A Strictly Linear Proof System for Propositional Classical Logic

Barrett, Victoria

Award date:
2024

Awarding institution:
University of Bath

A Strictly Linear Proof System for Propositional Classical Logic

Victoria Barrett

A thesis submitted for the degree of Doctor of Philosophy

University of Bath
Department of Computer Science

July 2024

Abstract

We present a proof system for a conservative extension of propositional classical logic with decision trees that is strictly linear. This means that not only there are no structural rules such as contraction and weakening but there are no rules for unit equalities either, and there is no negation. Yet, its classical semantics and proof-theoretic properties can be recovered via an interpretation map at a polynomial cost. Moreover, this system can p-simulate substitution Frege. Those results are made possible primarily by two technical advances: 1) an 'Eversion Lemma', that guarantees extreme flexibility in manipulating formulae to match a given logical context, and 2) a form of explicit substitution for derivations into derivations. We argue that this proof system represents a significant step towards a notion of factorisation for proofs. That will hopefully lead us to a semantics of proofs adequate to solve the proof identity problem.

Acknowledgements

First, and most of all, I would like to thank my supervisor, Alessio Guglielmi. He introduced me to deep inference, and gave me all the time and support and pastéis de nata I needed over the course of this PhD. I have had so much fun working on these ideas with him and I wish that he were able to see the finished work.

I would also like to thank my examiners, Guy McCusker and Lutz Straßburger. I hope that you agree that your comments and questions in the viva have improved this work significantly; I found them invaluable and I'm very grateful for your time and expertise.

There are also so many members, previous and current, of the Math Found group in Bath to whom I am grateful, and members of the community from further afield. I will mention only a few: Paola Bruscoli, who has been like family, for all of the advice and help that you have given me; Alessio Santamaria, for being such a good friend to me when I first started and helping me to find my feet; and Ben Ralph, for helping so much with the final push.

I would like to thank my family, who have been a constant source of support, encouragement, and belief. Finally I would like to thank my partner, Andy, who is always an inspiration to me. Without your LaTeXtroubleshooting, this thesis would be hand-written.

Declarations

Copyright notice

Declaration of any previous submission of the work

The material presented here for examination for the award of a research degree by research has not been incorporated into a submission for another degree

Declaration of authorship

I am the author of this thesis, and the work described therein was carried out by myself personally, in collaboration with my supervisor Alessio Guglielmi.

Contents

Chapter 1

Introduction

This work belongs to the area of structural proof theory called *deep inference* [18], and in particular to the recently introduced research on *subatomic proof systems* [2].

1.1 Deep Inference

Deep inference is a methodology for the design of proof formalisms; that is, for the design of the language of proofs and how they can be composed. Proof formalisms within the deep inference methodology are distinct from Gentzen formalisms, such as natural deduction and the sequent calculus, in which proofs are represented as trees of formulae where rules operate only on the main connective of a formula, and proofs can be composed horizontally only by conjunction and vertically by rule.

By contrast in deep inference formalisms, we can apply rules at any depth inside a formula. In the formalism of *open deduction* [20], which is the primary formalism used in this thesis, this is equivalent to allowing the free composition of proofs by any connective, extending the horizontal mechanism for composition by connective from formulae to proofs.

In this way, open deduction factors out the order in which rules are applied where this doesn't matter to the essence of the proof; what is called *bureaucracy of type A* in [34]. For example, suppose that we have an open deduction derivation as follows:

$$\begin{array}{c} A \quad C \\ \psi \| \ \star \ \chi \| \\ B \quad D \end{array} \quad ;$$

that is, a derivation ψ of B from a premise A and a derivation χ of D from a premise C, composed by a binary logical connective $\star$. This forms a derivation with premise $A \star C$ and conclusion $B \star D$. The derivations ψ and χ do not interfere with one another and so we can represent them as occurring in parallel rather than sequentially. In a Gentzen system for classical logic, this would be impossible when $\star$ is a disjunction.

This is one of the main goals of deep inference: to obtain a good notion of semantics of proofs, by progressively abstracting away syntactic differences and identifying equivalence classes of proofs which do not differ in the essentials but only in so-called 'bureaucracy'.

Another type of bureaucracy, which is called *type* B in [34], is one of the problems addressed by this thesis. An example of this is the following. Suppose that we have a derivation $\begin{array}{c} K\{x\} \\ \psi(x) \| \\ H\{x\} \end{array}$ and an inference rule $r \dfrac{A}{B}$. From these we can obtain the following two derivations:

$$\begin{array}{c} K\left\{ r \dfrac{A}{B} \right\} \\ \psi(B) \| \\ H\{B\} \end{array} \qquad \begin{array}{c} K\{A\} \\ \psi(A) \| \\ H\left\{ r \dfrac{A}{B} \right\} \end{array} \quad ,$$

which differ only in whether the inference step $r \dfrac{A}{B}$ occurs at the top or the bottom of the derivation.

Deep inference is also motivated by the pursuit of locality: inference steps in a derivation should be small and quick to check. We therefore want rules in deep inference systems to be either atomic or linear, such as the atomic contraction and the medial rule, shown here:

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

To check the validity of an occurrence of the atomic contraction, it suffices to check that the three atoms match. To check the validity of an occurrence

of the medial, we need only to check that the six connectives are in the correct configuration, not look inside the formulae (given a suitable representation of the proof system).

It is the medial rule which allows us to decompose structural rules such as the contraction to their atomic form, as shown by Brünnler and Tiu in [10]. For example, a contraction on a formula $A \wedge B$ can be decomposed progressively as follows:

$$\mathsf{c}\,\frac{(A \wedge B) \vee (A \wedge B)}{A \wedge B} \qquad \rightsquigarrow \qquad \mathsf{m}\,\frac{(A \wedge B) \vee (A \wedge B)}{\boxed{\mathsf{c}\,\frac{A \vee A}{A}} \wedge \boxed{\mathsf{c}\,\frac{B \vee B}{B}}} \quad .$$

While the medial rule could be introduced to a Gentzen system as a sound rule, Brünnler has shown that it could not be used to this same effect there [8].

The locality of rules in deep inference systems allows for good normalisation properties. We want to be able to use simple procedures to transform derivations to a normal form, and we want the derivations to have clear induction measures by which this can done.

One example of a normalisation procedure is *cut elimination*. A cut is a rule which breaks the subformula property: logical material appears in its premise which does not appear in its conclusion. The cut rule can create arbitrarily large amounts of material inside a proof, which complicates proof search. In Gentzen systems, cuts are eliminated by permuting the steps one through another, to move the cuts to the top of the proof where they can be more easily eliminated [16]. Whenever a cut is permuted through a contraction, everything is duplicated. In deep inference, the presence of linear rules such as the medial rule allows us to give cuts and contractions atomically, so that when we permute a cut through a contraction we are only duplicating the smallest amount that we need to duplicate.

In deep inference, the locality of rules allows us to decompose normalisation procedures into phases in a way not possible in Gentzen systems [30]. Deep inference also achieves cut elimination in quasipolynomial time [24, 7], achieving an exponential speed-up over Gentzen systems.

1.2 Subatomic Logic

Recent research into *subatomic logic* by Guglielmi and Aler Tubella has shown that by pursuing of locality even further, we can obtain standard proof-theoretic results, such as cut elimination, in proof systems with linear structural rules, and then interpret those results into standard logics [2, 3]. This can be done because many common inference rules, linear and non-linear, can be encoded by a common linear shape at the subatomic level, which we call the *subatomic shape*:

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

subject to a condition on the logical strength of the connectives α and β.

In subatomic logic, atoms are treated as non-commutative self-dual connectives whose arguments are their truth values; it has been shown by Tiu in [37] that deep inference is the only way to obtain cut elimination for such connectives in linear proof systems. Atoms are not distinguished from other connectives at the subatomic level: inference rules can apply to formulae in their scope.

An interpretation map C into classical logic can recover an atomic contraction and an atomic cut from subatomic logic as follows:

$$\frac{(\mathsf{0}\,\boldsymbol{a}\,\mathsf{1})\vee(\mathsf{0}\,\boldsymbol{a}\,\mathsf{1})}{(\mathsf{0}\vee\mathsf{0})\,\boldsymbol{a}\,(\mathsf{1}\vee\mathsf{1})}\mapsto_{\mathsf{C}}\frac{a\vee a}{a}\qquad\qquad\frac{(\mathsf{1}\,\boldsymbol{a}\,\mathsf{0})\wedge(\mathsf{0}\,\boldsymbol{a}\,\mathsf{1})}{(\mathsf{1}\wedge\mathsf{0})\,\boldsymbol{a}\,(\mathsf{0}\wedge\mathsf{1})}\mapsto_{\mathsf{C}}\frac{\bar{a}\wedge a}{\mathsf{f}}\quad.$$

Subatomic logic does not have negation; this is recovered by the interpretation map, by which the formulae $\mathsf{0}\,\boldsymbol{a}\,\mathsf{1}$ and $\mathsf{1}\,\boldsymbol{a}\,\mathsf{0}$ are interpreted as a and $\bar{a}$ respectively. In addition, the interpretation map collapses formulae according to unit equalities, so that $(\mathsf{1}\wedge\mathsf{0})\,\boldsymbol{a}\,(\mathsf{0}\wedge\mathsf{1})$ becomes $\mathsf{0}\,\boldsymbol{a}\,\mathsf{0}$, which is interpreted as the false unit f of classical logic.

The subatomic shape is able to capture a range of logics, including those which cannot be expressed in a Gentzen formalism such as BV, and characterise their normalisation in a general way, which is explored in [2]. Indeed, although translating a 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 normalise a standard, non-subatomic derivation, we can translate it

to subatomic logic and perform normalisation in that environment, before projecting back to the standard level.

This thesis 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 can apply these unit equalities as rewriting rules, 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 thesis we show that the answer is positive.

1.3 Strict Linearity

We call a proof system *strictly linear* if it contains only rules of the subatomic shape. Inference rules based on unit equalities such as $A \wedge \mathtt{t} = A$ are not strictly linear, because they are not linear in the units. We have two 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. In open deduction, derivations can be composed vertically by rule and horizontally by connective. These two mechanisms are both additive: the size of a derivation is the sum of the sizes of its constituent parts. We want to introduce a multiplicative mechanism for composition, based on explicit substitution. Factorising proofs into explicit substitutions will compress proofs and help to factor out bureaucracy of type B.

For example, if we have a proof $\phi(\psi, \psi, \psi)$, where ψ is a repeated subproof inside ϕ, we would like to be able to factorise this as $\langle \psi | x \rangle \, \phi(x, x, x)$. The substitution is a part of the formalism, and so the proof so obtained only contains one copy of ψ instead of three. We would like the notion of substitution to allow for modular normalisation of derivations, so that this also benefits the complexity of normalisation procedures. For example we would like to be able to normalise ψ inside the explicit substitution, normalise $\phi(x, x, x)$, and then apply the substitution to obtain the same result as if we had normalised $\phi(\psi, \psi, \psi)$.

We would also like to identify the two derivations

$$K\left\{\begin{array}{c} A \\ \hline B \end{array} r\right\} \qquad\qquad \begin{array}{c} K\{A\} \\ \psi(A)\,\| \\ H\left\{\begin{array}{c} A \\ \hline B \end{array} r\right\} \end{array}$$
$$\psi(B)\,\| \\ H\{B\}$$

shown in the example of bureaucracy of type B by factoring the rule into an explicit substitution, so that

$$\left\langle \begin{array}{c} A \\ \hline B \end{array} r \middle| x \right\rangle \begin{array}{c} K\{x\} \\ \psi(x)\,\| \\ H\{x\} \end{array}$$

is a representative of an equivalence class containing both of those derivations.

How precisely we could achieve these goals is a long-standing open problem in deep inference. It is understood that a derivation can be substituted into a cut or an identity by bending it around as shown here:

$$\left\langle \begin{array}{r} A \\ \psi_1\,\| \\ C \\ \psi_2\,\| \\ B \end{array} \middle| x \right\rangle \boxed{\begin{array}{c} \mathsf{t} \\ \hline x \vee \bar{x} \end{array} \mathsf{i}} \qquad \longmapsto \qquad \boxed{\begin{array}{c} \mathsf{t} \\ \hline \boxed{\begin{array}{r} C \\ \psi_2\,\| \\ B \end{array}} \vee \boxed{\begin{array}{r} \bar{C} \\ \bar{\psi}_1\,\| \\ \bar{A} \end{array}} \end{array} \mathsf{i}}$$

Similarly a derivation can be substituted into a contraction or cocontraction by spreading it through as shown here:

$$\left\langle \begin{array}{r} A \\ \psi_1\,\| \\ C \\ \psi_2\,\| \\ B \end{array} \middle| x \right\rangle \boxed{\begin{array}{c} x \vee x \\ \hline x \end{array} \mathsf{c}} \qquad \longmapsto \qquad \boxed{\begin{array}{c} \boxed{\begin{array}{r} A \\ \psi_1\,\| \\ C \end{array}} \vee \boxed{\begin{array}{r} A \\ \psi_1\,\| \\ C \end{array}} \\ \hline \boxed{\begin{array}{r} C \\ \psi_2\,\| \\ B \end{array}} \end{array} \mathsf{c}}$$

This is the intuition on which we want to base the theory of substitutions in deep inference, but difficulties arise because derivations can contain cycles between cuts and identities, as studied in [4]. It is not obvious how to define the substitution of a derivation into a cycle, in particular because these two

intuitions can lead to a conflict. This problem has been investigated by Santamaria in [31].

We could address this problem by defining the formalism for explicit substitutions such that while formulae can be substituted into cycles, derivations cannot be. We prefer not to do this, because such a restriction would mean that whenever we compose two derivations we must check whether we are creating a cycle, which sacrifices locality.

A strictly linear proof system can provide a theoretical foundation for the development of explicit substitutions without having to exclude those derivations which contain cycles. There is no duplication or negation of what we substitute into, and proofs at the standard level can be recovered from them. We can therefore first define explicit substitutions for strictly linear proofs and explore how they impact complexity and normalisation, and then in the future lift this to the standard, non-subatomic level using the interpretation from subatomic to standard logic.

We intend to develop the notion of explicit substitution in an incremental way; indeed, in this thesis we offer two definitions, the second of which subsumes the first. We do this because we want to limit other sources of complexity, such as the time taken to decide equality of two formulae in the presence of such substitutions; increasing the expressive power of explicit substitutions then increases the complexity of the decision procedure.

One of the strengths of subatomic logic is that it can describe normalisation procedures which apply to a wide range of logics. However, in this thesis 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 [15]. It is nevertheless our intention that these ideas be extended to a wider range of logics, including first- and higher-order logics.

1.4 Outline

In Chapters 2 and 3 we define the preliminaries necessary for the thesis.

- In Chapter 2, we introduce explicit substitutions and describe the ways in which derivations containing them can be composed. We do not commit to any proof system in particular in this section, because we

treat explicit substitutions as a composition mechanism belonging to the formalism and independent of any proof system.

- In Chapter 3 we introduce the proof systems which we will use in this work; in particular we introduce subatomic and strictly linear systems for classical logic. We introduce several properties of proof systems which will be of interest in the thesis, and give proof systems which vary only in these properties. This will allow us to compare the effect on complexity of eliminating unit-equality inference steps and introducing explicit substitutions.

In Chapters 4 and 5 we introduce the technical machinery which we will use to prove the results in the later chapters.

- In Chapter 4 we define a map which will allow us to abstract away properties of proofs in the varying systems for classical logic so that we can compare their essential structure and define when two proofs are 'structurally equivalent'. We aim to define this map in such a way that it will commute with normalisation procedures.

- In Chapter 5 we prove the technical lemma, called the Eversion Lemma, which enables us to prove the results in Chapters 6 and 8. The subatomic shape requires that the premise and conclusion follow a particular structure; and this lemma shows that we are able to mimic such structures in order to apply the subatomic rule.

In Chapters 6, 7 and 8 we prove three theorems which serve as tests for our notion of a strictly linear proof system with explicit substitutions. The first two tests concern complexity and the third concerns normalisation.

- In Chapter 6 we show that the unit-equality inference steps can be eliminated from a subatomic system for propositional logic, obtaining a strictly linear system which is complete for classical logic, without changing the structure of the derivation as defined in Chapter 4. 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 Chapter 7 we show that a strictly linear system within a formalism with explicit substitutions is powerful enough to p-simulate substitution Frege. In order to do this, we use a more expressive notion of

explicit substitution than we need in the previous chapter, resulting in a natural overall translation scheme.

- In Chapter 8 we show that the cut rule can be eliminated from the strictly linear system in a way which is preserved by interpretation to a more abstract subatomic system or the standard, non-subatomic level. In particular we show that proofs obtained from the standard level admit a very simple cut elimination procedure.

Chapter 2

Formalisms

In [20], pre-derivations are introduced as a logic-independent way to reason about the composition of derivations. The composition of pre-derivations does not rely on instances of composition by rule belonging to some proof system, because pre-derivations exist at the level of the formalism, prior to the selection of any proof system.

In this section, we extend the definition of pre-derivation to include explicit substitutions. We do this in the formalism, independent of any proof system, because we view the substitutions primarily as a composition mechanism and not as a rule. In order to reason generally about the composition of derivations with substitutions, we introduce first the definition of pre-derivation without any check for correctness, obtained only by the free composition by substitutions and the expansions of substitutions. We then fix a notion of correctness for these compositions.

We borrow the term 'explicit substitution' from work in the lambda calculus [1, 25]. In this thesis we do not use any technical results from this existing body of work, but it is possible that in extending our formalism beyond the case that we study here we will discover more overlap between these theories.

2.1 Pre-derivations

Definition 2.1.1. We assume that we are given the following mutually disjoint countable sets:

- *atoms* $\mathcal{A}$, denoted by a, b, c, $\ldots$;

- *connectives* $\mathcal{C} = \bigcup_1^n \mathcal{C}_i$, denoted by α, β, γ, ..., and other symbols;
- *units* $\mathcal{U}$, denoted by u_α, u_β, u_γ, ..., and other symbols.

We also have the following countable set, disjoint from $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{U}$:

- *variables* $\mathcal{V}$, denoted by x, y, z, w, ...

The set of *pre-derivations* $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$, denoted by ϕ, ψ, χ, ω, ..., is defined by the grammar:

$$
\begin{array}{lll}
\mathcal{P} ::= \mathcal{V} \mid \mathcal{A} \mid \mathcal{U} & & \\
\quad \Big| \; \mathcal{C}_1(\mathcal{P}) \mid \cdots \mid \mathcal{C}_n(\underbrace{\mathcal{P},\ldots,\mathcal{P}}_{n}) \mid \cdots & & \textit{composition by connective,} \\
\quad \mid \langle \mathcal{P} | \mathcal{V} \rangle \, \mathcal{P} & & \textit{composition by (explicit) substitution,} \\
\quad \Big| \; \boxed{\begin{array}{c} \mathcal{P} \\ \sim \\ \mathcal{P} \end{array}} & & \textit{composition by expansion,} \\
\quad \Big| \; \boxed{\begin{array}{c} \mathcal{P} \\ \hline \mathcal{P} \end{array}} & & \textit{composition by inference} \text{ or } \textit{inference step,}
\end{array}
$$

where $\mathcal{P}$ stands for $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$. 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, and we may assume that it is different from all free variables of ϕ in $\langle \phi | x \rangle \, \psi$. We denote by $\underline{\phi}$ the set of free variables appearing in the pre-derivation ϕ.

We say that a pre-derivation is *open* if it contains no units or atoms (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. A *(pre-derivation) context* $\phi\{\,\}\cdots\{\,\}$ is a pre-derivation in which some pre-derivations are substituted by holes; $\phi\{\psi_1\}\cdots\{\psi_n\}$ denotes an pre-derivation where the n holes in $\phi\{\,\}\cdots\{\,\}$ have been filled with ψ_1, ..., ψ_n. Connectives in $\mathcal{C}_2$ are usually written infix, for example $\alpha(\phi,\psi)$ is written $\phi \, \alpha \, \psi$. Connectives in $\mathcal{C}_1$ are called *unary*, those in $\mathcal{C}_2$ *binary*, and so on. Explicit substitution terms

such as $\langle \phi | x \rangle$ can be denoted by π, ρ, σ, τ, and so on. The empty substitution is denoted by ϵ. 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.

Example 2.1.2. For $a, b \in \mathcal{A}$, $\mathsf{u} \in \mathcal{U}$, $w, x, y, z \in \mathcal{V}$ and $\wedge, \vee \in \mathcal{C}_2$, the following constructions, ϕ_1, ϕ_2, ϕ_3 are pre-derivations, and the construction $\phi_4\{\ \}\{\ \}$ is a pre-derivation context:

$$\phi_1 \equiv \boxed{\dfrac{\boxed{\dfrac{a}{a \vee x}} \wedge \left\langle \dfrac{b \wedge \mathsf{u}}{b} \middle| y \right\rangle (y \wedge y)}{\dfrac{(a \vee x) \wedge (b \wedge b)}{(a \vee b) \vee (x \vee b)}}} \qquad \phi_2 \equiv \boxed{\dfrac{\boxed{\dfrac{w}{w \vee x}} \wedge \left\langle \dfrac{y \wedge z}{z} \middle| y \right\rangle (y \wedge y)}{\dfrac{(w \vee x) \wedge (z \wedge z)}{(w \vee z) \vee (x \vee z)}}}$$

$$\phi_3 \equiv \boxed{\dfrac{\boxed{\dfrac{a}{a \vee x}} \wedge \boxed{\dfrac{b}{b \vee c}}}{(a \vee b) \vee (x \vee c)}} \qquad \phi_4\{\ \}\{\ \} \equiv \boxed{\dfrac{\{\ \} \wedge \left\langle \dfrac{y}{y \vee z} \middle| y \right\rangle y}{\dfrac{(w \vee x) \wedge (y \vee z)}{(w \vee y) \vee \{\ \}}}}$$

The pre-derivation ϕ_2 is open, because it contains no units or atoms. The pre-derivation ϕ_3 is flat because it contains no explicit substitutions. Instances of composition by inference are not labelled by a rule because we are not committed to any proof system here. We have not yet introduced a notion of correctness for composition by expansion, but we may note that the instances of composition by expansion in ϕ_1 and ϕ_2 are correct with respect to Definition 2.3.4.

