \leq m$, B_j takes the place of A_i and Y_j that of x_i. One difference is that now Y_j might be one of the formulae X_is, and not just a variable. That said, each Y_j still only contains one variable (potentially in multiple copies), say y_j, and we note that y_j does not appear in $B_1, \ldots, B_j$ and might appear in $B_{j+1}, \ldots, B_m$. In Phase 2, each formula Z_j plays the same role as X_i in Phase 1, and the derivation ω_j, shown in Figure 3, plays the same role as χ_i. There, ω'_j is an up-merge and ω''_j the eversion that we outlined above in this proof. The substitution τ_j is propagated below ω_j; unlike σ_i, τ_j contains an additional substitution $\langle Y_j | y_j \rangle$ but for the rest its role is similar. The result of Phase 2 is a derivation ψ'' in KDTS.

Complexity. We establish upper bounds for the width and height of ψ'. The width of ϕ, say w, dominates the size of $A_1, \ldots, A_n$, and its height, say h, is such that wh dominates n and m. The maximum section width w' of ψ' occurs in the conclusion of some down-merge χ'_i, let us say χ'_n (see also Lemma 15). Therefore,

$$\begin{aligned} w' = \mathsf{w}\,\psi' &\leq \left|\langle X_l | x_l \rangle_{U_n} K_n\{\ \}\right| + 2\left|\langle X_l | x_l \rangle_{T_n} A_n\right| \\ &\leq |K_n\{\ \}| + 2\left|\left\langle\left\langle \breve{A}_1^{x_l} \middle| x_1 \right\rangle \cdots \left\langle \breve{A}_{l-1}^{x_l} \middle| x_{l-1} \right\rangle \breve{A}_l^{x_l} \middle| x_l \right\rangle_{1\ldots n-1} A_n\right| \\ &\leq w + 2(w + 2w + \cdots + (n-1)w + w) \\ &= O(w^3 h^2) \quad . \end{aligned}$$

Because of Lemma 15, the height of χ'_n also is $O(w^3h^2)$, therefore the height h' of ψ' is $O(w^3h^2)$. Similarly, the maximum section width w'' of ψ'' occurs in the premise of some up-merge ω'_j, let us say ω'_m. Therefore,

$$\begin{aligned}
w'' = \mathsf{w}\,\psi'' &\leq \left|\langle Z_l|y_l\rangle_{W_m} H_m\{\ \}\right| + \left|\langle Z_l|y_l\rangle_{V_m} B_m\right| + \left|[Y_m|v]_{\underline{B_m}} \langle Z_l|y_l\rangle_{V_m} \widehat{B}_m\right| \\
&\leq |H_m\{\ \}| + (1+|Y_m|)\left|\left\langle\left\langle \breve{B}_1^{y_l}\middle|y_1\right\rangle \cdots \left\langle \breve{B}_{l-1}^{y_l}\middle|y_{l-1}\right\rangle \breve{B}_l^{y_l}\middle|x_l\right\rangle_{1\ldots m-1} B_m\right| \\
&\leq w' + (1+w')(w' + 2w' + \cdots + (m-1)w' + w') \\
&= O((w')^2(wh)^2) \\
&= O(w^7h^5) \quad ,
\end{aligned}$$

and this is the width of ϕ'. Because of Lemmas 15 and 16, the height of ω_j is also $O(w^7h^5)$, which dominates h', therefore the height of ϕ' is $O(w^7h^5)$. ◀