Notation 2.1.3. Although it is not strictly necessary from a technical point of view, we informally consider derivations equivalent under the following associativity equations:

$$\boxed{\dfrac{\dfrac{\phi}{\psi}}{\chi}} = \boxed{\dfrac{\phi}{\dfrac{\psi}{\chi}}} \quad , \qquad \boxed{\dfrac{\dfrac{\phi}{\psi}}{\chi}} = \boxed{\dfrac{\phi}{\dfrac{\psi}{\chi}}} \quad , \qquad \boxed{\dfrac{\dfrac{\phi}{\psi}}{\chi}} = \boxed{\dfrac{\phi}{\dfrac{\psi}{\chi}}} \qquad \text{and} \qquad \boxed{\dfrac{\dfrac{\phi}{\psi}}{\chi}} = \boxed{\dfrac{\phi}{\dfrac{\psi}{\chi}}} \quad .$$

Consequently, we usually drop unnecessary boxes, and have indeed dropped them in the example above. The implicit understanding is that the precedence of the 'vertical' derivation constructors could be established if desired, and it would be respected by the operations that we define in this thesis.

Remark 2.1.4. In Definition 2.1.1, we distinguish atoms and variables. In standard proof systems they could be conflated because they both occur as leaves of a formula; however, in subatomic proof systems atoms are reclassified as connectives and so we define them separately. We distinguish typographically atoms-as-leaves and atoms-as-connectives by writing the latter in boldface: $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \ldots$.

Remark 2.1.5. We define composition by connective in such a way that connectives are positive in all of their arguments, but a more general definition with negative arguments could be given which is important for intuitionistic logics (see, for example, the atomic λ-calculus [22]).

Definition 2.1.6. The set of *formulae* $\mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U})$, denoted by A, B, C, D, ..., contains all and only the pre-derivations of $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$ that do not contain any composition by inference or by expansion. Depending on the circumstances, an involution called *negation* might be defined over formulae: $\bar{\cdot} : \mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U}) \to \mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U})$; if this is the case, variables and atoms denoted by x, y, ..., a, b, ... are called *positive* and those denoted by $\bar{x}$, $\bar{y}$, ..., $\bar{a}$, $\bar{b}$, ... are called *negative*. *Formula contexts* are denoted by $A\{\ \} \cdots \{\ \}$ or $A\{\ \}_{1\ldots n}$ (a formula with n holes), and other letters, in particular, H and K, can take the place of A; sometimes we write $A \equiv A\{B\}$ to indicate the formula A where the location of its subformula B has been singled out; we also write $A\{x_i\}_{1\ldots n}$ to stand for $A\{x_1\}\cdots\{x_n\}$. A *skeleton context* is a formula context where no atoms, units or free variables appear (but bound variables can appear).

Example 2.1.7. $A \equiv (B \wedge x) \vee (C\ \boldsymbol{a}\ D)$ is a formula, in which $\boldsymbol{a}$ appears as an atom-as-connective.

$A'\{\ \} \equiv (\{\ \} \wedge x) \vee (C\, \boldsymbol{a}\, D)$ is a formula context, and is such that $A'\{B\} \equiv A$.

$A''\{\ \}_{1\ldots4} \equiv (\{\ \}_1 \wedge \{\ \}_2) \vee (\{\ \}_3\ \boldsymbol{a}\ \{\ \}_4)$ is a skeleton context, and is such that $A\{B\}_1\{x\}_2\{C\}_3\{D\}_4 \equiv A$.

2.2 Substitutions

We now define actual substitutions, which generalise the standard notion of substitution on terms to pre-derivations. They have the usual properties of substitutions, for example, associativity. Their definition requires care because, contrary to standard substitutions for terms, we need to deal with negation and with the vertical composition of derivations.

Definition 2.2.1. *Actual substitutions* are maps $\mathcal{V} \to \mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U})$. We denote an actual substitution which maps x to A and leaves all other variables unchanged by $[A|x]$. Actual substitutions can be applied to pre-derivations and $[A|x]\,\phi$ stands for the pre-derivation obtained by replacing every free occurrence of x in ϕ by the formula A and, if an involution $\bar{\cdot}$ is defined, by replacing every free occurrence of $\bar{x}$ by $\bar{A}$; we say that this substitutes A for x in ϕ. When no involution is defined, actual substitutions can be extended to maps $\mathcal{V} \to \mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$, such that $[\phi|x]\,B$ is obtained by replacing every free occurrence of x in the formula B by the pre-derivation ϕ. Given $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{U}$, the set of actual substitutions is denoted by $\Sigma(\mathcal{A},\mathcal{C},\mathcal{U})$; actual substitutions are denoted by π, ρ, σ and τ, and so on, and the empty substitution is denoted by ϵ (the same as for explicit substitutions).

Note that the general case of application of actual substitutions of an pre-derivation into an pre-derivation is not defined. That would involve dealing with negation in a non-trivial way and is the subject of current research. The notion above suffices for the present work.

Example 2.2.2. The actual substitution of a formula C for the variable x in the pre-derivation ψ_1 in Example 2.1.2 is as follows:

$$[C|x]\boxed{\underbrace{\dfrac{\boxed{\dfrac{a}{a\vee x}}\wedge\left\langle\dfrac{b\wedge\mathsf{u}}{b}\middle|y\right\rangle(y\wedge y)}{\dfrac{(a\vee x)\wedge(b\wedge b)}{(a\vee b)\vee(x\vee b)}}}} \equiv \boxed{\underbrace{\dfrac{\boxed{\dfrac{a}{a\vee C}}\wedge\left\langle\dfrac{b\wedge\mathsf{u}}{b}\middle|y\right\rangle(y\wedge y)}{\dfrac{(a\vee C)\wedge(b\wedge b)}{(a\vee b)\vee(C\vee b)}}}} \quad .$$

Example 2.2.3. The actual substitution of a pre-derivation $\underbrace{\boxed{\dfrac{x}{y}}\wedge a}_{b}$ for the

variable x in a formula $(x \wedge x)$ is as follows:

$$\left[\underbrace{\boxed{\frac{x}{y}} \wedge a}_{b} \middle| x\right] (x \wedge x) \equiv \boxed{\underbrace{\boxed{\frac{x}{y}} \wedge a}_{b}} \wedge \boxed{\underbrace{\boxed{\frac{x}{y}} \wedge a}_{b}} \quad .$$

Definition 2.2.4. The *size* $|\phi|$ of a pre-derivation ϕ is the number of occurrences of variables, units and atoms appearing in it, not counting the substitution variables in explicit substitution terms. *i.e.*, $|\langle\phi|x\rangle\,\psi| = |\phi|+|\psi|$.

Example 2.2.5. The size of the derivation ϕ_1 in Example 2.1.2 is 16.

In this thesis, we define an equivalence relation on formulae based on 'flat expansion', which is the application of explicit substitutions as actual substitutions. Since applying substitutions normally increases the size of derivations, we show that there is a measure that decreases after such applications. We call it the 'rank' of a derivation and define it in the following way.

Definition 2.2.6. Given a pre-derivation ϕ, we associate a natural number called the *substitution degree* to each variable occurrence in ϕ; we denote the association of substitution degree d to the variable occurrence x in ϕ as in $\phi \equiv \phi\{x^d\}$. We assign substitution degrees this way:

1. Set a counter $d := 0$; assign the substitution degree 0 to all the free variables in ϕ; set the multiset $R := \emptyset_+$, the empty multiset.

2. Set $d := d + 1$; for every explicit substitution term $\langle\psi|x\rangle\,\chi$ appearing in ϕ such that all the variable occurrences in ψ have been assigned a substitution degree, assign the substitution degree d to the substitution variable x in $\langle\psi|x\rangle$ and to all the free occurrences of x in χ; set $R := R \uplus \{d\}_+$, the disjoint union of R and $\{d\}$; repeat Step 2 until a substitution degree has been assigned to all the variable occurrences in ϕ.

The multiset R so obtained is called the *rank* of ϕ and is denoted by $\mathsf{r}\,\phi$.

Example 2.2.7. The formula

$$\langle\langle x\ \alpha\ y|z\rangle((x\ \alpha\ z)\ \beta\ z)|w\rangle((x\ \alpha\ w)\ \beta\ w)$$

is progressively annotated with substitution degrees as follows:

1. $\langle\langle x^0\ \alpha\ y^0|z\rangle((x^0\ \alpha\ z)\ \beta\ z)|w\rangle((x^0\ \alpha\ w)\ \beta\ w)$
2. $\langle\langle x^0\ \alpha\ y^0|z^1\rangle((x^0\ \alpha\ z^1)\ \beta\ z^1)|w\rangle((x^0\ \alpha\ w)\ \beta\ w)$
3. $\langle\langle x^0\ \alpha\ y^0|z^1\rangle((x^0\ \alpha\ z^1)\ \beta\ z^1)|w^2\rangle((x^0\ \alpha\ w^2)\ \beta\ w^2)$

resulting in the rank $\{1,2\}_+$.

Example 2.2.8. The formula

$$\langle w|x\rangle\ \langle x\wedge w|y\rangle\ \langle x\wedge x|z\rangle(x\wedge(y\vee z))$$

is progressively annotated with substitution degrees as follows:

1. $\langle w^0|x\rangle\ \langle x\wedge w^0|y\rangle\ \langle x\wedge x|z\rangle(x\wedge(y\vee z))$
2. $\langle w^0|x^1\rangle\ \langle x^1\wedge w^0|y\rangle\ \langle x^1\wedge x^1|z\rangle(x^1\wedge(y\vee z))$
3. $\langle w^0|x^1\rangle\ \langle x^1\wedge w^0|y^2\rangle\ \langle x^1\wedge x^1|z^2\rangle(x^1\wedge(y^2\vee z^2))$

resulting in the rank $\{1,2,2\}_+$.

Remark 2.2.9. Each explicit substitution term $\langle\psi\big|x^d\rangle$ in a pre-derivation has substitution degree d. Intuitively, we can understand this substitution degree as corresponding to the longest chain of explicit substitutions which apply to the variable x. The rank of a pre-derivation is, therefore, a measure of the explicit substitutions and their depth, and can be used as an induction measure.

Proposition 2.2.10. *For any derivation context $\phi\{\ \}$, formula A and pre-derivation ψ, we have that $\mathsf{r}\,\phi\{[A|x]\,\psi\} < \mathsf{r}\,\phi\{\langle A|x\rangle\,\psi\}$ and $\mathsf{r}\,\phi\{[\psi|x]\,A\} < \mathsf{r}\,\phi\{\langle\psi|x\rangle\,A\}$ for the multiset ordering relation $<$.*

Proof. Removing an explicit substitution term $\langle A|x\rangle$ and applying it has two effects on the rank, both related to Step 2 in Definition 2.2.6. Let d be the substitution degree associated with $\langle A|x\rangle$ when Step 2 is applied to it:

- d is replaced in the rank by the elements of a multiset of lower degrees than d, associated with multiple copies of the explicit substitution terms contained in A – this multiset might be empty;
- the explicit substitution terms with variables bound by $\langle A|x\rangle$, and those bound by them and so on, do not get higher degrees – and might get lower degrees – after $\langle A|x\rangle$ is removed.

A similar argument works for a substitution term $\langle \psi | x \rangle$. $\square$

Example 2.2.11. In the formula of Example 2.2.7 there appear two substitution terms . They can be turned into actual substitutions in two ways:

$$\begin{aligned}&[\langle x\ \alpha\ y|z\rangle((x\ \alpha\ z)\ \beta\ z)|w](x\ \alpha\ w)\ \beta\ w)\\ &\equiv ((x^0\ \alpha\ \langle x^0\ \alpha\ y^0|z^1\rangle((x^0\ \alpha\ z^1)\ \beta\ z^1))\ \beta\ \langle x^0\ \alpha\ y^0|z^1\rangle((x^0\ \alpha\ z^1)\ \beta\ z^1))\end{aligned}$$

whose rank is $\{1,1\}_+$, and

$$\begin{aligned}&\langle[x\ \alpha\ y|z](x\ \alpha\ z)\ \beta\ z)|w\rangle((x\ \alpha\ w)\ \beta\ w)\\ &\equiv \langle(x^0\ \alpha\ (x^0\ \alpha\ y^0))\ \beta\ (x^0\ \alpha\ y^0)|w^1\rangle((x^0\ \alpha\ w^1)\ \beta\ w^1)\end{aligned}$$

whose rank is $\{1\}_+$. Each possible reduction therefore reduces the rank of the formula.

Reducing either again will result in

$$(x^0\ \alpha\ ((x^0\ \alpha\ (x^0\ \alpha\ y^0))\ \beta\ (x^0\ \alpha\ y^0)))\ \beta\ ((x^0\ \alpha\ (x^0\ \alpha\ y^0))\ \beta\ (x^0\ \alpha\ y^0))$$

whose rank is $\emptyset_+$.

2.3 Composition by Expansion

In this section we give a correctness criterion on composition by expansion for pre-derivations.

Definition 2.3.1. Given a formula A, we call the formula $\mathsf{fl}\, A$ the *flat expansion* of A if all the explicit substitution terms $\langle C|x\rangle\, D$ appearing in A are applied via actual substitutions $[C|x]\, D$, and $\mathsf{fl}\, A$ is the (unique) formula so obtained.

Remark 2.3.2. The uniqueness of the flat expansion can be seen by observing that there are no critical pairs. For example, consider a formula $\langle A|x\rangle\, \langle B|y\rangle\, C$ with two redexes. Applying the substitution $\langle A|x\rangle$ first results in the reduction

$$\langle A|x\rangle\, \langle B|y\rangle\, C \to \langle [A|x]\, B|y\rangle\, [A|x]\, C \to [[A|x]\, B|y]\, [A|x]\, C \quad ,$$

and applying the substitution $\langle B|x\rangle$ first results in the reduction

$$\langle A|x\rangle\, \langle B|y\rangle\, C \to \langle A|x\rangle\, [B|y]\, C \to [[A|x]\, B|y]\, [A|x]\, C \quad .$$

Example 2.3.3. The flat expansion of

$$\langle\langle x\ \alpha\ y|z\rangle((x\ \alpha\ z)\ \beta\ z)|w\rangle((x\ \alpha\ w)\ \beta\ w)$$

is

$$(x\ \alpha\ ((x\ \alpha\ (x\ \alpha\ y))\ \beta\ (x\ \alpha\ y)))\ \beta\ ((x\ \alpha\ (x\ \alpha\ y))\ \beta\ (x\ \alpha\ y)) \quad .$$

Definition 2.3.4. The two maps *premise* and *conclusion*, $\mathsf{pr}, \mathsf{cn} \colon \mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U}) \to \mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U})$ and the two maps *width* and *height*, $\mathsf{w}, \mathsf{h} \colon \mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U}) \to \mathbb{N}$ are so defined:

- If $\phi \in \mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U})$, then $\mathsf{pr}\,\phi \equiv \mathsf{cn}\,\phi \equiv \phi$ and $\mathsf{w}\,\phi = |\phi|$ and $\mathsf{h}\,\psi = 0$
- If $\phi \equiv \alpha(\psi_1, \ldots, \psi_n)$, then

$$\begin{aligned}\mathsf{pr}\,\phi &\equiv \alpha(\mathsf{pr}\,\psi_1, \ldots, \mathsf{pr}\,\psi_n)\\ \mathsf{cn}\,\phi &\equiv \alpha(\mathsf{cn}\,\psi_1, \ldots, \mathsf{cn}\,\psi_n)\\ \mathsf{w}\,\phi &= \mathsf{w}\,\psi_1 + \cdots + \mathsf{w}\,\psi_n\\ \mathsf{h}\,\phi &= \max(\mathsf{h}\,\psi_1, \ldots, \mathsf{h}\,\psi_n)\end{aligned}$$

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

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

- if $\phi \equiv \boxed{\dfrac{\psi}{\chi}}$ or $\phi \equiv \boxed{\overset{\psi}{\sim}\!\!\!\!\!\!-\ \chi}$, then

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

The set of *correct* pre-derivations $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$ contains all and only the pre-derivations of $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$ where for each term $\overset{\psi}{\sim}\!\!\!\!\!\!-\ \chi$ appearing in them $\mathsf{fl}\,\mathsf{cn}\,\psi \equiv \mathsf{fl}\,\mathsf{pr}\,\chi$. Henceforth, unless otherwise specified, we will use *pre-derivation* to mean *correct pre-derivation*.

Example 2.3.5. The example on the left is not a correct pre-derivation, because the composition is not between two pre-derivations; the pre-derivation on the right is correct (assuming that y does not occur free in $A\{x\}$):

$$\underset{\langle C\{B\}|x\rangle}{\underbrace{\langle B|y\rangle\,\langle C\{y\}|x\rangle}}A\{x\} \qquad\qquad \underset{\langle C\{B\}|x\rangle\,A\{x\}}{\underbrace{\langle B|y\rangle\,\langle C\{y\}|x\rangle\,A\{x\}}} \quad .$$

For simplicity, our formalism is not equipped with a notion of beta-reduction. The correctness of an instance $\underset{\psi}{\overset{\phi}{\sim}}$ of composition by expansion is defined only by the identity of $\mathsf{fl}\,\mathsf{cn}\,\phi$ and $\mathsf{fl}\,\mathsf{pr}\,\psi$. In order to use this formalism as a basis for proof systems, we require that composition by expansion can be decided in polynomial time; we therefore give the following method for checking the identity of the flat expansions of two formulae, which treats formulae as term graphs as in [29] and adapts the algorithm for linear unification given in [28].

Proposition 2.3.6. *Given formulae A and B, the identity $\mathsf{fl}\,A \equiv \mathsf{fl}\,B$ can be decided in linear time on $|A| + |B|$ by comparing the flat expansions of A and B without actually performing the substitutions. 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, ..., x_n are fresh, distinct variables, C_0, ..., C_n are flat, $C_0 \notin \{x_1, \ldots, x_n\}$, each of C_1, ..., 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* $C_i \equiv \alpha(C_i^1, \ldots, C_i^l)$ *and* $D_j \equiv \alpha(D_j^1, \ldots, D_j^l)$ *and, for* $1 \leq h \leq l$*,* $p(C_i^h, D_j^h)$ *succeeds, then return success;*
4. *otherwise, return failure.*

Proof. 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 such as $\langle C_i|x_i\rangle\, K\{x_i\} \ldots \{x_i\}$ and $\langle D_j|y_j\rangle\, K\{y_j\} \ldots \{y_j\}$, the procedure $p(C_i, D_j)$ need only be performed once, so the comparison is linear on the size of the original formulae. □

2.4 Synchronal Composition

Definition 2.4.1. Let ψ and χ be two pre-derivations such that $\mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi$. We define a pre-derivation called the *synchronal composition* of ψ and χ, denoted as

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

we do so by structural induction, as follows:

1. if ψ is a formula, then $\boxed{\begin{matrix}\psi\\ \cdots\\ \chi\end{matrix}} \equiv \chi$; similarly, if χ is a formula, then $\boxed{\begin{matrix}\psi\\ \cdots\\ \chi\end{matrix}} \equiv \psi$;
2. if $\psi \equiv \alpha(\psi_1, \ldots, \psi_n)$ and $\chi \equiv \alpha(\chi_1, \ldots, \chi_n)$, then $\boxed{\begin{matrix}\psi\\ \cdots\\ \chi\end{matrix}} \equiv \alpha\left(\boxed{\begin{matrix}\psi_1\\ \cdots\\ \chi_1\end{matrix}}, \ldots, \boxed{\begin{matrix}\psi_n\\ \cdots\\ \chi_n\end{matrix}}\right)$;

3. if $\psi \equiv \langle \psi_1 | x \rangle \, \psi_2$ and $\chi \equiv \langle \chi_1 | x \rangle \, \chi_2$, then $\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi\end{array}} \equiv \left\langle \boxed{\begin{array}{c}\psi_1\\ \cdots\\ \chi_1\end{array}} \middle| x \right\rangle \boxed{\begin{array}{c}\psi_2\\ \cdots\\ \chi_2\end{array}}$;

4. if $\psi \equiv \boxed{\begin{array}{c}\psi_1\\ \sim\\ \psi_2\end{array}}$, then $\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi\end{array}} \equiv \boxed{\begin{array}{c}\psi_1\\ \sim\\ \boxed{\begin{array}{c}\psi_2\\ \cdots\\ \chi\end{array}}\end{array}}$; similarly, if $\chi \equiv \boxed{\begin{array}{c}\chi_1\\ \sim\\ \chi_2\end{array}}$, then $\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi\end{array}} \equiv \boxed{\begin{array}{c}\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi_1\end{array}}\\ \sim\\ \chi_2\end{array}}$;

5. if $\psi \equiv \boxed{\dfrac{\psi_1}{\psi_2}}$, then $\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi\end{array}} \equiv \boxed{\begin{array}{c}\psi_1\\ \hline \boxed{\begin{array}{c}\psi_2\\ \cdots\\ \chi\end{array}}\end{array}}$; similarly, if $\chi \equiv \boxed{\dfrac{\chi_1}{\chi_2}}$, then $\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi\end{array}} \equiv \boxed{\begin{array}{c}\boxed{\begin{array}{c}\psi\\ \cdots\\ \chi_1\end{array}}\\ \hline \chi_2\end{array}}$.

Remark 2.4.2. Note that the dotted line is a composition operator and is not part of the language. As is shown in [20], the synchronal composition of two pre-derivations is unique because we can interpret it as a constructor subject to term rewriting (in the direction suggested by the above definition). That rewriting relation is then easily proved to be terminating and confluent, under the assumption that compositions by expansion and inference are associative. Moreover, synchronal composition is associative, and therefore we omit many boxes in the rest of the thesis.

Remark 2.4.3. Given $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{U}$, any set of pre-derivations $\mathcal{P}(\mathcal{A}, \mathcal{C}, \mathcal{U})$ is closed under synchronal composition and composition by expansion. If two pre-derivations can be composed synchronally, they can be composed by expansion.

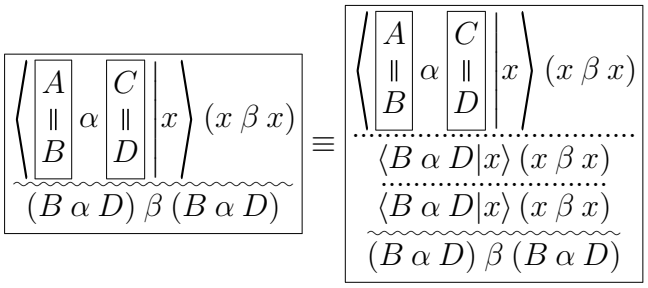
Definition 2.4.4. A *section* of a pre-derivation ϕ is any formula A such that

$$\phi \equiv \boxed{\begin{array}{c}\psi\\ \cdots\\ A\\ \cdots\\ \chi\end{array}} \quad ,$$

for some pre-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.

Example 2.4.5. This is how we can extract a section $\langle B \, \alpha \, D | x \rangle (x \, \beta \, x)$ above

a composition by expansion in the derivation on the left:



2.5 Open Deduction with Substitution

Definition 2.5.1. We define two *formalisms*:

- *(propositional) open deduction with substitution*, denoted by ODS, is the union of all sets of pre-derivations $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$, for all sets of atoms $\mathcal{A}$, connectives $\mathcal{C}$ and units $\mathcal{U}$;
- *(propositional) open deduction*, denoted by OD, is the subset of ODS that contains all pre-derivations where no explicit substitutions and no compositions by expansion appear.

A generic formalism is denoted by $\mathcal{O}$.

Chapter 3

Proof Systems

In this chapter we define the proof systems which we investigate and use in this work. In our design of proof systems, we are particularly motivated by the pursuit of linearity and avoidance of negation, which result in proof systems that interact with explicit substitution in a straightforward way.

The main logic of interest in this work is propositional classical logic, and we will introduce and compare multiple systems for this logic: those systems with and without unit-equality inference steps, explicit substitutions, atoms-as-connectives, and the mix rule. We focus on propositional classical logic because it best allows us to assess the effect on the size of proofs from introducing explicit substitutions and eliminating unit-equality inference steps, by comparison in Chapter 7 with the benchmark class of systems substitution Frege. Systems for propositional classical logic are also sufficiently symmetric that in Chapter 5 we are able to prove a technical result, the Eversion Lemma, with a high degree of generality. Some results, such as the completeness of the system obtained by eliminating unit-equality inference steps in Chapter 6, would be possible to show for multiplicative linear logic with units (albeit without the complexity result); we draw comparisons where appropriate and refer to the system SAMLLS in [2].

3.1 Preliminaries

Definition 3.1.1. Given $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{U}$, a *proof system* is a finite set of relations on $\mathcal{F}(\mathcal{A}, \mathcal{C}, \mathcal{U})$, each of which is called an *inference rule* and is decidable in polynomial time on the size of its arguments. We denote inference rules with

the letter r and various other notations. 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

$$r\,\frac{\psi}{\chi} \quad .$$

Given a formalism $\mathcal{O}$ and a proof system $\mathcal{S}$ built on $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{U}$, we say that a pre-derivation ϕ in $\mathcal{P}(\mathcal{A},\mathcal{C},\mathcal{U})$ and in $\mathcal{O}$ is a *derivation in* $\mathcal{S}$-$\mathcal{O}$ if every inference step in ϕ is an instance of some rule of $\mathcal{S}$; one way to denote such a derivation is

$$\begin{array}{c} A \\ \phi \,\|\, \mathcal{S}\text{-}\mathcal{O} \\ B \end{array} \quad ,$$

where A and B are the premise and conclusion of ϕ. The set of derivations in $\mathcal{S}$-$\mathcal{O}$ is denoted as $\mathcal{D}_{\mathcal{S}\text{-}\mathcal{O}}$. When it is clear from the context, indications of formalism might be omitted. Sometimes, a couple $\mathcal{S}$-$\mathcal{O}$ will be referred to as a proof system.

Remark 3.1.2. If a derivation is in $\mathcal{S}$-OD then it is in $\mathcal{S}$-ODS.

Example 3.1.3. Consider a proof system $\mathcal{S}$ with rules r_1, r_2 such that $(A, B) \in r_1$ and $(B\,\alpha\,B, C) \in r_2$, and consider the pre-derivations

$$\phi_1 \equiv \boxed{r_2\,\frac{\boxed{r_1\,\frac{A}{B}}\;\alpha\;\boxed{r_1\,\frac{A}{B}}}{C}} \qquad \text{and} \qquad \phi_2 \equiv \boxed{r_2\,\frac{\underbrace{\left\langle r_1\,\frac{A}{B} \middle| x \right\rangle}\,(x\;\alpha\;x)}{B\;\alpha\;B}\;\Big/\;C} \quad .$$

Then ϕ_1 is a derivation in both $\mathcal{S}$-OD and $\mathcal{S}$-ODS and ϕ_2 is a derivation in $\mathcal{S}$-ODS but not in $\mathcal{S}$-OD.

In specifying separately the formalism and the proof system, we are drawing a distinction between two aspects of proofs that would otherwise be conflated: the mechanisms for composition and the set of inferences. In Frege and Gentzen systems this is not so easily done, because the rules and the composition mechanisms are given together. This distinction allows us to reason generally about derivations, without being committed to a particular proof system. We put composition by expansion at the level of the formalism rather than introducing a new rule to the proof system because that does not depend on a particular logical system.

Remark 3.1.4. For every $\mathcal{S}$ and $\mathcal{O}$, the set of derivations $\mathcal{D}_{\mathcal{S}\text{-}\mathcal{O}}$ is closed under synchronal composition and, if $\mathcal{O} = \mathsf{ODS}$, also under composition by expansion.

In many cases, proof systems are equipped with equivalence relations on formulae. Those equivalences can be viewed as special cases of 'invertible' inference rules, that is, rules that are valid in both directions, typically representing semantic equivalence. In this thesis, equivalence relations allow us to define classes of structurally equivalent derivations and, at the same time, generate canonical derivations that represent those classes. For these reasons, we introduce some special notation, in particular, to represent inference rules and inference steps.

Notation 3.1.5. If A and B belong to a given set of formulae $\mathcal{F}(\mathcal{A},\mathcal{C},\mathcal{U})$, and unless specified differently, the notation

$$ {}_r\frac{A}{B} \qquad \text{stands for} \qquad r = \{\, (\sigma A, \sigma B) \mid \sigma \in \Sigma(\mathcal{A},\mathcal{C},\mathcal{U}) \,\} \quad . $$

If an equivalence relation $=$ is defined on formulae, the notation

$$ {}_r\frac{A}{\overline{\overline{B}}} \qquad \text{stands for} \qquad r = \left\{ (A', B') \;\middle|\; \begin{array}{l} A' = \sigma A,\ B' = \sigma B \\ \sigma \in \Sigma(\mathcal{A},\mathcal{C},\mathcal{U}) \end{array} \right\} \quad . $$

Inference steps can be decorated by a label r denoting the rule r of which they are an instance, and, in case the rule was defined via $=$, the double line is sometimes used, too.

This notation works well for propositional inference rules and allows us to use for their definition the same notion of substitution that we use for proof systems. The variables in the rules are object-level variables (x, y, ...) rather than meta-level ones (A, B, ...). It does not work equally well for rules with quantifiers because, for them, variable capture might become an issue. However, we only mention quantifier rules informally in this thesis and, for them, we use a more standard, informal notation.

The use of inference steps modulo equality subsumes the invertible inference steps and will be instrumental in comparing the structure of proof systems with explicit substitutions and those with rules for unit equations, which we will do in Chapter 6.

Example 3.1.6. Suppose that we have a proof system $\mathcal{S}$ containing a medial rule modulo equality $m\dfrac{(w \wedge x) \vee (y \wedge z)}{(w \vee y) \wedge (x \vee z)}$ and invertible rules $x \wedge \mathsf{t} = x$ and $x \vee \mathsf{f} = x$. Then the inference step $m\dfrac{A \vee (\mathsf{f} \wedge D)}{A \wedge (\mathsf{t} \vee D)}$ is an instance of the medial rule modulo equality. In a system without rules modulo equality, this would be expanded as

$$m\dfrac{\boxed{=\dfrac{A}{A \wedge \mathsf{t}}} \vee (\mathsf{f} \wedge D)}{\boxed{=\dfrac{A \vee \mathsf{f}}{A}} \wedge (\mathsf{t} \vee D)} \quad .$$

3.2 Standard Proof Systems

The deep-inference methodology is motivated in part by the pursuit of locality, meaning that we want to be able to check inference steps in constant time. In order to achieve this, rules should be as 'small' as possible. This has been achieved in a natural way by giving proof systems all of whose rules are either linear or atomic [11, 9, 6, 7]. In particular, locality benefits the proof theory of classical logic via the normalisation mechanisms employing 'atomic flows' [4, 21, 19]. In this work, we develop the idea that a more extreme form of linearity, in which we require not only linearity on the atoms but also the units, can subsume the need for atomic rules and lead to proof systems with good properties, in particular a better form of closure under substitution.

We therefore introduce the standard systems for classical logic in deep inference, including a variant with inference steps defined via an equality relation on formulae.

Definition 3.2.1. Given $\mathcal{A} = \{a, b, c, \dots\}$, $\mathcal{C} = \mathcal{C}_2 = \{\vee, \wedge\}$ and $\mathcal{U} = \{\mathsf{f}, \mathsf{t}\}$, we define the proof systems SKSg, $\mathsf{SKSg}^{=}$, SKS and $\mathsf{SKS}^{=}$. We assume that there is an involution $\bar{\cdot}$ on formulae that realises De Morgan's laws, *i.e.*, $\overline{A \vee B} \equiv \bar{A} \wedge \bar{B}$ and $\overline{A \wedge B} \equiv \bar{A} \vee \bar{B}$. The equality relation $=$ on formulae is defined by the equalities in Figure 3.1, closed by reflexivity, symmetry and transitivity, and where $K\{\ \}$ is any context. SKSg is the set of inference rules in Figure 3.2 and $\mathsf{SKSg}^{=}$ is the set of those in Figure 3.3. Systems SKS and $\mathsf{SKS}^{=}$ are obtained from SKSg and $\mathsf{SKSg}^{=}$, respectively, by restricting

$$
\begin{array}{cc}
x \vee y = y \vee x & x \vee \mathsf{f} = x \\
x \wedge y = y \wedge x & x \wedge \mathsf{t} = x \\
x \vee (y \vee z) = (x \vee y) \vee z & \mathsf{t} \vee \mathsf{t} = \mathsf{t} \\
x \wedge (y \wedge z) = (x \wedge y) \wedge z & \mathsf{f} \wedge \mathsf{f} = \mathsf{f}
\end{array}
$$

$$
\text{if } A = B \text{ then } K\{A\} = K\{B\}
$$

Figure 3.1: Equivalence relation $=$ on the formulae of systems SKSg, SKS, $\mathsf{SKSg}^{=}$ and $\mathsf{SKS}^{=}$.

$$
\begin{array}{cccc}
\bar{\mathsf{i}}\dfrac{x \wedge \bar{x}}{\mathsf{f}} & \overline{\mathsf{w}}\dfrac{x}{\mathsf{t}} & \overline{\mathsf{c}}\dfrac{x}{x \wedge x} & \overline{\mathsf{sw}}\dfrac{(x \vee y) \wedge (z \wedge w)}{(x \wedge z) \vee (y \wedge w)} \\
\mathsf{i}\dfrac{\mathsf{t}}{x \vee \bar{x}} & \mathsf{w}\dfrac{\mathsf{f}}{x} & \mathsf{c}\dfrac{x \vee x}{x} & \mathsf{sw}\dfrac{(x \vee y) \wedge (z \vee w)}{(x \wedge z) \vee (y \vee w)}
\end{array}
\qquad
\mathsf{m}\dfrac{(x \wedge y) \vee (z \wedge w)}{(x \vee z) \wedge (y \vee w)} \qquad =
$$

Figure 3.2: Inference rules of systems SKSg and SKS, where the inference rule $=$ is defined in Figure 3.1.

to variables, atoms and units the formulae that are substituted into x and $\bar{x}$ in the rules i, $\bar{\mathsf{i}}$, w, $\overline{\mathsf{w}}$, c and $\overline{\mathsf{c}}$. In any of these proof systems, a *proof* is a derivation whose premise is t.

Normally, there are no variables in standard derivations with no substitutions, but their presence would not cause any issues because they can be treated as atoms.

Remark 3.2.2. We present the switch here as two distinct rules, $\overline{\mathsf{sw}}$ and sw.

$$
\begin{array}{cccc}
\bar{\mathsf{i}}\dfrac{x \wedge \bar{x}}{\mathsf{f}} & \overline{\mathsf{w}}\dfrac{x}{\mathsf{t}} & \overline{\mathsf{c}}\dfrac{x}{x \wedge x} & \overline{\mathsf{sw}}\dfrac{(x \vee y) \wedge (z \wedge w)}{(x \wedge z) \vee (y \wedge w)} \\
\mathsf{i}\dfrac{\mathsf{t}}{x \vee \bar{x}} & \mathsf{w}\dfrac{\mathsf{f}}{x} & \mathsf{c}\dfrac{x \vee x}{x} & \mathsf{sw}\dfrac{(x \vee y) \wedge (z \vee w)}{(x \wedge z) \vee (y \vee w)}
\end{array}
\qquad
\mathsf{m}\dfrac{(x \wedge y) \vee (z \wedge w)}{(x \vee z) \wedge (y \vee w)} \qquad =
$$

Figure 3.3: Inference rules of systems $\mathsf{SKSg}^{=}$ and $\mathsf{SKS}^{=}$, where the inference rule $=$ is defined in Figure 3.1.

These could be collapsed into a single rule

$$\frac{x \wedge (y \vee z)}{(x \wedge y) \vee z} \quad ,$$

from which $\overline{\mathsf{sw}}$ and sw can be recovered via unit equations. We present the switch as two rules here for compatibility with its presentation in subatomic systems.

Example 3.2.3. The following two derivations have the same premise and conclusion, both are in $\mathsf{SKS}^{=}\text{-}\mathsf{OD}$ but the left one is also in $\mathsf{SKS}\text{-}\mathsf{OD}$ while the right one is not:

$$\mathsf{m}\dfrac{\mathsf{=}\dfrac{b}{\mathsf{i}\dfrac{\mathsf{t}}{\bar{a} \vee a} \wedge \mathsf{=}\dfrac{b}{\mathsf{0} \vee b}}}{(\bar{a} \wedge \mathsf{0}) \vee (a \vee b)} \qquad \text{and} \qquad \mathsf{m}\dfrac{\mathsf{=}\dfrac{b}{\mathsf{i}\dfrac{\mathsf{t}}{\bar{a} \vee a} \wedge b}}{(\bar{a} \wedge \mathsf{0}) \vee (a \vee b)} \quad .$$

Note that $\mathsf{i}\dfrac{\mathsf{t}}{\bar{a} \vee a}$ is an instance of $\mathsf{i}\dfrac{\mathsf{t}}{x \vee \bar{x}}$ where $\bar{a}$ is substituted into x and we apply $\bar{\bar{a}} \equiv a$.

3.3 Subatomic Proof Systems

The rules in systems SKSg, $\mathsf{SKSg}^{=}$, SKS and $\mathsf{SKS}^{=}$ are given in three parts: the so-called structural rules i, w, c and their duals; the equality rule expressing the equalities in Figure 3.1; and the logical rules sw, $\overline{\mathsf{sw}}$, and m, which share a common shape. This shape is characteristic of deep-inference logical systems.

By considering the atoms as self-dual, non-commutative binary connectives, whose arguments correspond to the atom's possible truth values, we can also give the structural rules as instances of the same linear shape for a wide range of logical systems [3]. The uniformity and linearity of the rule shape has the benefit of simplifying normalisation arguments because it reduces the possibilities for how inference steps interact with one another in a derivation.

In this rule shape, there is no duplication or negation of the arguments; this is recovered via an interpretation map back to the language with standard atoms. For example, $\mathsf{0}\,\boldsymbol{a}\,\mathsf{1}$ is interpreted as a positive instance of the atom a and $\mathsf{1}\,\boldsymbol{a}\,\mathsf{0}$ is interpreted as a negative instance $\bar{a}$. In addition, if both arguments of an atom are equal then that argument is the interpretation, so that $\mathsf{0}\,\boldsymbol{a}\,\mathsf{0}$ is interpreted as f and $\mathsf{1}\,\boldsymbol{a}\,\mathsf{1}$ is interpreted as t. Then the inference steps

$$\frac{(\mathsf{0}\,\boldsymbol{a}\,\mathsf{1}) \vee (\mathsf{0}\,\boldsymbol{a}\,\mathsf{1})}{(\mathsf{0} \vee \mathsf{0})\,\boldsymbol{a}\,(\mathsf{1} \vee \mathsf{1})} \qquad \text{and} \qquad \frac{(\mathsf{0}\,\boldsymbol{a}\,\mathsf{1}) \wedge (\mathsf{1}\,\boldsymbol{a}\,\mathsf{0})}{(\mathsf{0} \wedge \mathsf{1})\,\boldsymbol{a}\,(\mathsf{1} \wedge \mathsf{0})}$$

are instances of the rule shape, corresponding, via the unit equalities in Figure 3.1, respectively to $\mathsf{c}\dfrac{a \vee a}{a}$, which is an instance of contraction in System SKS, and $\mathsf{i}\dfrac{a \wedge \bar{a}}{\mathsf{f}}$, a cut. In [2, 3], this interpretation is formally defined and it is shown that cut elimination and other normalisations can be lifted from the subatomic proof systems to the standard ones. Such an interpretation is an example of 'abstraction', which is a more general concept than interpretation, and which we define in Chapter 4. The particular abstraction that we define in this thesis allows us to go further and design subatomic proof systems that do not possess unit equalities but still retain the structure of standard systems concerning normalisation.

Definition 3.3.1. Let $\mathcal{C}_2$ be a set of binary connectives and let the two maps *down-saturation* and *up-saturation* be defined, respectively, as $\check{\cdot}\colon \mathcal{C}_2 \to \mathcal{C}_2$ and $\hat{\cdot}\colon \mathcal{C}_2 \to \mathcal{C}_2$, such that, for $\alpha \in \mathcal{C}_2$,

$$\check{\check{\alpha}} = \hat{\check{\alpha}} = \check{\alpha} \qquad \text{and} \qquad \hat{\hat{\alpha}} = \check{\hat{\alpha}} = \hat{\alpha} \quad .$$

Given atoms $\mathcal{A}$ and units $\mathcal{U}$, the proof systems

$$\mathsf{DT}(\mathcal{A}, \mathcal{C}_2, \check{\cdot}, \hat{\cdot}, \mathcal{U}) \quad \text{and} \quad \mathsf{rDT}(\mathcal{A}, \mathcal{C}_2, \check{\cdot}, \hat{\cdot}, \mathcal{U})$$

consist of the rules in Figure 3.4. We denote $\mathsf{DT}(\mathcal{A}, \mathcal{C}_2, \check{\cdot}, \hat{\cdot}, \mathcal{U})$ as DT and $\mathsf{rDT}(\mathcal{A}, \mathcal{C}_2, \check{\cdot}, \hat{\cdot}, \mathcal{U})$ as rDT when the language constructors are not important or are clear from the context. We will sometimes use the square connectives $\sqcap, \sqcup$ for a pair of connectives $\hat{\alpha}, \check{\alpha}$ to improve readability.

The system rDT realises a 'rigid' form of inference and the system DT realises a 'lax' one, and the distinction between these will be made clear in

$$\mathsf{DT}\quad \widehat{\alpha}\,\frac{x\;\widehat{\alpha}\;y}{x\;\alpha\;y} \qquad \breve{\alpha}\,\frac{x\;\alpha\;y}{x\;\breve{\alpha}\;y}$$

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

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

Figure 3.4: Systems rDT and DT.

Chapter 5. For the technical purposes of this thesis, we could ignore the rigid system and only work with the lax one. However, there are three reasons to develop the theory of both systems:

- the properties of both systems can be proved at the same time with no extra effort;
- the rigid system, rDT, is slightly simpler and has implicitly been adopted already for subatomic logics;
- the rigid and lax forms of inference generate different systems with unary connectives (modalities and quantifiers) that can have different uses in different logics, and this thesis provides a foundation for them.

Remark 3.3.2. In the subatomic systems of this thesis, sets of connectives closed by saturation and negation coincide, *i.e.*, $\{\alpha, \breve{\alpha}, \widehat{\alpha}\} = \{\alpha, \bar{\alpha}\}$ for any connective α. Indeed, for example, $\{\vee, \breve{\vee}, \widehat{\vee}\} = \{\vee, \bar{\vee}\} = \{\vee, \wedge\}$ and $\{\boldsymbol{a}, \breve{\boldsymbol{a}}, \widehat{\boldsymbol{a}}\} = \{\boldsymbol{a}, \bar{\boldsymbol{a}}\} = \{\boldsymbol{a}\}$. However, this is not a consequence of the definition and there are there could be proof systems where this does not happen.

Decision trees are binary trees in which each interior node is labelled by an atom and each leaf is labelled by a unit or variable [38]. We read the decision tree represented by $\boldsymbol{AaB}$ as 'if a then B, else A'. In [5], the standard language of propositional classical logic is augmented by decision trees. This way, we obtain a proof system, called $\mathsf{DT}^{\mathsf{sa}}$ in [5] and rKDT in this thesis, containing exactly those rules that are generated by the subatomic rule shape of rDT, with good normalisation properties and efficient derivations. That system realises a conservative extension of standard propositional logic. Even though we introduce new rules to a proof system for classical propositional

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

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

where $\alpha, \beta \in \{\vee, \wedge, \boldsymbol{a}, \boldsymbol{b}, \dots\}$

and $\quad \breve{\vee} = \breve{\wedge} = \vee \qquad \breve{\boldsymbol{a}} = \widehat{\boldsymbol{a}} = \boldsymbol{a}$

$\qquad\quad \widehat{\vee} = \widehat{\wedge} = \wedge \qquad \breve{\boldsymbol{b}} = \widehat{\boldsymbol{b}} = \boldsymbol{b}$

$\qquad\qquad\qquad\qquad\qquad\quad \vdots$

Figure 3.5: System rKDT.

logic and extend its semantic interpretation, the complexity of the proof system is reduced, in particular concerning its normalisation theory, because of the uniformity of rDT.

The next definition introduces five proof systems for propositional classical logic with decision trees that specialise rDT and DT. Four of them differ for two characteristics in all the possible combinations: rigidity vs. laxness and the presence vs. the absence of inference rules for equality. The fifth system operates inference modulo equality and contains all the derivations of all the other four systems: we use it to define an abstraction map, in Chapter 4, that allows us to state our main result on strict linearity. All these systems share a common language and all the inference rules that control the interactions between connectives. When there is no reason to deal with the differences, we collectively refer to all five systems as KDT, for classical decision trees.

Definition 3.3.3. Given $\mathcal{A} = \emptyset, \mathcal{C} = \mathcal{C}_2 = \{\vee, \wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$ and $\mathcal{U} = \{0, 1\}$, we define the proof system rKDT (for *rigid classical decision trees*), which is a special case of rDT, in Figure 3.5. Connectives $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, ... are called *atom connectives*. System KEq consists of the inference rules $=_i$s defined in Figure 3.6. We then define $\mathsf{rKDTEq} = \mathsf{rKDT} \cup \mathsf{KEq}$. Systems KDT and

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

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

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

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

Figure 3.6: System KEq.

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

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

where $\alpha, \beta \in \{\vee, \wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$

$\breve{\vee} = \breve{\wedge} = \vee \qquad \breve{\boldsymbol{a}} = \widehat{\boldsymbol{a}} = \boldsymbol{a}$

$\widehat{\vee} = \widehat{\wedge} = \wedge \qquad \breve{\boldsymbol{b}} = \widehat{\boldsymbol{b}} = \boldsymbol{b}$

$\vdots$

$$\breve{\wedge} \frac{x \wedge y}{x \vee y}$$

$=$ equivalence relation defined by

$x \vee \mathsf{0} = \mathsf{0} \vee x = x \qquad \mathsf{0}\,\gamma\,\mathsf{0} = \mathsf{0}$

$x \wedge \mathsf{1} = \mathsf{1} \wedge x = x \qquad \mathsf{1}\,\delta\,\mathsf{1} = \mathsf{1}$

where $\gamma \in \{\wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$, $\delta \in \{\vee, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$

$\langle \boldsymbol{A}|x\rangle\, \boldsymbol{B} = [\boldsymbol{A}|x]\, \boldsymbol{B}$

if $\boldsymbol{A} = \boldsymbol{B}$ then $\boldsymbol{K}\{\boldsymbol{A}\} = \boldsymbol{K}\{\boldsymbol{B}\}$

Figure 3.7: System $\mathsf{KDTEq}^{=}$.

KDTEq are the proof systems obtained by adding the mix rule

$$\breve{\wedge}\frac{x \wedge y}{x \vee y}$$

to Systems rKDT and rKDTEq, respectively. System $\mathsf{KDTEq}^{=}$ is defined in Figure 3.7, where $=$ is an equivalence relation, *i.e.*, it is closed by reflexivity, symmetry and transitivity. In these systems, a *proof* is a derivation whose premise is equal to $\mathsf{1}$ by the equality relation $=$.

Remark 3.3.4. We have the following inclusions between the proof systems for classical logic:

$$\mathcal{D}_{\mathsf{rKDT}\text{-}\mathcal{O}} \begin{array}{c} \subset \\ \subset \end{array} \begin{array}{c} \mathcal{D}_{\mathsf{KDT}\text{-}\mathcal{O}} \\ \mathcal{D}_{\mathsf{rKDTEq}\text{-}\mathcal{O}} \end{array} \begin{array}{c} \subset \\ \subset \end{array} \mathcal{D}_{\mathsf{KDTEq}\text{-}\mathcal{O}} \subset \mathcal{D}_{\mathsf{KDTEq}^{=}\text{-}\mathcal{O}} \quad ,$$

where $\mathcal{O} \in \{\mathsf{OD}, \mathsf{ODS}\}$.

Remark 3.3.5. No free variables appear in proofs because if a free variable appears anywhere in a proof, it appears in its premise, but a formula with a free variable cannot be equal to $\mathsf{1}$.

Remark 3.3.6. We can check the equality $\boldsymbol{A} = \boldsymbol{B}$ in linear time, therefore $\mathsf{KDTEq}^{=}$ is indeed a proof system. The check can be performed by the same procedure of Remark 2.3.6, where the notion of normal representation is modified by adding the following condition: in the normal representation $\langle C_n | x_n \rangle \cdots \langle C_1 | x_1 \rangle \, x_1$ of a formula, no C_i matches the left side of a unit equality in Figure 3.7.

The rules $\breve{\alpha}$ and $\widehat{\alpha}$ in Figure 3.4, when specialised to the language of KDT, are sound for classical logic, because $x \wedge y$ entails $x \vee y$. It is also sound for other logics, such as, for example, linear logic with mix [17] (because $x \otimes y$ entails $x \parr y$) and BV [18] (because $x \otimes y$ entails $x \triangleleft y$, which in turn entails $x \parr y$). These rules do not contribute to the completeness of KDTEq because the rules obtained from rDT are (more than) sufficient. However, it allows us to obtain derivations that behave well under the operation of 'projection', which we need in Chapter 8 to prove cut elimination, and does not affect any of the other results in this thesis. Moreover, they fits a very general scheme for proof systems that we define in Chapter 5 and make the difference between rKDT (the 'rigid' version, whence the 'r') and KDT (the 'lax' version).

Proposition 3.3.7. *Systems* rKDTEq, KDTEq *and* KDTEq$^{=}$ *are sound and complete for propositional classical logic with decision trees, as shown by C. Barrett and Guglielmi in [5].*

Remark 3.3.8. Standard derivations can be translated to subatomic in a straightforward way. We include here the translation from SKS-OD to KDTEq-OD, which we call sa.

- If $\phi \in \mathcal{V}$ then $\mathsf{sa}\,\phi \equiv \phi$
- If $\phi \equiv \mathsf{t}$ then $\mathsf{sa}\,\phi \equiv 1$; and if $\phi \equiv \mathsf{f}$ then $\mathsf{sa}\,\phi \equiv 0$.
- If $\phi \equiv a$ then $\mathsf{sa}\,\phi \equiv 0 \,\boldsymbol{a}\, 1$
- If $\phi \equiv \bar{a}$ then $\mathsf{sa}\,\phi \equiv 1 \,\boldsymbol{a}\, 0$
- If $\phi \equiv (\psi\ \alpha\ \chi)$, for $\alpha \in \{\wedge, \vee\}$ then $\mathsf{sa}\,\phi \equiv (\mathsf{sa}\,\psi\ \alpha\ \mathsf{sa}\,\chi)$
- If $\phi \equiv \mathsf{c}\dfrac{\begin{array}{c}\psi\\ \vdots\\ a \vee a\end{array}}{\begin{array}{c}a\\ \vdots\\ \chi\end{array}}$ then $\mathsf{sa}\,\phi \equiv \dfrac{\begin{array}{c}\mathsf{sa}\,\psi\\ \vdots\\ (0\,\boldsymbol{a}\,1) \vee (0\,\boldsymbol{a}\,1)\end{array}}{\boldsymbol{a}^{\check{\vee}}\ \begin{array}{c}\mathsf{sa}\,\chi\end{array}}$; similarly for $\bar{\mathsf{c}}$.
- If $\phi \equiv \mathsf{i}\dfrac{\begin{array}{c}\psi\\ \vdots\\ 1\end{array}}{\begin{array}{c}a \vee \bar{a}\\ \vdots\\ \chi\end{array}}$ then $\mathsf{sa}\,\phi \equiv \dfrac{\begin{array}{c}\mathsf{sa}\,\psi\\ \vdots\\ 1\end{array}}{\vee^{\check{\boldsymbol{a}}}\ \begin{array}{c}\mathsf{sa}\,\chi\end{array}}$; similarly for $\bar{\mathsf{i}}$.

- If $\phi \equiv \dfrac{\begin{array}{c}\psi\\ \vdots\\ 0\end{array}}{\mathsf{w}\,\dfrac{}{a}}\ \begin{array}{c}\\ \vdots\\ \chi\end{array}$ then $\mathsf{sa}\,\phi \equiv \dfrac{\mathsf{sa}\,\psi}{0\ \boldsymbol{a}\ \check{\wedge}\,\dfrac{=\dfrac{0}{0\wedge 1}}{=\dfrac{0\vee 1}{1}}}\ \mathsf{sa}\,\chi$; similarly for $\overline{\mathsf{w}}$.

- If $\phi \equiv \dfrac{\begin{array}{c}\psi\\ \vdots\\ A\vee B\end{array}}{=\dfrac{}{B\vee A}}\ \begin{array}{c}\\ \vdots\\ \chi\end{array}$ then $\mathsf{sa}\,\phi \equiv \dfrac{\mathsf{sa}\,\psi}{\vee\check{\vee}\,\dfrac{=\dfrac{\mathsf{sa}\,A}{0\vee\mathsf{sa}\,A}\vee=\dfrac{\mathsf{sa}\,B}{\mathsf{sa}\,B\vee 0}}{=\dfrac{0\vee\mathsf{sa}\,B}{\mathsf{sa}\,B}\vee=\dfrac{\mathsf{sa}\,A\vee 0}{\mathsf{sa}\,A}}}\ \mathsf{sa}\,\chi$; similarly for $A\wedge B = B\wedge A$.

- If $\phi \equiv \dfrac{\begin{array}{c}\psi\\ \vdots\\ A\vee(B\vee C)\end{array}}{=\dfrac{}{(A\vee B)\vee C}}\ \begin{array}{c}\\ \vdots\\ \chi\end{array}$ then $\mathsf{sa}\,\phi \equiv \dfrac{\mathsf{sa}\,\psi}{\vee\check{\vee}\,\dfrac{=\dfrac{\mathsf{sa}\,A}{\mathsf{sa}\,A\vee 0}\vee(\mathsf{sa}\,B\vee\mathsf{sa}\,C)}{(\mathsf{sa}\,A\vee\mathsf{sa}\,B)\vee=\dfrac{0\vee\mathsf{sa}\,C}{\mathsf{sa}\,C}}}\ \mathsf{sa}\,\chi$; similarly for $A\wedge(B\wedge C) = (A\wedge B)\wedge C$.

- If $\phi \equiv =\dfrac{\psi}{\chi}$, where $=$ is an instance of one of the equalities $A\vee\mathsf{f} = A$, $A\wedge\mathsf{t} = A$, $\mathsf{t}\vee\mathsf{t} = \mathsf{t}$, or $\mathsf{f}\wedge\mathsf{f} = \mathsf{f}$, then $\mathsf{sa}\,\phi \equiv =\dfrac{\mathsf{sa}\,\psi}{\mathsf{sa}\,\chi}$.

We say that derivations in rKDT-OD and KDT-OD are *strictly linear* because each inference step is linear (contrary to, for example, derivations

with unit-equality steps). By extension, we also call derivations in $\mathsf{rKDT\text{-}ODS}$ and $\mathsf{KDT\text{-}ODS}$ strictly linear. If explicit substitutions are present, we do not have the usual linearity in the sense of term rewriting. However, we can consider substitutions as simply a notation to compress the derivations. Substitutions do not interfere with the logic of the derivation and do not affect its normalisation properties, as we show in the rest of the thesis. Therefore considering those derivations strictly linear is justified. We also call strictly linear the proof systems rKDT and KDT themselves.

Notation 3.3.9. We denote subatomic formulae with boldface letters $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, ...

A more complete treatment of the subatomic logic system for propositional classical logic endowed with decision trees and its proof theory can be found in [5], where the proof system rKDTEq is called $\mathsf{DT}^{\mathsf{sa}}$. $\mathsf{DT}^{\mathsf{sa}}$ does not contain variables, while KDTEq and its variants do because we use them in a formalism with substitutions, *i.e.*, ODS and further generalisations of it. This is not a substantial difference from the point of view of the normalisation and complexity properties of the systems with equality because the inference rules are the same. There might be differences in the semantics, of course, but the natural interpretations only give us conservative extensions of propositional classical logic with decision trees.

Remark 3.3.10. The naming scheme for rules in subatomic logic systems is designed to convey three pieces of information: which two connectives are concerned, and in which corner the rule is saturated. For example, the name $\alpha\widehat{\beta}$ tells us that the connectives are α and β, and that the top-right instance of β is saturated up, so that we know the rule is

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

The name $\check{\wedge}\boldsymbol{a}$ tells us that the connectives are $\wedge$ and $\boldsymbol{a}$, and that the bottom-left instance of $\wedge$ is saturated down, so that we know the rule is

$$\check{\wedge}\boldsymbol{a}\,\frac{(w\ \boldsymbol{a}\ x)\wedge(y\ \boldsymbol{a}\ z)}{(w\vee y)\ \boldsymbol{a}\ (x\wedge z)}\quad .$$

Therefore, the name $\boldsymbol{a}\check{\wedge}$ is not equivalent to $\boldsymbol{a}\vee$, which is indeed not a name, even though $\check{\wedge}=\vee$, because in rule names $\widehat{\ }$ and $\check{\ }$ are considered as decorations. This naming scheme leads to collisions; for example, $\wedge\check{\vee}$ and $\vee\widehat{\wedge}$ both

denote the medial rule

$$\frac{(w \wedge x) \vee (y \wedge z)}{(w \vee y) \wedge (x \vee z)} \quad .$$

Remark 3.3.11. The equality relation $=$ of $\mathsf{KDTEq}^{=}$ subsumes expansion, in the sense that $\mathsf{fl}\, \boldsymbol{A} = \boldsymbol{A}$ and if $\frac{\phi}{\psi}$ is a derivation in $\mathsf{KDTEq}^{=}$, then $=\frac{\phi}{\psi}$ also is.

The reason why we make expansion part of the equality relation is that doing this simplifies the technical treatment of the abstraction map defined in Chapter 4.

Remark 3.3.12. An inference step might be classified both as a rule step and an equality step, for example

$$\hat{\vee}\wedge \frac{(\boldsymbol{A} \wedge 1) \wedge (1 \vee 0)}{(\boldsymbol{A} \wedge 1) \vee (1 \wedge 0)} \qquad \text{and} \qquad =\frac{(\boldsymbol{A} \wedge 1) \wedge (1 \vee 0)}{(\boldsymbol{A} \wedge 1) \vee (1 \wedge 0)} \quad .$$

Notice also that the following inference steps are valid:

$$\hat{\vee}\wedge \frac{\boldsymbol{A} \wedge 1}{\boldsymbol{A} \vee 0} \quad , \qquad =\frac{\boldsymbol{A} \wedge 1}{\boldsymbol{A} \vee 0} \qquad \text{and} \qquad =\frac{\boldsymbol{A} \wedge 1}{\boldsymbol{A} \vee 0} \quad .$$

The following two propositions are straightforward.

Proposition 3.3.13. *In Systems* $\mathsf{rKDTEq\text{-}ODS}$, $\mathsf{KDTEq\text{-}ODS}$ *and* $\mathsf{KDTEq}^{=}\mathsf{\text{-}ODS}$, *for every inference rule* r *and formulae* $\boldsymbol{A}$ *and* $\boldsymbol{B}$, *if* $(\boldsymbol{A}, \boldsymbol{B}) \in r$ *then* $(\mathsf{fl}\, \boldsymbol{A}, \mathsf{fl}\, \boldsymbol{B}) \in r$.

Proposition 3.3.14. *If* $\langle \boldsymbol{A}|x\rangle\, \boldsymbol{B}$ *is a formula in* $\mathsf{KDTEq}^{=}$ *such that* $[\boldsymbol{A}|x]\, \boldsymbol{B} \not\equiv \boldsymbol{B}$ *and either* $\langle \boldsymbol{A}|x\rangle\, \boldsymbol{B} = 0$ *or* $\langle \boldsymbol{A}|x\rangle\, \boldsymbol{B} = 1$, *then either* $\boldsymbol{A} = 0$ *or* $\boldsymbol{A} = 1$.

Note that the above proposition would not be true if there were equalities of the kind $x \wedge 0 = 0$.

Definition 3.3.15. The equivalence relation $=$ over derivations of $\mathsf{KDTEq}^{=}\mathsf{\text{-}ODS}$ is obtained by closing by reflexivity, symmetry and transitivity the following relations:

- $\phi\{\boldsymbol{A}\} = \phi\{\boldsymbol{B}\}$ where $\boldsymbol{A} = \boldsymbol{B}$;

- $\phi\{\psi\} = \phi\left\{=\dfrac{\boldsymbol{C}}{\psi}\right\}$ where $\boldsymbol{C} = \mathsf{pr}\,\psi$;
- $\phi\{\psi\} = \phi\left\{=\dfrac{\psi}{\boldsymbol{C}}\right\}$ where $\boldsymbol{C} = \mathsf{cn}\,\psi$.

Chapter 4

Abstraction and Structural Equivalence

In this section, we define a notion of abstraction for derivations. An abstraction is a map that transforms derivations into derivations and has two properties:

- normalisation is preserved through the abstraction, and
- all derivations that are mapped by the abstraction to the same derivation (or a suitable equivalence class) belong to a natural equivalence class.

In this context, 'natural' means that the abstraction captures some property of independent interest, typically related to normalisation, semantics and complexity. The idea is illustrated in Figure 4.1, using a permutation of inference steps as an example of a typical normalisation case in proof theory. We suppose that we have a set of derivations that we call 'abstract', on the right, whose derivations we can normalise by permuting the inference steps χ_1 and ω_1. There exists another set of derivations, which we call 'concrete', on the left, whose derivations contain more information than those in the abstract set. We want to make sure that normalisation in the abstract set can be recovered from the concrete set. One reason to do all that is that normalisation in the concrete set could be simpler, for example, because the proof system adopted in that set is more regular. Typically, many different derivations in the concrete set map to the same derivation in the abstract one, and we want to collect all those concrete derivations in an equivalence class.

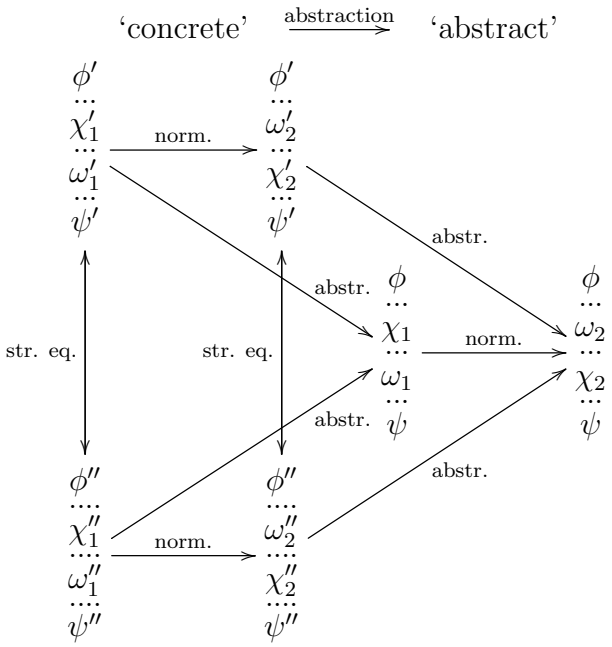


Figure 4.1: Example of an abstraction map (abstr.) from a 'concrete' to an 'abstract' set of derivations, and the induced structural equivalence (str. eq.), when normalising derivations by permuting inference steps (norm.).

Hopefully, if the abstraction map is well conceived, the induced equivalence, which we call 'structural', is natural.

One example of abstraction is the so-called 'interpretation' map from the subatomic system SAKS (which would be considered 'concrete') to SKS ('abstract'), studied in [3]. SAKS is a subsystem of KDTEq where atom connectives cannot be nested and no variables occur. Its interpretation into SKS consists of mapping subatomic formulae built around atom connectives into standard atoms, as we have seen already. The same standard atom can be obtained from infinitely many expressions involving the corresponding atom connective and boolean expression of units, but that does not make a difference for normalisation. The induced structural equivalence is natural from the point of view of the standard interpretation of propositional logic formulae.

Somewhat similarly, in this thesis, we build an abstraction map that collapses unit expressions in a subatomic system to their minimal form. Compared to the interpretations mentioned above, this new map has two additional features: it flattens certain explicit substitutions and it collapses vertical equalities when they are identities (see Figure 4.2). The abstract derivations belong to a subatomic system where inference rules work modulo equality. This allows us to clarify how the large amount of logical material injected into derivations to eliminate unit equalities affects neither normalisation nor complexity at a natural level of abstraction. Concerning semantics, the abstraction is natural because it preserves logical equivalence.

The notions of abstraction and structural equivalence that we present in this section are to be considered a first approximation of more general notions to be developed in the future, alongside the development of the deep-inference proof theory of first and higher order, classical and non-classical, logics. However, the specific abstraction map and structural equivalence that we define in this section are sufficient to prove the results of this work.

4.1 Definitions

As we wrote, an abstraction map aims to respect whatever notion of normalisation we might have. This is a vague objective because normalisation can be any manipulation of derivations – a very broad concept. That said, by looking at the way we compose derivations, we note that horizontal composition is free and therefore does not represent a problem, but vertical composition is

constrained, and that is where we must concentrate our efforts. The vertical composition of two derivations is constrained in three different ways:

- synchronal composition, by which premise and conclusion match;
- composition by expansion, by which the flat expansions of premise and conclusion match;
- composition by inference, by which premise and conclusion belong to an inference rule.

Equivalence relations on formulae, in particular those generated by unit equalities, affect the linearity of derivations, and are the prime target of our investigation. Therefore, we cannot hope for a notion of abstraction that is independent of proof systems. For example, one depending only on synchronal composition would be too weak; Remark 4.2.8 shows that the abstraction map in this section would not be possible.

We proceed by first defining what we mean by vertical composition in a way that is very natural in the context of this thesis but could be modified in future works. It encompasses all three composition methods outlined above. Based on vertical composition, we then define the abstraction map.

Definition 4.1.1. Given a proof system $\mathcal{S}$-$\mathcal{O}$, we say that its derivations ψ and χ can be *vertically composed* if $\mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi$, or if $\mathsf{fl}\,\mathsf{cn}\,\psi \equiv \mathsf{fl}\,\mathsf{pr}\,\chi$, or if $(\mathsf{cn}\,\psi, \mathsf{pr}\,\chi)$ belongs to an inference rule of $\mathcal{S}$-$\mathcal{O}$; we denote this by

$$\mathcal{S}\text{-}\mathcal{O} \begin{array}{c} \psi \\ \approx \\ \chi \end{array} ,$$

where the indication of the proof system will be omitted when clear from the context.

Definition 4.1.2. Given a proof system $\mathcal{S}$-$\mathcal{O}$, an *abstraction in* $\mathcal{S}$-$\mathcal{O}$ is a map $a\colon \mathcal{D}_{\mathcal{S}\text{-}\mathcal{O}} \to \mathcal{D}_{\mathcal{S}\text{-}\mathcal{O}}$ such that, for all derivations ψ and χ:

- $a(\psi) \equiv a(a(\psi))$;
- if $\begin{array}{c} \psi \\ \approx \\ \chi \end{array} \in \mathcal{D}_{\mathcal{S}\text{-}\mathcal{O}}$ then $\begin{array}{c} a(\psi) \\ \approx\approx \\ a(\chi) \end{array} \in \mathcal{D}_{\mathcal{S}\text{-}\mathcal{O}}$;

- if ψ and χ can be vertically composed as $\begin{array}{c}\psi\\ \approx\\ \chi\end{array}$ (resp., as $\begin{array}{c}\chi\\ \approx\\ \psi\end{array}$), for all ψ' such that $a(\psi') \equiv a(\psi)$, there exists a χ' such that:
  - $a(\chi') \equiv a(\chi)$ and
  - χ' can be vertically composed with ψ' as $\begin{array}{c}\psi'\\ \approx\\ \chi'\end{array}$ (resp., as $\begin{array}{c}\chi'\\ \approx\\ \psi'\end{array}$).

We say that two abstractions a and a' are *independent* if they commute.

We now define structural equivalence. We could define it by using the identity of derivations as the canonical element of the equivalence class. In other words, if two derivations, say ϕ and ψ, map to the same, canonical, derivation, then they are in the equivalence class. This could be done, but the price to pay would be a high level of technical complexity. We choose a different method, that, in our case, simplifies matters a lot. We require that $a(\phi) = a(\psi)$, i.e., we map ϕ and ψ to the equivalence class generated by the equality relation on derivations of Definition 3.3.15. This is a natural thing to do and deals in a very simple way with certain situations; Remark 4.2.8 shows an example illustrating this point. One way of looking at this design choice is that we add one degree of freedom when defining an abstraction map.

Definition 4.1.3. Given an abstraction a in $\mathcal{S}$-$\mathcal{O}$, we say that derivations ψ and χ are *structurally equivalent for* a if $a(\psi) = a(\chi)$, whenever $\mathcal{S}$-$\mathcal{O}$ has an equality $=$, or if $a(\psi) \equiv a(\chi)$, in case $\mathcal{S}$-$\mathcal{O}$ has no equality.

Example 4.1.4. A simple abstraction can be defined from derivations in SKSg to derivations in SKS by reducing inference steps into their so-called atomic form using the sw, $\overline{\mathsf{sw}}$ and m rules. For example,

$$\mathsf{i}\,\frac{\mathsf{t}}{(A \wedge B) \vee \left(\bar{A} \vee \bar{B}\right)} \qquad \text{becomes} \qquad \mathsf{sw}\,\frac{=\,\dfrac{\mathsf{t}}{\boxed{\mathsf{i}\,\dfrac{\mathsf{t}}{A \vee \bar{A}}} \vee \boxed{\mathsf{i}\,\dfrac{\mathsf{t}}{B \vee \bar{B}}}}}{(A \wedge B) \vee \left(\bar{A} \vee \bar{B}\right)} \quad ,$$

and

$$\mathsf{c}\frac{(A \wedge B) \vee (A \wedge B)}{A \wedge B} \qquad \text{becomes} \qquad \mathsf{m}\frac{(A \wedge B) \vee (A \wedge B)}{\boxed{\mathsf{c}\frac{A \vee A}{A}} \wedge \boxed{\mathsf{c}\frac{B \vee B}{B}}} \quad ,$$

and those reductions are iterated until the derivations obtained are in SKS. This construction first appeared in [11]. Checking that it satisfies the definition of abstraction is trivial because there is no equality and premise and conclusion are conserved – just note that ψ' can be chosen as ψ. The atomic form contains the same information as all the derivations that map to it (so it is not much of an abstraction in the colloquial sense), but it can play the role of the canonical element of their equivalence class. This fact is exploited by the theory of atomic flows [21, 19], and atomic flows are indeed a proper abstraction over derivations, conserving their normalisation properties.

Even if we do not do so in this thesis, we might want to use independent abstractions to extract different properties from derivations, one for each map, therefore it is important to know whether their composition is an abstraction. The following proposition is trivial to prove.

Proposition 4.1.5. *The composition of two independent abstractions is an abstraction.*

4.2 Abstraction for Strictly Linear Systems

We now define an abstraction map in $\mathsf{KDTEq}^{=}\text{-}\mathsf{ODS}$ that collapses as many units as possible via unit equalities. In particular, subderivations whose every section is mapped to the same unit are compressed to a single instance of that unit. The behaviour of this map is straightforward. We work through the structure of a derivation and we do two things:

- we eliminate all redundant units, and
- we remove all redundant vertical compositions.

We emphasise the fact that this definition is intuitive because we hope that abstractions will be used routinely to extract properties of interest. The ultimate goal is to simplify normalisation, therefore it would not be useful if the difficulty in dealing with normalisation would be transferred to the dealing with abstraction.

$$\mathsf{kdt}\, A \equiv A \quad \text{for } A \in \mathcal{V} \cup \mathcal{U} \tag{4.1}$$

$$\mathsf{kdt}(\psi\ \alpha\ \chi) \equiv \begin{cases} \mathsf{kdt}\,\psi & \text{if } \alpha \equiv \vee \text{ and } \mathsf{kdt}\,\chi \equiv \mathsf{0}, \text{ or if } \alpha \equiv \wedge \text{ and } \mathsf{kdt}\,\chi \equiv \mathsf{1} \\ \mathsf{kdt}\,\chi & \text{if } \alpha \equiv \vee \text{ and } \mathsf{kdt}\,\psi \equiv \mathsf{0}, \text{ or if } \alpha \equiv \wedge \text{ and } \mathsf{kdt}\,\psi \equiv \mathsf{1} \\ \mathsf{0} & \text{if } \alpha \in \{\wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\} \text{ and } \mathsf{kdt}\,\psi \equiv \mathsf{kdt}\,\chi \equiv \mathsf{0} \\ \mathsf{1} & \text{if } \alpha \in \{\vee, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\} \text{ and } \mathsf{kdt}\,\psi \equiv \mathsf{kdt}\,\chi \equiv \mathsf{1} \\ \mathsf{kdt}\,\psi\ \alpha\ \mathsf{kdt}\,\chi & \text{otherwise} \end{cases} \tag{4.2}$$

$$\mathsf{kdt}(\langle\psi|x\rangle\ \chi) \equiv \begin{cases} \mathsf{kdt}([\mathsf{0}|x]\ \chi) & \text{if } \mathsf{kdt}\,\psi \equiv \mathsf{0} \\ \mathsf{kdt}([\mathsf{1}|x]\ \chi) & \text{if } \mathsf{kdt}\,\psi \equiv \mathsf{1} \\ \langle \mathsf{kdt}\,\psi|x\rangle\ \mathsf{kdt}\,\chi & \text{otherwise} \end{cases} \tag{4.3}$$

$$\mathsf{kdt}\boxed{\begin{matrix}\psi\\ \sim\\ \chi\end{matrix}} \equiv \begin{cases} \mathsf{kdt}\boxed{\begin{matrix}\psi\\ \cdots\\ \chi\end{matrix}} & \text{if } \mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi \\ \begin{matrix}\mathsf{kdt}\,\psi\\ \cdots\cdots\\ \mathsf{kdt}\,\chi\end{matrix} & \text{if } \mathsf{cn}\,\mathsf{kdt}\,\psi \equiv \mathsf{pr}\,\mathsf{kdt}\,\chi \text{ and } \mathsf{cn}\,\psi \not\equiv \mathsf{pr}\,\chi \\ = \dfrac{\mathsf{kdt}\,\psi}{\mathsf{kdt}\,\chi} & \text{otherwise} \end{cases} \tag{4.4}$$

$$\mathsf{kdt}\boxed{r\,\begin{matrix}\psi\\ =\\ \chi\end{matrix}} \equiv \begin{cases} \mathsf{kdt}\boxed{\begin{matrix}\psi\\ \cdots\\ \chi\end{matrix}} & \text{if } \mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi \\ \begin{matrix}\mathsf{kdt}\,\psi\\ \cdots\cdots\\ \mathsf{kdt}\,\chi\end{matrix} & \text{if } \mathsf{cn}\,\mathsf{kdt}\,\psi \equiv \mathsf{pr}\,\mathsf{kdt}\,\chi \text{ and } \mathsf{cn}\,\psi \not\equiv \mathsf{pr}\,\chi \\ r\,\dfrac{\mathsf{kdt}\,\psi}{\mathsf{kdt}\,\chi} & \text{otherwise} \end{cases} \tag{4.5}$$

Figure 4.2: Abstraction map kdt.

Definition 4.2.1. In Figure 4.2, the map

$$\mathsf{kdt}\colon \mathsf{KDTEq}^{=}\text{-}\mathsf{ODS} \to \mathcal{P}(\emptyset, \{\vee, \wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}, \{0, 1\})$$

is defined recursively on the size of the argument.

Remark 4.2.2. Given derivations ψ and χ as in Figure 4.2, it could be that $\mathsf{cn}\,\psi \not\equiv \mathsf{pr}\,\chi$ and $\mathsf{cn}\,\mathsf{kdt}\,\psi \equiv \mathsf{pr}\,\mathsf{kdt}\,\chi$, as, for example, in:

$$\mathsf{kdt}\boxed{\vdots\!=\!\frac{x \wedge 1}{x}} \equiv \boxed{\frac{\mathsf{kdt}\,(x \wedge 1)}{\mathsf{kdt}\,x}} \equiv \boxed{\begin{matrix} x \\ \vdots \\ x \end{matrix}} \equiv x \quad .$$

It could also be that $\mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi$ and $\mathsf{cn}\,\mathsf{kdt}\,\psi \not\equiv \mathsf{pr}\,\mathsf{kdt}\,\chi$, as, for example, in:

$$\mathsf{kdt}\boxed{\vdots\!=\!\frac{x \wedge \boxed{\vee\!\hat{\wedge}\,\frac{(0 \wedge 1) \vee (1 \wedge 0)}{(0 \vee 1) \wedge (1 \vee 0)}}}{x \wedge ((0 \vee 1) \wedge (1 \vee 0))}} \equiv \mathsf{kdt}\boxed{\frac{x \wedge \boxed{\vee\!\hat{\wedge}\,\frac{(0 \wedge 1) \vee (1 \wedge 0)}{(0 \vee 1) \wedge (1 \vee 0)}}}{x \wedge ((0 \vee 1) \wedge (1 \vee 0))}} \equiv x \wedge \boxed{\vee\!\hat{\wedge}\,\frac{0}{1}} \quad .$$

We now study the properties of kdt. We first prove that it is indeed an abstraction and then we characterize the derivations that are collapsed by kdt into a unit. We need two technical lemmas (4.2.3 and 4.2.9) whose proofs are stringent but straightforward case analyses. Again, we emphasise this aspect because everything that concerns an abstraction map must be intuitive and simple.

Lemma 4.2.3. *Let $=$ be the equality relation of System $\mathsf{KDTEq}^{=}$. For any derivation ϕ in $\mathsf{KDTEq}^{=}$-ODS:*

$$\mathsf{pr}\,\mathsf{kdt}\,\phi = \mathsf{pr}\,\phi \qquad \textit{and} \qquad \mathsf{cn}\,\mathsf{kdt}\,\phi = \mathsf{cn}\,\phi \quad .$$

Proof. We prove the equality $\mathsf{pr}\,\mathsf{kdt}\,\phi = \mathsf{pr}\,\phi$ by induction on the size of ϕ. The following are the possible cases, in accordance with the numbering in Figure 4.2.

1. $\phi \in \mathcal{V} \cup \mathcal{U}$. This is the base case. We have $\mathsf{pr}\,\mathsf{kdt}\,\phi \equiv \phi \equiv \mathsf{pr}\,\phi$.

2. There are seven cases:

   (a) $\phi \equiv \psi \vee \chi$ and $\mathsf{kdt}\,\chi \equiv 0$:

   $$\mathsf{pr}\,\mathsf{kdt}\,\phi = \mathsf{pr}\,\mathsf{kdt}\,\psi \vee \mathsf{pr}\,0 \equiv \mathsf{pr}\,\mathsf{kdt}\,\psi \vee \mathsf{pr}\,\mathsf{kdt}\,\chi = \mathsf{pr}\,\psi \vee \mathsf{pr}\,\chi \equiv \mathsf{pr}\,\phi \quad .$$

(b) $\phi \equiv \psi \wedge \chi$ and $\mathsf{kdt}\,\chi \equiv \mathsf{1}$: analogous to Case 2a.

(c) $\phi \equiv \psi \vee \chi$ and $\mathsf{kdt}\,\psi \equiv \mathsf{0}$: analogous to Case 2a.

(d) $\phi \equiv \psi \wedge \chi$ and $\mathsf{kdt}\,\psi \equiv \mathsf{1}$: analogous to Case 2a.

(e) $\phi \equiv \psi\ \alpha\ \chi$, where $\alpha \in \{\wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$ and $\mathsf{kdt}\,\chi \equiv \mathsf{kdt}\,\psi \equiv \mathsf{0}$:

$$\mathsf{pr\,kdt}\,\phi \equiv \mathsf{0} = \mathsf{pr\,kdt}\,\psi\ \alpha\ \mathsf{pr\,kdt}\,\chi = \mathsf{pr}\,\psi\ \alpha\ \mathsf{pr}\,\chi \equiv \mathsf{pr}\,\phi \quad .$$

(f) $\phi \equiv \psi\ \alpha\ \chi$, where $\alpha \in \{\vee, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$ and $\mathsf{kdt}\,\chi \equiv \mathsf{kdt}\,\psi \equiv \mathsf{1}$: analogous to Case 2e.

(g) $\phi \equiv \psi\ \alpha\ \chi$ and none of the above conditions apply:

$$\mathsf{pr\,kdt}\,\phi \equiv \mathsf{pr\,kdt}\,\psi\ \alpha\ \mathsf{pr\,kdt}\,\chi = \mathsf{pr}\,\psi\ \alpha\ \mathsf{pr}\,\chi \equiv \mathsf{pr}\,\phi \quad .$$

3. $\phi \equiv \langle\psi|x\rangle\,\chi$. There are three cases:

(a) $\mathsf{kdt}\,\psi \equiv \mathsf{0}$:

$$\begin{aligned}\mathsf{pr\,kdt}\,\phi \equiv \mathsf{pr\,kdt}([\mathsf{0}|x]\,\chi) = \mathsf{pr}([\mathsf{0}|x]\,\chi) \equiv [\mathsf{pr\,0}|x]\,\mathsf{pr}\,\chi \equiv [\mathsf{pr\,kdt}\,\psi|x]\,\mathsf{pr}\,\chi \\ = [\mathsf{pr}\,\psi|x]\,\mathsf{pr}\,\chi = \langle\mathsf{pr}\,\psi|x\rangle\,\mathsf{pr}\,\chi = \mathsf{pr}\,\phi \quad .\end{aligned}$$

(b) $\mathsf{kdt}\,\psi \equiv \mathsf{1}$: analogous to Case 3a.

(c) $\mathsf{0} \not\equiv \mathsf{kdt}\,\psi \not\equiv \mathsf{1}$:

$$\mathsf{pr\,kdt}\,\phi \equiv \mathsf{pr}(\langle\mathsf{kdt}\,\psi|x\rangle\,\mathsf{kdt}\,\chi) \equiv \langle\mathsf{pr\,kdt}\,\psi|x\rangle\,\mathsf{pr\,kdt}\,\chi = \langle\mathsf{pr}\,\psi|x\rangle\,\mathsf{pr}\,\chi \equiv \mathsf{pr}\,\phi \quad .$$

4. $\phi \equiv \boxed{\begin{matrix}\psi \\ \sim \\ \chi\end{matrix}}$. There are three cases:

(a) $\mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi$:

$$\mathsf{pr\,kdt}\,\phi \equiv \mathsf{pr\,kdt}\boxed{\begin{matrix}\psi \\ \cdots \\ \chi\end{matrix}} = \mathsf{pr}\boxed{\begin{matrix}\psi \\ \cdots \\ \chi\end{matrix}} \equiv \mathsf{pr}\,\psi \equiv \mathsf{pr}\,\phi \quad .$$

(b) $\mathsf{cn\,kdt}\,\psi \equiv \mathsf{pr\,kdt}\,\chi$ and $\mathsf{cn}\,\psi \not\equiv \mathsf{pr}\,\chi$:

$$\mathsf{pr\,kdt}\,\phi \equiv \mathsf{pr}\boxed{\begin{matrix}\mathsf{kdt}\,\psi \\ \cdots\cdots \\ \mathsf{kdt}\,\chi\end{matrix}} \equiv \mathsf{pr\,kdt}\,\psi = \mathsf{pr}\,\psi \equiv \mathsf{pr}\,\phi \quad .$$

(c) Otherwise:

$$\mathsf{pr}\,\mathsf{kdt}\,\phi \equiv \mathsf{pr}\,\boxed{=\frac{\mathsf{kdt}\,\psi}{\mathsf{kdt}\,\chi}} \equiv \mathsf{pr}\,\mathsf{kdt}\,\psi = \mathsf{pr}\,\psi \equiv \mathsf{pr}\,\phi \quad .$$

5. Analogous to Case 4.

A similar argument proves $\mathsf{cn}\,\mathsf{kdt}\,\phi = \mathsf{cn}\,\phi$. □

Corollary 4.2.4. $\mathsf{kdt}\,\mathsf{pr}\,\phi = \mathsf{pr}\,\mathsf{kdt}\,\phi$ *and* $\mathsf{kdt}\,\mathsf{cn}\,\phi = \mathsf{cn}\,\mathsf{kdt}\,\phi$.

Proof. It follows from Lemma 4.2.3. The first part is proved this way:

$$\mathsf{kdt}\,\mathsf{pr}\,\phi \equiv \mathsf{pr}\,\mathsf{kdt}\,\mathsf{pr}\,\phi = \mathsf{pr}\,\mathsf{pr}\,\phi \equiv \mathsf{pr}\,\phi = \mathsf{pr}\,\mathsf{kdt}\,\phi \quad .$$

The second part is proved analogously. □

Proposition 4.2.5. $\mathsf{kdt}\,\phi$ *is in* $\mathsf{KDTEq}^{=}$-ODS.

Proof. We must check that the inference steps created by applying kdt are valid instances of $\mathsf{KDTEq}^{=}$-ODS rules. We refer to the numbering of cases in Figure 4.2. Cases 4.1 to 4.3 do not create inference steps. Cases 4.4 and 4.5 are as follows:

- if $\phi \equiv \boxed{\begin{matrix}\psi \\ \sim \\ \chi\end{matrix}}$ then $\mathsf{cn}\,\mathsf{kdt}\,\psi = \mathsf{cn}\,\psi = \mathsf{fl}\,\mathsf{cn}\,\psi \equiv \mathsf{fl}\,\mathsf{pr}\,\chi = \mathsf{pr}\,\chi = \mathsf{pr}\,\mathsf{kdt}\,\chi$;
- if $\phi \equiv \boxed{\begin{matrix}\psi \\ r= \\ \chi\end{matrix}}$ then $(\mathsf{cn}\,\psi, \mathsf{pr}\,\chi) \in r$, for some rule r of $\mathsf{KDTEq}^{=}$ (including $=$); by Lemma 4.2.3 we have $\mathsf{cn}\,\mathsf{kdt}\,\psi = \mathsf{cn}\,\psi$ and $\mathsf{pr}\,\mathsf{kdt}\,\chi = \mathsf{pr}\,\chi$, therefore $(\mathsf{cn}\,\mathsf{kdt}\,\psi, \mathsf{pr}\,\mathsf{kdt}\,\chi) \in r$.

In both cases, if kdt creates an inference step, it is valid. □

Lemma 4.2.6. $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\mathsf{kdt}\,\phi$.

Proof. Consider $\mathsf{kdt}\,\phi$ and its definition in Figure 4.2. The only non-trivial cases to consider are the synchronal compositions of Cases 4.4 and 4.5 because they rearrange the structure of the derivation, potentially creating new subderivations that could be acted upon by a further application of kdt. We reason on definition of kdt. Let us call 'collapsing conditions' the conditions in Cases 4.2 and 4.3 that are not 'otherwise'. The only way for $\mathsf{kdt}\,\mathsf{kdt}\,\phi$ to not be identical to $\mathsf{kdt}\,\phi$ is if there were a subderivation ω in $\mathsf{kdt}\,\phi$ such that:

- $\mathsf{kdt}\,\omega \equiv 0$ or $\mathsf{kdt}\,\omega \equiv 1$ and one of the collapsing conditions in Case 4.2 or 4.3 is satisfied;
- ω resulted from the synchronal composition of two subderivations of $\mathsf{kdt}\,\phi$, say ω' and ω'', in Case 4.4 or 4.5, *i.e.*,

$$\mathsf{kdt}\,\phi \equiv \phi_1 \left\{ \begin{matrix} \phi_2\{\omega'\} \\ \cdots\cdots \\ \phi_3\{\omega''\} \end{matrix} \right\} \equiv \phi_1 \left\{ \phi_4 \left\{ \begin{matrix} \omega' \\ \cdots \\ \omega'' \end{matrix} \right\} \right\} \equiv \phi_1\{\phi_4\{\omega\}\} \quad .$$

If that were true, ω would be removed from $\mathsf{kdt}\,\phi$ or replaced by a unit, resulting in $\mathsf{kdt}\,\mathsf{kdt}\,\phi \not\equiv \mathsf{kdt}\,\phi$. However, that is impossible because ω' and ω'' would satisfy the collapsing conditions. □

Proposition 4.2.7. *The map* $\mathsf{kdt}\colon \mathsf{KDTEq}^{=}\text{-}\mathsf{ODS} \to \mathsf{KDTEq}^{=}\text{-}\mathsf{ODS}$ *is an abstraction.*

Proof. We check the three conditions of Definition 4.1.2.

- We proved in Lemma 4.2.6 that $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\mathsf{kdt}\,\phi$, for every $\psi \in \mathsf{KDTEq}^{=}\text{-}\mathsf{ODS}$.
- Consider $\overset{\psi}{\underset{\chi}{\approx}}$: because, by Lemma 4.2.3, $\mathsf{cn}\,\mathsf{kdt}\,\psi = \mathsf{cn}\,\psi = \mathsf{fl}\,\mathsf{cn}\,\psi$ and $\mathsf{pr}\,\mathsf{kdt}\,\chi = \mathsf{pr}\,\chi = \mathsf{fl}\,\mathsf{pr}\,\chi$, at least one of the following cases applies:
  - if $\mathsf{cn}\,\psi \equiv \mathsf{pr}\,\chi$ or $\mathsf{fl}\,\mathsf{cn}\,\psi \equiv \mathsf{fl}\,\mathsf{pr}\,\chi$, then $=\dfrac{\mathsf{kdt}\,\psi}{\mathsf{kdt}\,\chi}$ can be built, and possibly also $\begin{matrix} \mathsf{kdt}\,\psi \\ \cdots\cdots \\ \mathsf{kdt}\,\chi \end{matrix}$;
  - if $(\mathsf{cn}\,\psi, \mathsf{pr}\,\chi)$ belongs to an inference rule r, then we can build $r\dfrac{\mathsf{kdt}\,\psi}{\mathsf{kdt}\,\chi}$.
- If we are given $\overset{\psi}{\underset{\chi}{\approx}}$ and ψ' such that $\mathsf{kdt}\,\psi' \equiv \mathsf{kdt}\,\psi$, then choosing $\chi' \equiv \chi$ satisfies the definition. Indeed, by Lemma 4.2.3, $\mathsf{cn}\,\psi' = \mathsf{cn}\,\mathsf{kdt}\,\psi' \equiv \mathsf{cn}\,\mathsf{kdt}\,\psi = \mathsf{cn}\,\psi$. Therefore χ' can be vertically composed with ψ'.

□

Remark 4.2.8. It is not necessarily the case that $a\boxed{\begin{matrix}\phi\\ \cdots\\ \psi\end{matrix}}$ is identical to $\begin{matrix}a\phi\\ \cdots\\ a\psi\end{matrix}$. Consider, for example,

$$\phi \equiv x \wedge \boxed{\vee\hat{\wedge}\frac{(0 \wedge 1) \vee (1 \wedge 0)}{(0 \vee 1) \wedge (1 \vee 0)}} \qquad \text{and} \qquad \mathsf{kdt}\,\phi \equiv x \wedge \boxed{\vee\hat{\wedge}\frac{0}{1}} \quad ;$$

if we take $\psi \equiv \mathsf{cn}\,\phi$, we have that $\mathsf{cn}\,\mathsf{kdt}\,\phi \equiv x \wedge 1 \not\equiv \mathsf{pr}\,\mathsf{kdt}\,\psi \equiv x$. This example also shows that $a\boxed{\begin{matrix}\phi\\ \approx\\ \psi\end{matrix}}$ is not necessarily identical to $a\boxed{\begin{matrix}a\phi\\ \approx\\ a\psi\end{matrix}}$. In fact,

$$\mathsf{kdt}\boxed{\begin{matrix}\phi\\ \cdots\\ \psi\end{matrix}} \equiv x \wedge \boxed{\vee\hat{\wedge}\frac{0}{1}} \not\equiv \boxed{=\frac{x \wedge \boxed{\vee\hat{\wedge}\frac{0}{1}}}{x}} \equiv \mathsf{kdt}\boxed{=\frac{x \wedge \boxed{\vee\hat{\wedge}\frac{0}{1}}}{x}} \equiv \mathsf{kdt}\boxed{\begin{matrix}\mathsf{kdt}\,\phi\\ \approx\\ \mathsf{kdt}\,\psi\end{matrix}} \quad .$$

if we take $\psi \equiv \mathsf{cn}\,\phi$, we have that $\mathsf{cn}\,\mathsf{kdt}\,\phi \equiv x \wedge 1 \not\equiv \mathsf{pr}\,\mathsf{kdt}\,\psi \equiv x$. However, note that

$$\mathsf{kdt}\boxed{\begin{matrix}\phi\\ \cdots\\ \psi\end{matrix}} \equiv x \wedge \boxed{\vee\hat{\wedge}\frac{0}{1}} = \boxed{=\frac{x \wedge \boxed{\vee\hat{\wedge}\frac{0}{1}}}{x}} \equiv \mathsf{kdt}\boxed{=\frac{\mathsf{kdt}\,\phi}{\mathsf{kdt}\,\psi}} \quad ,$$

where we used Definition 3.3.15. Therefore $\begin{matrix}\phi\\ \cdots\\ \psi\end{matrix}$ and $=\frac{\mathsf{kdt}\,\phi}{\mathsf{kdt}\,\psi}$ are structurally equivalent.

The above remark suggests that some conditions involving equality might be added to the definition of abstraction, for example,

$$a\boxed{\begin{matrix}\phi\\ \cdots\\ \psi\end{matrix}} = a\boxed{\begin{matrix}a\phi\\ \cdots\\ a\psi\end{matrix}} \quad .$$

For the time being, we prefer to keep the notion of abstraction as independent as possible from proof systems.

We have shown that the map kdt is an abstraction. We now characterise those derivations that are collapsed by kdt into a single unit. This is useful in

the rest of the thesis to claim that strictly linear derivations can be normalised the same way as their non-strict counterparts. This lemma is proved in detail but the proof is straightforward. We omit similar proofs to similar statements in the rest of the thesis, one reason being that the abstraction map is supposed to be straightforward and its use should simplify (not complicate) whatever other investigation it supports.

Lemma 4.2.9. *Let $=$ be the equality relation of System* $\mathsf{KDTEq}^{=}$*. For any derivation ϕ in* $\mathsf{KDTEq}^{=}$*-*ODS *the following statements hold:*

- $\mathsf{kdt}\,\phi \equiv \mathsf{0}$ *if and only if every section $\boldsymbol{A}$ of ϕ is such that $\boldsymbol{A} = \mathsf{0}$.*
- $\mathsf{kdt}\,\phi \equiv \mathsf{1}$ *if and only if every section $\boldsymbol{A}$ of ϕ is such that $\boldsymbol{A} = \mathsf{1}$.*

Proof. We prove the first statement by induction on the size of ϕ. The following are the possible cases, in accordance with the numbering in Figure 4.2. We say that a derivation ϕ is $\mathsf{0}$-*good* (resp., $\mathsf{1}$-*good*) if every section $\boldsymbol{A}$ of ϕ is such that $\boldsymbol{A} = \mathsf{0}$ (resp., $\boldsymbol{A} = \mathsf{1}$).

1. $\phi \in \mathcal{V} \cup \mathcal{U}$. This is the base case and the statement is trivially true.
2. There are seven cases:
   (a) $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,(\psi \vee \chi) \equiv \mathsf{kdt}\,\psi$ and $\mathsf{kdt}\,\chi \equiv \mathsf{0}$, which implies that χ is $\mathsf{0}$-good. If $\mathsf{kdt}\,\phi \equiv \mathsf{0}$ then $\mathsf{kdt}\,\psi \equiv \mathsf{0}$ then ψ is $\mathsf{0}$-good then $\phi \equiv \psi \vee \chi$ is $\mathsf{0}$-good. Conversely, if ϕ is $\mathsf{0}$-good then ψ is $\mathsf{0}$-good then $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\psi \equiv \mathsf{0}$.
   (b) $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,(\psi \wedge \chi) \equiv \mathsf{kdt}\,\psi$ and $\mathsf{kdt}\,\chi \equiv \mathsf{1}$, which implies that χ is $\mathsf{1}$-good. If $\mathsf{kdt}\,\phi \equiv \mathsf{0}$ then $\mathsf{kdt}\,\psi \equiv \mathsf{0}$ then ψ is $\mathsf{0}$-good then $\phi \equiv \psi \wedge \chi$ is $\mathsf{0}$-good. Conversely, if ϕ is $\mathsf{0}$-good then ψ is $\mathsf{0}$-good then $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\psi \equiv \mathsf{0}$.
   (c) $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,(\psi \vee \chi) \equiv \mathsf{kdt}\,\chi$ and $\mathsf{kdt}\,\psi \equiv \mathsf{0}$: analogous to Case 2a.
   (d) $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,(\psi \wedge \chi) \equiv \mathsf{kdt}\,\chi$ and $\mathsf{kdt}\,\psi \equiv \mathsf{1}$: analogous to Case 2b.
   (e) $\phi \equiv \psi\,\alpha\,\chi$, where $\alpha \in \{\wedge, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$ and $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\chi \equiv \mathsf{kdt}\,\psi \equiv \mathsf{0}$. Therefore ψ and χ are $\mathsf{0}$-good and then ϕ is $\mathsf{0}$-good.
   (f) $\phi \equiv \psi\,\alpha\,\chi$, where $\alpha \in \{\vee, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots\}$ and $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\chi \equiv \mathsf{kdt}\,\psi \equiv \mathsf{1}$: analogous to Case 2e.

(g) $\phi \equiv \psi\, \alpha\, \chi$ and none of the above conditions apply, *i.e.*, $\mathsf{kdt}\,\phi \not\equiv 0$ (resp., $\mathsf{kdt}\,\phi \not\equiv 1$), $\alpha \not\equiv \vee$ (resp., $\alpha \not\equiv \wedge$) and one of ψ and χ is not 0-good (resp., 1-good); therefore ϕ is not 0-good (resp., 1-good).

3. $\phi \equiv \langle \psi | x \rangle\, \chi$. We assume that x appears free in χ, otherwise $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\chi$ and the statement follows from the induction hypothesis. There are three cases:

   (a) If $\mathsf{kdt}\,\psi \equiv 0$, and then ψ is 0-good, we have:

   $$\mathsf{kdt}\,\phi \equiv \mathsf{kdt}([0|x]\,\chi) \equiv 0 \qquad \text{iff} \qquad [0|x]\,\chi \text{ is 0-good} \qquad \text{iff} \qquad \phi \text{ is 0-good.}$$

   (b) If $\mathsf{kdt}\,\psi \equiv 1$, and then ψ is 1-good, we have:

   $$\mathsf{kdt}\,\phi \equiv \mathsf{kdt}([1|x]\,\chi) \equiv 0 \qquad \text{iff} \qquad [1|x]\,\chi \text{ is 0-good} \qquad \text{iff} \qquad \phi \text{ is 0-good.}$$

   (c) Otherwise, $\mathsf{kdt}\,\phi \not\equiv 0$ by the definition of kdt, and ϕ is not 0-good because the converse would contradict Proposition 3.3.14.

4. $\phi \equiv \boxed{\begin{matrix}\psi \\ \sim \\ \chi\end{matrix}}$. There are three cases:

   (a) If $\mathsf{cn}\,\psi \equiv \mathsf{kdt}\,\chi$, we have:

   $$\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\boxed{\begin{matrix}\psi \\ \cdots \\ \chi\end{matrix}} \equiv 0 \qquad \text{iff} \qquad \phi \equiv \boxed{\begin{matrix}\psi \\ \sim \\ \chi\end{matrix}} \text{ is 0-good.}$$

   (b) Otherwise, if $\mathsf{cn}\,\mathsf{kdt}\,\psi \equiv \mathsf{pr}\,\mathsf{kdt}\,\chi$, we have:

   $$\mathsf{kdt}\,\phi \equiv \boxed{\begin{matrix}\mathsf{kdt}\,\psi \\ \cdots\cdots \\ \mathsf{kdt}\,\chi\end{matrix}} \equiv 0 \qquad \text{iff} \qquad \mathsf{kdt}\,\psi \equiv \mathsf{kdt}\,\chi \equiv 0 \qquad \text{iff} \qquad \phi \equiv \boxed{\begin{matrix}\psi \\ \sim \\ \chi\end{matrix}} \text{ is 0-good.}$$

   (c) Otherwise, $\mathsf{kdt}\,\phi \equiv \boxed{= \dfrac{\mathsf{kdt}\,\psi}{\mathsf{kdt}\,\chi}} \not\equiv 0$ and ϕ is not 0-good, otherwise ψ and χ would be 0-good and $\mathsf{cn}\,\mathsf{kdt}\,\psi \equiv \mathsf{pr}\,\mathsf{kdt}\,\chi \equiv 0$.

5. Analogous to Case 4.

An analogous argument proves the second statement. □

The map kdt that we give here is sufficient to show that the transformation by which we eliminate the unit-equality inference steps in Chapter 6 does not alter the structure of the derivation in any meaningful way. The definitions that we give in this chapter are only intended to be preliminary: it could be that a different notion of abstraction map is necessary in the future, or that an map other than kdt is needed to compare derivations inside $\mathsf{KDTEq}^{=}$-ODS. As we mention in the introduction, explicit substitutions are designed to abstract away the bureaucracy of where precisely one derivation occurs inside a thread of another; we hope to define a map which can capture this abstraction.

Chapter 5

The Eversion Lemma

In this chapter, we prove a technical lemma that we call 'eversion'. It is at the core of our results.

The problem that we want to solve is illustrated by the following example. Suppose that we have the open formula $A \vee B$ and we know that $\pi B = \mathsf{f}$ for some substitution π that assigns units to variables. We will often assume that there is a global substitution π that assigns units to variables in such a way that we can exclusively operate on open formulae when transforming derivations. We want to mimic the non-linear inference step

$$\frac{A}{A \vee B}$$

via a strictly linear subatomic derivation, without disrupting the structure of the surrounding derivation, in a sense that might be captured precisely by an abstraction map.

The subatomic shape only allows us to merge two formulae if they have the same structure modulo saturation, but we cannot assume that this is the case because A and B have, in principle, nothing to do with each other. The idea of the Eversion Lemma is to consider a copy of A where all its leaves have been substituted with copies of B. This formula is still interpreted as the disjunctive unit in many logics. Remarkably, by a conceptually simple construction, it is possible to derive a formula where the roles of A and B are reversed: the external structure is that of B and A's structure is replicated in its leaves. In other words, what is outside gets inside and vice versa; this

is what we approximately get:

$$\begin{array}{c} [B|v]_{v \text{ in } A}\, A \\ \| \\ [A|v]_{v \text{ in } B}\, B \end{array} .$$

For example, and paying attention to the variables individually, we can see the construction as distributing the leaves of the formulae like this:

$$\begin{array}{c} (((w\ \alpha\ x)\ \beta\ y)\ \gamma\ z)\ \delta\ ((((w'\ \alpha\ x')\ \beta\ y')\ \gamma\ z')\ \boldsymbol{a}\ (((w''\ \alpha\ x'')\ \beta\ y'')\ \gamma\ z'')) \\ \| \\ (((w\ \delta\ (w'\ \boldsymbol{a}\ w''))\ \alpha\ (x\ \delta\ (x'\ \boldsymbol{a}\ x'')))\ \beta\ (y\ \delta\ (y'\ \boldsymbol{a}\ y''))\ \gamma\ (z\ \delta\ (z'\ \boldsymbol{a}\ z'')) \end{array} ,$$

where we are taking A to be the open formula $(v\ \delta\ v')\ \boldsymbol{a}\ v''$ and B to be instances of the open formula $((w\ \alpha\ x)\ \beta\ y)\ \gamma\ z$.

This only works under certain conditions, so the above constructions are not precise, but serve to show the idea. The conditions under which eversion works are flexible enough that it can be used as a powerful tool in the situations that we have encountered.

The problem mentioned above is dealt with in the following way. The derivation on the left is transformed into the derivation on the right:

$$\dfrac{\pi K\left\{\dfrac{\overset{\pi\sigma\phi}{\vdots}}{\sigma A \vee B}\right\}}{\pi\psi} \quad \longrightarrow \quad \dfrac{\pi K\left\{\begin{array}{c} \sigma\left[v \vee B^{\mathsf{f}}\middle|v\right]_{v \text{ in } A} A \\ \| \\ \boxed{\sigma A \vee \begin{array}{c}\boxed{\left[B^{\mathsf{f}}\middle|v\right]_{v \text{ in } A} A} \\ \| \\ \boxed{\left[A^{\mathsf{f}}\middle|v\right]_{v \text{ in } B} B}\end{array}} \end{array}\right\}}{\pi\left[A^{\mathsf{f}}\middle|v\right]_{v \text{ in } B}\psi} \quad \overset{\pi\sigma\left[v \vee B^{\mathsf{f}}\middle|v\right]_{v \text{ in } A}\phi}{\vdots} ,$$

where A^{f} and B^{f} are A and B where every variable has been replaced by a variable equal to f under π and σ is the substitution resulting from propagating previous instances of this transformation. By doing this transformation, the inference step becomes linear. The price to pay is the propagation of substitutions up and down the derivation but this does not affect its structure and is transparent to the abstraction map kdt.

By doing this transformation repeatedly, for example, to eliminate all the nonlinear inference steps, we blow up the size of the derivation exponentially. However, the structures travelling up and down the derivations are taken

from the initial derivation (A and B in the example). They do not blow up; they are simply composed repeatedly via substitutions. There is then hope to avoid an exponential explosion of the size of the derivation by employing explicit substitutions. That this is possible is the main result of Chapter 6.

5.1 Merge

The first step to getting the Eversion Lemma is to prove what we call the 'Merge Lemma', which is a generalisation of a foundational construction in deep inference. We state the Merge and Eversion Lemmas for System rDT, which does not commit to any special logical language. We do this because the definitions involved are not specific to any particular logic and to draw more attention to which rules are necessary for a proof system to verify the Merge and Eversion Lemmas.

In this work, we use the lemmas for System rKDT, and we show how the abstraction kdt is affected by the constructions in the lemmas. We also assess the complexity of our constructions.

We introduce the notions of 'down-' and 'up-conjugacy'. Given a formula A, we can obtain two formulae from it, A' and A'', by copying A verbatim, except that each connective occurrence is saturated down either in A' or A'', which together are called 'down-conjugates' of A. In turn, A' and A'' can each produce down-conjugates of themselves, recursively.

There are two versions of conjugacy, the 'rigid' one requires that each connective occurrence is left intact either in A' or A'' (but not in both), while the 'lax' version allows for it to be saturated in both. 'Up-conjugacy' is similarly defined. This notion is key to understanding the Eversion Lemma.

Definition 5.1.1. For $m \geq 1$, let $A_1\{\ \}_{1\ldots n}, \ldots, A_m\{\ \}_{1\ldots n}$ be skeleton contexts that are obtained from skeleton context $A\{\ \}_{1\ldots n}$ by replacing connective occurrences with other connective occurrences, which we call *corresponding*. We say that

- $A_1\{\ \}_{1\ldots n}, \ldots, A_m\{\ \}_{1\ldots n}$ are *rigid down-conjugates* (resp., *rigid up-conjugates*) of $A\{\ \}_{1\ldots n}$ if for each connective occurrence α in $A\{\ \}_{1\ldots n}$, one corresponding occurrence in any of $A_1\{\ \}_{1\ldots n}, \ldots, A_m\{\ \}_{1\ldots n}$ is α and all the others are $\breve{\alpha}$ (resp., $\widehat{\alpha}$).

- $A_1\{\ \}_{1\ldots n}, \ldots, A_m\{\ \}_{1\ldots n}$ are *lax down-conjugates* (resp., *lax up-conjugates*) of $A\{\ \}_{1\ldots n}$ if for each connective occurrence α in $A\{\ \}_{1\ldots n}$, at most one

corresponding occurrence in any of $A_1\{\ \}_{1\ldots n}$, $\ldots$, $A_m\{\ \}_{1\ldots n}$ is α and at least one and all the others are $\breve{\alpha}$ (resp., $\widehat{\alpha}$);

We extend the notions of conjugacy to formulae obtained from filling corresponding holes of conjugate contexts with the same atoms, units or variables.

Rigid conjugacy can be summarised by the slogan 'one unsaturated, all the rest saturated', and lax conjugacy by 'at most one unsaturated, at least one saturated'.

Example 5.1.2. In KDT, consider the formula $\boldsymbol{A} \equiv (w \wedge x) \wedge (y \vee z)$. The formulae

$$\begin{aligned} \boldsymbol{A}' &\equiv (w \vee x) \wedge (y \vee z) \\ \boldsymbol{A}'' &\equiv (w \wedge x) \vee (y \vee z) \end{aligned}$$

form a pair of (rigid and lax) down-conjugates for $\boldsymbol{A}$. In turn, the formulae

$$\begin{aligned} \boldsymbol{A}'' &\equiv (w \wedge x) \vee (y \vee z) \\ \boldsymbol{A}''' &\equiv (w \vee x) \vee (y \vee z) \end{aligned}$$

form a pair of (rigid and lax) down-conjugates for $\boldsymbol{A}''$, so that the formulae $\boldsymbol{A}'$, $\boldsymbol{A}''$, and $\boldsymbol{A}'''$ form a trio of (rigid and lax) down-conjugates for $\boldsymbol{A}$.

The formulae $\boldsymbol{A}'$ and $\boldsymbol{A}'''$ form a pair of lax down-conjugates for $\boldsymbol{A}$ but not a pair of rigid down-conjugates for $\boldsymbol{A}$, because in both the leftmost instance of $\wedge$ in $\boldsymbol{A}$ has been saturated down.

Example 5.1.3. In KDT, consider

$$\begin{aligned} \boldsymbol{A}\{\ \}_{1\ldots5} &\equiv \langle \{\ \} \wedge \{\ \} | y \rangle (y \wedge (\{\ \}\ \boldsymbol{a}\ (\{\ \} \vee \{\ \}))) \quad , \\ \boldsymbol{A}_1\{\ \}_{1\ldots5} &\equiv \langle \{\ \} \wedge \{\ \} | y \rangle (y \vee (\{\ \}\ \boldsymbol{a}\ (\{\ \} \vee \{\ \}))) \quad , \\ \boldsymbol{A}_2\{\ \}_{1\ldots5} &\equiv \langle \{\ \} \vee \{\ \} | y \rangle (y \wedge (\{\ \}\ \boldsymbol{a}\ (\{\ \} \vee \{\ \}))) \quad , \\ \boldsymbol{A}_3\{\ \}_{1\ldots5} &\equiv \langle \{\ \} \vee \{\ \} | y \rangle (y \wedge (\{\ \}\ \boldsymbol{a}\ (\{\ \} \vee \{\ \}))) \quad , \\ \boldsymbol{A}_4\{\ \}_{1\ldots5} &\equiv \langle \{\ \} \vee \{\ \} | y \rangle (y \vee (\{\ \}\ \boldsymbol{a}\ (\{\ \} \vee \{\ \}))) \quad . \end{aligned}$$

We have that

- $\boldsymbol{A}_1\{\ \}_{1\ldots5}$ and $\boldsymbol{A}_2\{\ \}_{1\ldots5}$ are rigid and lax down-conjugates of $\boldsymbol{A}\{\ \}_{1\ldots5}$;
- $\boldsymbol{A}_3\{\ \}_{1\ldots5}$ and $\boldsymbol{A}_4\{\ \}_{1\ldots5}$ are rigid and lax down-conjugates of $\boldsymbol{A}_2\{\ \}_{1\ldots5}$;

- $\boldsymbol{A}_1\{\ \}_{1\ldots5}$, $\boldsymbol{A}_3\{\ \}_{1\ldots5}$ and $\boldsymbol{A}_4\{\ \}_{1\ldots5}$ are rigid and lax down-conjugates of $\boldsymbol{A}\{\ \}_{1\ldots5}$;
- $\boldsymbol{A}_3\{\ \}_{1\ldots5}$ and $\boldsymbol{A}_4\{\ \}_{1\ldots5}$ are lax but not rigid down-conjugates of $\boldsymbol{A}\{\ \}_{1\ldots5}$;
- $\boldsymbol{A}_3\{\ \}_{1\ldots5}$ is a rigid but not lax down-conjugate of itself;
- $\boldsymbol{A}_2\{\ \}_{1\ldots5}$ and $\boldsymbol{A}_3\{\ \}_{1\ldots5}$ cannot be rigid or lax down-conjugates of any formula context because two corresponding connective occurrences are not saturated down.

Remark 5.1.4. Every formula and formula context is the only single rigid down-conjugate and the only single rigid up-conjugate of itself.

Remark 5.1.5. Given $A_1, \ldots, A_m$ rigid (resp., lax) down-conjugates of A, where $m \geq 2$, we can split them into two groups $A_1, \ldots, A_l$ and $A_{l+1}, \ldots, A_m$ and build, for each group, some formulae A' and A'' such that:

- $A_1, \ldots, A_l$ are rigid (resp., lax) down-conjugates of A',
- $A_{l+1}, \ldots, A_m$ are rigid (resp., lax) down-conjugates of A'' and
- A' and $A_{l+1}, \ldots, A_m$ are rigid (resp., lax) down-conjugates of A.
- A'' and $A_1, \ldots, A_l$ are rigid (resp., lax) down-conjugates of A.
- A' and A' are rigid (resp., lax) down-conjugates of A.

We build A' by taking the least saturated corresponding connective occurrences from $A_1, \ldots, A_l$; we similarly build A''. A similar remark holds for up-conjugacy.

Definition 5.1.6. We extend the notions of up- and down-saturation from connectives to formulae. Given a formula A in rDT-ODS, its *up-saturation* $\widehat{A}$ is the formula obtained by replacing each connective α in A by its up-saturation $\widehat{\alpha}$. Similarly, its *down-saturation* $\breve{A}$ is the formula obtained by replacing each connective with its down-saturation. A similar notation is used for contexts.

Example 5.1.7. In KDT for $\boldsymbol{A} \equiv (0\ \boldsymbol{a}\ x) \vee (1 \wedge y)$ we have:

$$\widehat{\boldsymbol{A}} \equiv (0\ \boldsymbol{a}\ x) \wedge (1 \wedge y) \quad ,$$
$$\breve{\boldsymbol{A}} \equiv (0\ \boldsymbol{a}\ x) \vee (1 \vee y) \quad .$$

Remark 5.1.8. For any A, A and $\breve{A}$ form a pair of (rigid or lax) down-conjugates of A, and A and $\widehat{A}$ form a pair of (rigid or lax) up-conjugates of A.

Consider, in rDT, an open formula $A\{x_i\}_{1\dots n}$ and its rigid down-conjugates A' and A''. A 'down-merge of A via A', A'' and β' consists in a derivation whose conclusion is $A'\{y_i\}_{1\dots n}\,\beta\,A''\{z_i\}_{1\dots n}$, after a renaming of variables, and whose premise is A where each variable x_i is substituted by $y_i\ \beta\ z_i$. This construction is possible if A' and A'' have matching structures with the saturations in the right place, which is precisely what their rigid down-conjugacy captures. The following is an example with a formula $A \equiv x_1 \wedge (x_2 \vee x_3)$, some connective β and a choice of rigid down-conjugates:

$$\breve{\wedge}\beta\,\frac{(y_1\ \beta\ z_1) \wedge \boxed{\beta\breve{\vee}\,\dfrac{(y_2\ \beta\ z_2) \vee (y_3\ \beta\ z_3)}{(y_2 \vee y_3)\ \beta\ (z_2 \vee z_3)}}}{(y_1 \vee (y_2 \vee y_3))\ \beta\ (z_1 \wedge (z_2 \vee z_3))}\quad .$$

The down-merge construction is a linear derivation whose variables can be further instantiated, in particular with units. A more concrete case is the usual deep-inference construction that allows us to reduce a generic identity axiom to its atomic form, for example:

$$\breve{\wedge}\vee\,\frac{(1 \vee 0) \wedge \boxed{\vee\breve{\wedge}\,\dfrac{(0 \vee 1) \wedge (1 \vee 0)}{(0 \wedge 1) \vee (1 \vee 0)}}}{(1 \vee (0 \wedge 1)) \vee (0 \wedge (1 \vee 0))}\quad .$$

Note that the two formulae disjuncted in the conclusion are one the negation of the other. Under kdt, the above derivation collapses to the unit $\mathsf{1}$. This illustrates the main idea in this thesis: to obtain strictly linear derivations we create structures that mimic other given structures (in the case above, A) and that are transparent to an abstraction.

Notation 5.1.9. We denote by $A(x)$ a formula A where any number of occurrences of x (including zero) appear free. Note that, for example, the notation $A\{x\}(y)$ could denote the formula $y \wedge (x \vee y)$, or the formula x, but not the formula y. We use this notation in order to prove the inductive cases for $\langle A|x\rangle\, B$ for the following lemmas without assuming that x occurs free in B and without applying the explicit substitution.

The Merge Lemma can be stated more conveniently as two variants, according to where the saturations are observed.

Lemma 5.1.10 (Merge Lemma)**.** *Let $A\{\ \}_{1\dots n}$ be a skeleton context in* rDT-$\mathcal{O}$*, where $\mathcal{O} \in \{\mathsf{OD}, \mathsf{ODS}\}$; let $\mathcal{C}_A$ be the set of connectives appearing in $A\{\ \}_{1\dots n}$; let β be a connective of* rDT*. If $A'\{\ \}_{1\dots n}$ and $A''\{\ \}_{1\dots n}$ are rigid down-conjugates (resp., rigid up-conjugates) of $A\{\ \}_{1\dots n}$ then there exists a derivation*

$$\begin{array}{c} A\{y_i\ \beta\ z_i\}_{1\dots n} \\ \Big\| \{\beta\breve{\alpha}, \breve{\alpha}\beta \mid \alpha \in \mathcal{C}_A\}\text{-}\mathcal{O} \\ A'\{y_i\}_{1\dots n}\ \beta\ A''\{z_i\}_{1\dots n} \end{array} \qquad \textit{(resp.,} \qquad \begin{array}{c} A'\{y_i\}_{1\dots n}\ \beta\ A''\{z_i\}_{1\dots n} \\ \Big\| \{\beta\hat{\alpha}, \hat{\alpha}\beta \mid \alpha \in \mathcal{C}_A\}\text{-}\mathcal{O} \\ A\{y_i\ \beta\ z_i\}_{1\dots n} \end{array} \textit{)}.$$

Both the width and the height of the derivation are at most $2|A\{x_i\}_{1\dots n}|$.

Proof. We build the derivation on the left by induction on the structure of $A\{\ \}_{1\dots n}$.

For the base case $A \equiv \{\ \}$ we take the derivation $y_1\ \beta\ z_1$.

There are two inductive cases:

1. If
$$\begin{aligned} A\{\ \}_{1\dots n} &\equiv\ E\{\ \}_{1\dots l}\ \alpha\ \ F\{\ \}_{l+1\dots n}\ , \\ A'\{\ \}_{1\dots n} &\equiv\ E'\{\ \}_{1\dots l}\ \alpha'\ F'\{\ \}_{l+1\dots n}\ , \\ A''\{\ \}_{1\dots n} &\equiv E''\{\ \}_{1\dots l}\ \alpha''\ F''\{\ \}_{l+1\dots n}\ , \end{aligned}$$
we build
$$r\,\frac{\boxed{\begin{array}{c} E\{y_i\ \beta\ z_i\}_{1\dots l} \\ \| \\ E'\{y_i\}_{1\dots l}\ \beta\ E''\{z_i\}_{1\dots l} \end{array}}\ \alpha\ \boxed{\begin{array}{c} F\{y_i\ \beta\ z_i\}_{l+1\dots n} \\ \| \\ F'\{y_i\}_{l+1\dots n}\ \beta\ F''\{z_i\}_{l+1\dots n} \end{array}}}{(E'\{y_i\}_{1\dots l}\ \alpha'\ F'\{y_i\}_{l+1\dots n})\ \beta\ (E''\{z_i\}_{1\dots l}\ \alpha''\ F''\{z_i\}_{l+1\dots n})}\ ,$$
where $r \in \{\beta\breve{\alpha}, \breve{\alpha}\beta\}$.

2. If
$$\begin{aligned} A\{\ \}_{1\dots n} &\equiv\ \ \langle F\{\ \}_{1\dots l} | x\rangle\ E\{\ \}_{l+1\dots n}(x)\ \ , \\ A'\{\ \}_{1\dots n} &\equiv\ \langle F'\{\ \}_{1\dots l} | x\rangle\ E'\{\ \}_{l+1\dots n}(x)\ , \\ A''\{\ \}_{1\dots n} &\equiv \langle F''\{\ \}_{1\dots l} | x\rangle\ E''\{\ \}_{l+1\dots n}(x)\ , \end{aligned}$$

we build

$$\dfrac{\left\langle \begin{array}{c} F\{y_i\ \beta\ z_i\}_{1\ldots l} \\ \| \\ F'\{y_i\}_{1\ldots l}\ \beta\ F''\{z_i\}_{1\ldots l} \end{array} \middle| x \right\rangle E\{y_i\ \beta\ z_i\}_{l+1\ldots n}(x)}{\langle F'\{y_i\}_{1\ldots l}|y, F''\{z_i\}_{1\ldots l}|z\rangle \boxed{\begin{array}{c} E\{y_i\ \beta\ z_i\}_{l+1\ldots n}(y\ \beta\ z) \\ \| \\ E'\{y_i\}_{l+1\ldots n}(y)\ \beta\ E''\{z_i\}_{l+1\ldots n}(z) \end{array}}} \quad ,$$

$$\langle F'\{y_i\}_{1\ldots l}|x\rangle\ E'\{y_i\}_{l+1\ldots n}(x)\ \beta\ \langle F''\{z_i\}_{1\ldots l}|x\rangle\ E''\{z_i\}_{l+1\ldots n}(x)$$

where y and z are fresh variables.

The width of the derivation is at most $2|A\{x_i\}_{1\ldots n}|$; note that $|A\{y_i\ \beta\ z_i\}_{1\ldots n}| < 2|A\{x_i\}_{1\ldots n}|$ if there are explicit substitutions in $A\{\ \}_{1\ldots n}$ (for example, $A\{\ \} \equiv \langle\{\ \}|x\rangle\ x$).

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 $2|A\{x_i\}_{1\ldots n}|$, the worst scenario being a formula only composed of explicit substitutions, each of which is of size 1.

The proof of the second statement is analogous. □

Remark 5.1.11. The Merge Lemma holds in rKDT-ODS for all connectives β, because for all connectives α we have that $\beta\breve{\alpha}, \breve{\alpha}\beta, \beta\widehat{\alpha}, \widehat{\alpha}\beta \in$ rKDT. In the subatomic proof system for multiplicative linear logic SAMLLS given by Aler Tubella in [2], the derivation on the left can be constructed if $\beta = \parr$ because the system contains the rules $\parr\breve{\otimes}$, $\parr\breve{\boldsymbol{a}}$, $\breve{\otimes}\parr$ and $\breve{\boldsymbol{a}}\parr$ for every atom $\boldsymbol{a}$, and the rule $\parr\breve{\parr}$ can be derived from associativity of $\parr$. Dually, the derivation on the right can be constructed if $\beta = \otimes$. This may seem limiting, but we will explain in Chapter 6 how the unit-equality inference steps could nevertheless be eliminated, albeit with an exponential complexity penalty.

The construction in Lemma 5.1.10 is deterministic, therefore we can use it for the following definition.

Definition 5.1.12. The two derivations constructed for Lemma 5.1.10 are called, respectively and left to right, the *down-merge* and the *up-merge of A via A', A'' and β*. We call the same any derivation obtained from them by applying substitutions.

Example 5.1.13. Let $\boldsymbol{A} \equiv \langle w \vee x | w \rangle (w \,\boldsymbol{a}\, (y \wedge z))$ in $\mathsf{KDT\text{-}ODS}$. Then we can build the following down-merge via the conjugate pair $\langle w \vee x | w \rangle (w \,\boldsymbol{a}\, (y \wedge z))$ and $\langle 0 \vee 0 | w \rangle (w \,\boldsymbol{a}\, (0 \vee 0))$ and the connective $\vee$:

$$\left\langle \vee\check{\vee} \dfrac{(w \vee 0) \vee (x \vee 0)}{(w \vee x) \vee (0 \vee 0)} \middle| w \right\rangle (w \,\boldsymbol{a}\, ((z \vee 0) \wedge (y \vee 0)))$$
$$\langle w \vee x | w \rangle \, \langle 0 \vee 0 | w' \rangle \quad \boldsymbol{a}\check{\vee} \dfrac{(w \vee w') \,\boldsymbol{a}\, \boxed{\vee\check{\wedge} \dfrac{(y \vee 0) \wedge (z \vee 0)}{(y \wedge z) \vee (0 \vee 0)}}}{(w \,\boldsymbol{a}\, (y \wedge z)) \vee (w' \,\boldsymbol{a}\, (0 \vee 0))}$$
$$\langle w \vee x | w \rangle \, (w \,\boldsymbol{a}\, (y \wedge z)) \vee \langle 0 \vee 0 | w \rangle \, (w \,\boldsymbol{a}\, (0 \vee 0))$$

.

Remark 5.1.14. The specification of the proof system for the merge derivations could be made tighter by tracking more closely which connectives are saturated.

The following notation is helpful when applying the Merge and Eversion Lemmas.

Notation 5.1.15. $[B_i|x_i]_{1\ldots n} A$ stands for the simultaneous actual substitution $[B_1|x_1, \ldots, B_n|x_n]\, A$. Sometimes indices will be denoted by superscripts instead of subscripts, as in $[B^i|x^i]^{1\ldots n} A$. Given a set of variables $S = \{x_1, \ldots, x_n\}$, the notation $[B_i|x_i]_S\, A$ stands for $[B_1|x_1, \ldots, B_n|x_n]\, A$; we might also write $[B_v|v]_S\, A$ to stand for $[B_{x_1}|x_1, \ldots, B_{x_n}|x_n]\, A$. These conventions extend 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 2.1.1).

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.

Remark 5.1.16. For many logics, the reduction of the identity and cut rules to atomic form is made possible by special cases of the merge construction. It is a standard deep-inference technique dating back to the origins [18]. For example, a cut

$$\dfrac{A \otimes \bar{A}}{\bot}$$

in linear logic, would be obtained as follows. Translate A into its subatomic version $\pi\boldsymbol{A}$, where $\boldsymbol{A}$ is an open formula and π an actual substitution $\underline{\boldsymbol{A}} \to \{\bot, 1\}$; from $\boldsymbol{A}$ obtain $\widetilde{\boldsymbol{A}}$ by replacing $\parr$ with $\otimes$ and $\otimes$ with $\parr$; let $\pi' \colon \underline{\boldsymbol{A}} \to \{\bot, 1\}$ be such that $\pi' x \equiv \bot$ iff $\pi x \equiv 1$. Observe that $\boldsymbol{A}$ and $\widetilde{\boldsymbol{A}}$ are rigid up-conjugates of $\breve{\boldsymbol{A}}$ and build

$$\begin{array}{c} \pi\boldsymbol{A} \otimes \pi'\widetilde{\boldsymbol{A}} \\ \| \\ [\pi v \otimes \pi' v | v]_{\underline{\boldsymbol{A}}}\breve{\boldsymbol{A}} \end{array} ,$$

which is an up-merge of $\pi\boldsymbol{A}$ and $\pi'\widetilde{\boldsymbol{A}}$ via $\otimes$. The translation of that derivation back into a standard system realises the decomposition of a generic cut rule into several instances of its atomic version (plus switches).

The following variant of the Merge Lemma is useful in the following. In its statement, we exchange the roles of A and B, and of α and β, to make Remark 5.2.4 clearer. This variant allows us to avoid forming down-conjugates (resp. up-conjugates), leaving A intact when the connective of the merge is saturated up.

Lemma 5.1.17 (Merge Lemma – Variant)**.** *Let $B\{\ \}_{1\ldots n}$ be a skeleton context in rDT-$\mathcal{O}$, where $\mathcal{O} \in \{\mathsf{OD}, \mathsf{ODS}\}$; let $\mathcal{C}_B$ be the set of connectives appearing in $B\{\ \}_{1\ldots n}$; let α be a connective of rDT. Then there exists a derivation*

$$\begin{array}{c} B\{y_i \mathbin{\widehat{\alpha}} z_i\}_{1\ldots n} \\ \| \{\widehat{\alpha}\beta, \beta\widehat{\alpha} | \beta \in \mathcal{C}_B\}\text{-}\mathcal{O} \\ B\{y_i\}_{1\ldots n} \mathbin{\widehat{\alpha}} B\{z_i\}_{1\ldots n} \end{array} \quad \textit{(resp.,} \quad \begin{array}{c} B\{y_i\}_{1\ldots n} \mathbin{\breve{\alpha}} B\{z_i\}_{1\ldots n} \\ \| \{\breve{\alpha}\beta, \beta\breve{\alpha} | \beta \in \mathcal{C}_B\}\text{-}\mathcal{O} \\ B\{y_i \mathbin{\breve{\alpha}} z_i\}_{1\ldots n} \end{array} \quad \textit{).}$$

Both the width and the height of the derivation are at most $2|B\{x_i\}_{1\ldots n}|$.

Proof. The proof is identical to the proof of Lemma 5.1.10, except for where an inference step is used. That will be either $\widehat{\beta}\alpha$, or $\alpha\widehat{\beta}$, or $\breve{\beta}\alpha$, or $\alpha\breve{\beta}$. □

Example 5.1.18. As in the previous example, let $\boldsymbol{A} \equiv \langle w \vee x | w \rangle (w \mathbin{\boldsymbol{a}} (y \wedge z))$. Then we can build the following variant down-merge in KDT-ODS via the conjugate pair $\langle w \vee x | w \rangle (w \mathbin{\boldsymbol{a}} (y \wedge z))$ and $\langle w' \vee x' | w \rangle (w \mathbin{\boldsymbol{a}} (y' \vee z'))$ and the

connective $\wedge$:

$$\dfrac{\left\langle \vee\hat{\wedge}\,\dfrac{(w \wedge w') \vee (x \wedge x')}{(w \vee x) \wedge (w' \vee x')} \middle| w \right\rangle (w \,\boldsymbol{a}\, ((z \wedge z') \wedge (y \wedge y')))}{\langle w \vee x | w \rangle \, \langle w' \vee x' | w' \rangle \, \dfrac{(w \wedge w') \,\boldsymbol{a}\, \boxed{\wedge\hat{\wedge}\,\dfrac{(y \wedge y') \wedge (z \wedge z')}{(y \wedge z) \wedge (y' \wedge z')}}}{\boldsymbol{a}\hat{\wedge}\,\dfrac{}{(w \,\boldsymbol{a}\, (y \wedge z)) \wedge (w' \,\boldsymbol{a}\, (y' \wedge z'))}}}{\langle w \vee x | w \rangle \, (w \,\boldsymbol{a}\, (y \wedge z)) \wedge \langle w' \vee x' | w \rangle \, (w \,\boldsymbol{a}\, (y' \wedge z'))}$$

Remark 5.1.19. The reduction of the contraction rule to atomic form is made possible by a special case of the variant merge construction. Again, it is a standard deep-inference technique dating back to the origins [11].

The following two propositions apply kdt to the merge constructions that are employed in the next section to eliminate unit equalities from KDTEq; they serve to show that the transformation described in Theorem 6.3.1 is transparent to the abstraction map kdt. The first, Proposition 5.1.20, relates to unit equalities of the form $\boldsymbol{A}\,\beta\,\mathsf{u}_\beta = \boldsymbol{A}$, and the second, Proposition 5.1.22, relates to atomic unit equalities of the form $u \,\boldsymbol{a}\, u = u$.

Proposition 5.1.20. *Let* $\boldsymbol{A}\{\ \}_{1\ldots n}$ *be a skeleton context in* $\mathsf{KDTEq}^{=}$-ODS*; let* $\beta \in \{\vee, \wedge\}$*; let* u_β *be the unit for* β *(*i.e.*,* $x\,\beta\,\mathsf{u}_\beta = x$*); let* $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_n$ *be any formulae such that* $\mathsf{kdt}\,\boldsymbol{D}_1 \equiv \cdots \equiv \mathsf{kdt}\,\boldsymbol{D}_n \equiv \mathsf{u}_\beta$*. Consider the following down-merge and up-merge derivations of* $\boldsymbol{A}\{\ \}_{1\ldots n}$ *via* β*:*

$$\phi \equiv \dfrac{\boldsymbol{A}\{\boldsymbol{C}_i\,\beta\,\boldsymbol{D}_i\}_{1\ldots n}}{\boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n}\,\beta\,\breve{\boldsymbol{A}}\{\boldsymbol{D}_i\}_{1\ldots n}} \quad \mathsf{KDTEq}^{=}\text{-}\mathsf{ODS} \qquad \textit{and} \qquad \psi \equiv \dfrac{\boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n}\,\beta\,\widehat{\boldsymbol{A}}\{\boldsymbol{D}_i\}_{1\ldots n}}{\boldsymbol{A}\{\boldsymbol{C}_i\,\beta\,\boldsymbol{D}_i\}_{1\ldots n}} \quad \mathsf{KDTEq}^{=}\text{-}\mathsf{ODS}\ ,$$

where $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_n$ *are any formulae. Then* $\mathsf{kdt}\,\phi = \mathsf{kdt}\,\boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n} = \mathsf{kdt}\,\psi$*. If* $\boldsymbol{A}\{\ \}_{1\ldots n}$ *is free from explicit substitutions, then* $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n} \equiv \mathsf{kdt}\,\psi$*. A similar statement holds if we exchange the role of* $\boldsymbol{C}_i$*s and* $\boldsymbol{D}_i$*s and the corresponding down and up-saturations of* $\boldsymbol{A}\{\ \}_{1\ldots n}$*.*

Proof. We follow the construction in Lemma 5.1.10 and apply kdt. If there are no explicit substitutions, each inference step r produced by the construction exhibits a premise and a conclusion that are identical to $\mathsf{kdt}\,\boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n}$.

Those r steps are therefore collapsed by kdt into synchronal compositions. In case explicit substitutions are present, then the merge construction produces two compositions by expansion for each substitution. Those compositions are turned by kdt into equality inference steps. $\square$

Example 5.1.21. The construction in Example 5.1.13 satisfies the conditions of this proposition. Applying kdt to the construction collapses the structure and results in $\langle w \vee x | w \rangle (w\ \boldsymbol{a}\ (y \wedge z))$.

Proposition 5.1.22. *Let $\boldsymbol{A} \equiv \boldsymbol{A}\{\ \}_{1\ldots n}$ be a skeleton context in $\mathsf{KDTEq}^{=}\text{-}\mathsf{ODS}$; let $\beta \in \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots\}$; let $\mathsf{u} \in \{0, 1\}$; let $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_n, \boldsymbol{D}_1, \ldots, \boldsymbol{D}_n$ be any formulae such that $\mathsf{kdt}\,\boldsymbol{C}_1 \equiv \cdots \equiv \mathsf{kdt}\,\boldsymbol{C}_n \equiv \mathsf{kdt}\,\boldsymbol{D}_1 \equiv \cdots \equiv \mathsf{kdt}\,\boldsymbol{D}_n \equiv \mathsf{u}$. Consider the following down-merge and up-merge derivations of $\boldsymbol{A}$ via β:*

$$\phi \equiv \boxed{\begin{array}{c} \boldsymbol{A}\{\boldsymbol{C}_i\ \beta\ \boldsymbol{D}_i\}_{1\ldots n} \\ \Vert\,\mathsf{KDTEq}^{=}\text{-}\mathsf{ODS} \\ \boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n}\ \beta\ \check{\boldsymbol{A}}\{\boldsymbol{D}_i\}_{1\ldots n} \end{array}} \qquad \textit{and} \qquad \psi \equiv \boxed{\begin{array}{c} \boldsymbol{A}\{\boldsymbol{C}_i\}_{1\ldots n}\ \beta\ \hat{\boldsymbol{A}}\{\boldsymbol{D}_i\}_{1\ldots n} \\ \Vert\,\mathsf{KDTEq}^{=}\text{-}\mathsf{ODS} \\ \boldsymbol{A}\{\boldsymbol{C}_i\ \beta\ \boldsymbol{D}_i\}_{1\ldots n} \end{array}}\ .$$

Then $\mathsf{kdt}\,\phi = \mathsf{kdt}\,\boldsymbol{A}\{\mathsf{u}\}_{1\ldots n} = \mathsf{kdt}\,\psi = \mathsf{u}$. If $\boldsymbol{A}$ is free from explicit substitutions, then $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\boldsymbol{A}\{\mathsf{u}\}_{1\ldots n} \equiv \mathsf{kdt}\,\psi \equiv \mathsf{u}$. A similar statement holds if we exchange $\boldsymbol{A}$ and $\check{\boldsymbol{A}}$ in the conclusion of ϕ or $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ in the premise of ψ.

Proof. The proof is similar to the one for Proposition 5.1.20. Note that, in this case, all variables in $\boldsymbol{A}$ are substituted by formulae equivalent to u, therefore $\mathsf{kdt}\,\phi$ and $\mathsf{kdt}\,\psi$ are equivalent or identical to u. $\square$

Example 5.1.23. We consider the following down-merge in $\mathsf{KDT}\text{-}\mathsf{ODS}$ via the conjugate pair $\langle \mathsf{u} \wedge \mathsf{u} | w \rangle (w\ \boldsymbol{b}\ (\mathsf{u} \vee \mathsf{u}))$ and $\langle \mathsf{u} \vee \mathsf{u} | w \rangle (w\ \boldsymbol{b}\ (\mathsf{u} \wedge \mathsf{u}))$ and the connective $\boldsymbol{a}$:

$$\boxed{\begin{array}{c} \left\langle \boldsymbol{a}\check{\wedge}\, \dfrac{(\mathsf{u}\ \boldsymbol{a}\ \mathsf{u}) \wedge (\mathsf{u}\ \boldsymbol{a}\ \mathsf{u})}{(\mathsf{u} \wedge \mathsf{u})\ \boldsymbol{a}\ (\mathsf{u} \vee \mathsf{u})} \middle| w \right\rangle (w\ \boldsymbol{b}\ ((\mathsf{u}\ \boldsymbol{a}\ \mathsf{u}) \wedge (\mathsf{u}\ \boldsymbol{a}\ \mathsf{u}))) \\ \sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim \\ \langle \mathsf{u} \wedge \mathsf{u} | w_1 \rangle\ \langle \mathsf{u} \vee \mathsf{u} | w_2 \rangle \boxed{\boldsymbol{a}\check{\boldsymbol{b}}\, \dfrac{(w_1\ \boldsymbol{a}\ w_2)\ \boldsymbol{b}\ \boxed{\check{\wedge}\boldsymbol{a}\, \dfrac{(\mathsf{u}\ \boldsymbol{a}\ \mathsf{u}) \wedge (\mathsf{u}\ \boldsymbol{a}\ \mathsf{u})}{(\mathsf{u} \vee \mathsf{u})\ \boldsymbol{a}\ (\mathsf{u} \wedge \mathsf{u})}}}{(w_1\ \boldsymbol{b}\ (\mathsf{u} \vee \mathsf{u}))\ \boldsymbol{a}\ (w_2\ \boldsymbol{b}\ (\mathsf{u} \wedge \mathsf{u}))}} \\ \sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim\sim \\ \langle \mathsf{u} \wedge \mathsf{u} | w \rangle\ (w\ \boldsymbol{b}\ (\mathsf{u} \vee \mathsf{u})) \wedge \langle \mathsf{u} \vee \mathsf{u} | w \rangle\ (w\ \boldsymbol{b}\ (\mathsf{u} \wedge \mathsf{u})) \end{array}}\ .$$

This derivation satisfies the conditions of this proposition. Applying kdt to this construction results in u as the entire derivation is collapsed.

5.2 Eversion

We can now prove the Eversion Lemma and make precise the conditions mentioned in the introduction to this chapter. The main point to understand is that, for the construction to work, it is necessary to have one and only one un-saturated occurrence of each connective occurrence in A among its conjugates substituted into B. This allows us to trigger the Merge Lemma and incrementally insert A into B.

Lemma 5.2.1 (Eversion Lemma)**.** *In* rDT-ODS*, let*

- $A_1\{\ \}_{1\dots n}, \dots, A_m\{\ \}_{1\dots n}$ *be skeleton contexts that are rigid down-conjugates of* $A\{\ \}_{1\dots n}$ *and*
- $A^1\{\ \}_{1\dots n}, \dots, A^m\{\ \}_{1\dots n}$ *be skeleton contexts that are rigid up-conjugates of* $A\{\ \}_{1\dots n}$*;*

let $B \equiv B\{y^j\}^{1\dots m}$ *be an open formula such that* $\underline{B} = \{y^1, \dots, y^m\}$*; assume that, for* $1 \leq j \leq m$*,* y^j *appears at least once in* $\mathsf{fl}\, B$ *and if it appears more than once, then*

- $A_j\{\ \}_{1\dots n} \equiv \breve{A}\{\ \}_{1\dots n}$ *and*
- $A^j\{\ \}_{1\dots n} \equiv \widehat{A}\{\ \}_{1\dots n}$*;*

let $\mathcal{C}_A$ *and* $\mathcal{C}_B$ *be the sets of connectives appearing in* A *and* B*, respectively. Then the following derivations exist:*

$$\begin{array}{c} A\left\{B\{x_i^j\}^{1\dots m}\right\}_{1\dots n} \\ \Big\| \{\beta\breve{\alpha}, \breve{\alpha}\beta \mid \alpha\in\mathcal{C}_A, \beta\in\mathcal{C}_B\} \\ B\left\{A_j\{x_i^j\}_{1\dots n}\right\}^{1\dots m} \end{array} \quad \textit{and} \quad \begin{array}{c} B\left\{A^j\{x_i^j\}_{1\dots n}\right\}^{1\dots m} \\ \Big\| \{\beta\widehat{\alpha}, \widehat{\alpha}\beta \mid \alpha\in\mathcal{C}_A, \beta\in\mathcal{C}_B\} \\ A\left\{B\{x_i^j\}^{1\dots m}\right\}_{1\dots n} \end{array},$$

where $1 \leq i \leq n$ *and* $1 \leq j \leq m$*, and explicit substitutions appear if and only if they appear in* A *or* B*. Both the width and the height of the derivations are* $O(|A||B|)$*.*

Proof. We build the derivation on the left in the statement by induction on the structure of B.

For the base case $B \equiv y^1$, we take the derivation $A\{x_i^1\}_{1\dots n}$.

There are two inductive cases:

1. If $B\{y^j\}^{1\ldots m} \equiv C\{y^j\}^{1\ldots l}\ \beta\ D\{y^j\}^{l+1\ldots m}$ we build the following derivation, where ψ and χ are obtained by induction and ϕ by Lemma 5.1.10.

$$\dfrac{\dfrac{A\left\{C\{x_i^j\}^{1\ldots l}\ \beta\ D\{x_i^j\}^{l+1\ldots m}\right\}_{1\ldots n}}{\phi \Big\| \{\,\beta\check{\alpha},\check{\alpha}\beta|\alpha\in\mathcal{C}_A\,\}} \qquad \dfrac{\dfrac{A'\left\{C\{x_i^j\}^{1\ldots l}\right\}_{1\ldots n}}{\psi \|}}{C\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots l}} \ \beta\ \dfrac{\dfrac{A''\left\{D\{x_i^j\}^{l+1\ldots m}\right\}_{1\ldots n}}{\chi \|}}{D\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{l+1\ldots m}}}$$

   $A'\{\ \}_{1\ldots n}$ and $A''\{\ \}_{1\ldots n}$ are obtained as in Remark 5.1.5 and are rigid down-conjugates of $A\{\ \}_{1\ldots n}$.

2. If $B\{y^j\}^{1\ldots m} \equiv \left\langle C\{y^j\}^{1\ldots l}\middle|y\right\rangle D\{y^j\}^{l+1\ldots m}(y)$ we build the following derivation, where ψ and χ are obtained by induction.

$$\dfrac{A\left\{\left\langle C\{x_i^j\}^{1\ldots l}\middle|y\right\rangle D\{x_i^j\}^{l+1\ldots m}(y)\right\}_{1\ldots n}}{\left\langle C\{x_i^j\}^{1\ldots l}\middle|x_i\right\rangle_{1\ldots n} \quad \dfrac{A\left\{D\{x_i^j\}^{l+1\ldots m}(x_i)\right\}_{1\ldots n}}{\chi \|}\ D\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{l+1\ldots m}(A'\{x_i\}_{1\ldots n})}$$
$$\left\langle \dfrac{\dfrac{A'\left\{C\{x_i^j\}^{1\ldots l}\right\}_{1\ldots n}}{\psi \|}}{C\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots l}} \middle| y \right\rangle D\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{l+1\ldots m}(y)$$

   $A'\{\ \}_{1\ldots n}$ is obtained as in Remark 5.1.5. There might be multiple copies of $A'\{\ \}_{1\ldots n}$ in the conclusion of χ, but the conditions in the statement of the lemma guarantee that if there is more than one copy, then they are saturated down, therefore do not break the induction hypothesis. It is also important that $A'\{\ \}_{1\ldots n}$ appears at least once in the conclusion of χ, again to validate the induction hypothesis.

The construction of the derivation on the right in the statement is analogous.

The width of the derivations is $O(|A||B|)$. The worst-case scenario for the height is Case 1, where the height increases by $2|A|$ at most, in the merge ϕ. Since the number of iterations is dominated by $|B|$, we have that the height is $O(|A||B|)$. □

Definition 5.2.2. We call *eversions* those derivations constructed in Lemma 5.2.1.

Example 5.2.3. Let $\boldsymbol{A} \equiv (\{\ \} \wedge \{\ \})\, \boldsymbol{a}\, \{\ \}$ and $\boldsymbol{B} \equiv \{\ \}\, \boldsymbol{b}\, (\{\ \} \vee \{\ \})$. Then we can construct the eversion:

$$\dfrac{\dfrac{(x_1 \,\boldsymbol{b}\, (y_1 \vee z_1)) \wedge (x_2 \,\boldsymbol{b}\, (y_2 \vee y_3))}{\check{\wedge}\boldsymbol{b}}\ \boldsymbol{a}\ (x_3 \,\boldsymbol{b}\, (y_3 \vee z_3))}{\boldsymbol{b}\check{\boldsymbol{a}}}$$

$$\dfrac{(x_1 \vee x_2) \,\boldsymbol{b}\, \dfrac{(y_1 \vee z_1) \wedge (y_2 \vee y_3)}{\check{\vee}\vee}}{((x_1 \vee x_2) \,\boldsymbol{a}\, x_3) \,\boldsymbol{b}\, \dfrac{((y_1 \vee y_2) \vee (z_1 \wedge z_2)) \,\boldsymbol{a}\, (y_3 \vee z_3)}{\vee\check{\boldsymbol{a}}}}$$

$$\dfrac{\dfrac{(x_1 \,\boldsymbol{b}\, (y_1 \vee z_1)) \wedge (x_2 \,\boldsymbol{b}\, (y_2 \vee y_3))}{\check{\wedge}\boldsymbol{b}}\ \boldsymbol{a}\ (x_3 \,\boldsymbol{b}\, (y_3 \vee z_3))}{\boldsymbol{b}\check{\boldsymbol{a}}}$$

Remark 5.2.4. The Merge Lemma 5.1.10 is a special case of the Eversion Lemma where $B\{\ \}\{\ \} \equiv \{\ \}\, \beta\, \{\ \}$; its variant Lemma 5.1.17 is a special case of the Eversion Lemma where the down-merge is obtained by setting $A\{\ \}\{\ \} \equiv \{\ \}\, \widehat{\alpha}\, \{\ \}$ and the up-merge by setting $A\{\ \}\{\ \} \equiv \{\ \}\, \check{\alpha}\, \{\ \}$.

The following corollary of the Eversion Lemma is useful because it is a simpler, special case of the lemma when $A\{\ \}_{1\ldots n} \equiv \check{A}\{\ \}_{1\ldots n}$, which is a common occurrence when investigating the properties of strictly linear derivations.

Corollary 5.2.5. *Let $A\{\ \}_{1\ldots n}$ and $B\{\ \}^{1\ldots m}$ be skeleton contexts and let $\mathcal{C}_A$ and $\mathcal{C}_B$ be the respective sets of connectives. Then there exist the derivations*

$$\begin{array}{l} \check{A}\left\{B\{z_i^j\}^{1\ldots m}\right\}_{1\ldots n} \\ \qquad \Vert_{\{\beta\check{\alpha},\check{\alpha}\beta \mid \alpha\in\mathcal{C}_A,\beta\in\mathcal{C}_B\}} \\ B\left\{\check{A}\{z_i^j\}_{1\ldots n}\right\}^{1\ldots m} \end{array} \qquad \textit{and} \qquad \begin{array}{l} B\left\{\widehat{A}\{z_i^j\}_{1\ldots n}\right\}^{1\ldots m} \\ \qquad \Vert_{\{\beta\widehat{\alpha},\widehat{\alpha}\beta \mid \alpha\in\mathcal{C}_A,\beta\in\mathcal{C}_B\}} \\ \widehat{A}\left\{B\{z_i^j\}^{1\ldots m}\right\}_{1\ldots n} \end{array},$$

where explicit substitutions appear if and only if they appear in A or B.

Remark 5.2.6. Because of the above corollary, every instance of a rule of rDT has the same premise and conclusion as some eversion.

5.3 System DT^*

In this section, we illustrate briefly how eversion can be useful in the design of proof systems. We leave the full exploration of these ideas to other papers but we think that the few considerations here help to appreciate the results in this thesis.

We have defined two variants of conjugacy, a rigid one and a lax one, and established the Eversion Lemma for the rigid variant, which corresponds to System rDT. We can prove a different and somewhat more general version of the lemma with the lax variant of conjugacy if we extend rDT to DT; that is, if we add mix rules to the system. That would allow for a connective to be weakened along the order induced by saturation. For classical logic, the new rules do not affect provability, but the availability of new derivations simplifies certain constructions. For example, the cut-elimination construction in Chapter 8 benefits from it.

We can state for DT the following two 'lax' versions of the Eversion Lemma 5.2.1 and its corollary 5.2.5, and a similar remark to 5.2.6.

Lemma 5.3.1 (Eversion Lemma – Lax Variant)**.** *In* DT-ODS*, let*

- $A_1\{\ \}_{1\ldots n}, \ldots, A_m\{\ \}_{1\ldots n}$ *be skeleton contexts that are lax down-conjugates of* $A\{\ \}_{1\ldots n}$ *and*
- $A^1\{\ \}_{1\ldots n}, \ldots, A^m\{\ \}_{1\ldots n}$ *be skeleton contexts that are lax up-conjugates of* $A\{\ \}_{1\ldots n}$*;*

let $B \equiv B\{y^j\}^{1\ldots m}$ *be an open formula such that* $\underline{B} = \{y^1, \ldots, y^m\}$*; assume that, for* $1 \leq j \leq m$*,* y^j *appears at least once in* $\mathsf{fl}\, B$ *and if it appears more than once, then*

- $A_j\{\ \}_{1\ldots n} \equiv \breve{A}\{\ \}_{1\ldots n}$ *and*
- $A^j\{\ \}_{1\ldots n} \equiv \widehat{A}\{\ \}_{1\ldots n}$*;*

let $\mathcal{C}_A$ *and* $\mathcal{C}_B$ *be the sets of connectives appearing in* A *and* B*, respectively. Then the following derivations exist:*

$$\begin{array}{c} A\left\{B\{x_i^j\}^{1\ldots m}\right\}_{1\ldots n} \\ \Big\| \{\breve{\alpha}, \beta\breve{\alpha}, \breve{\alpha}\beta \mid \alpha \in \mathcal{C}_A, \beta \in \mathcal{C}_B\} \\ B\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots m} \end{array} \qquad \textit{and} \qquad \begin{array}{c} B\left\{A^j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots m} \\ \Big\| \{\hat{\alpha}, \beta\hat{\alpha}, \hat{\alpha}\beta \mid \alpha \in \mathcal{C}_A, \beta \in \mathcal{C}_B\} \\ A\left\{B\{x_i^j\}^{1\ldots m}\right\}_{1\ldots n} \end{array},$$

where $1 \leq i \leq n$ *and* $1 \leq j \leq m$*, and explicit substitutions appear if and only if they appear in* A *or* B*. Both the width and the height of the derivations are* $O(|A||B|)$*.*

Proof. We can build the derivation on the left of the statement this way:

$$\begin{array}{l} A\left\{B\{x_i^j\}^{1\ldots m}\right\}_{1\ldots n} \\ \qquad\qquad \Big\| \{\breve{\alpha}, | \alpha \in \mathcal{C}_A\} \\ A'\left\{B\{x_i^j\}^{1\ldots m}\right\}_{1\ldots n} \\ \qquad\qquad \phi \Big\| \{\beta\breve{\alpha}, \breve{\alpha}\beta | \alpha \in \mathcal{C}_A, \beta \in \mathcal{C}_B\} \\ B\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots m} \end{array} ,$$

where $A'\{\ \}_{1\ldots n}$ is such that $A_1\{\ \}_{1\ldots n}$, $\ldots$, $A_m\{\ \}_{1\ldots n}$ are its rigid down-conjugates and ϕ is obtained by Lemma 5.2.1. We can build analogously the derivation on the right of the statement. □

This variant of the Eversion Lemma holds in the system KDT-ODS, where we have the mix rule $\breve{\wedge}$.

Corollary 5.3.2. *Let $A\{\ \}_{1\ldots n}$ and $B\{\ \}^{1\ldots m}$ be skeleton contexts and let $\mathcal{C}_A$ and $\mathcal{C}_B$ be the respective sets of connectives. Then there exist the derivations*

$$\begin{array}{l} A\left\{B\{z_i^j\}^{1\ldots m}\right\}_{1\ldots n} \\ \qquad\qquad \Big\| \{\breve{\alpha}, \beta\breve{\alpha}, \breve{\alpha}\beta | \alpha \in \mathcal{C}_A, \beta \in \mathcal{C}_B\} \\ B\left\{\breve{A}\{z_i^j\}_{1\ldots n}\right\}^{1\ldots m} \end{array} \quad \textit{and} \quad \begin{array}{l} B\left\{\widehat{A}\{z_i^j\}_{1\ldots n}\right\}^{1\ldots m} \\ \qquad\qquad \Big\| \{\hat{\alpha}, \beta\hat{\alpha}, \hat{\alpha}\beta | \alpha \in \mathcal{C}_A, \beta \in \mathcal{C}_B\} \\ A\left\{B\{z_i^j\}^{1\ldots m}\right\}_{1\ldots n} \end{array} ,$$

where explicit substitutions appear if and only if they appear in A or B.

Remark 5.3.3. Because of the above corollary, every instance of a rule of DT has the same premise and conclusion as some eversion.

It is possible to check in polynomial time whether two given formulae are the premise and conclusion of an eversion either in rDT or DT, therefore we can use eversion as an inference rule.

Definition 5.3.4. The following two proof systems are defined in Figure 5.1, where $A_1\{\ \}_{1\ldots n}$, $\ldots$, $A_m\{\ \}_{1\ldots n}$, $A^1\{\ \}_{1\ldots n}$, $\ldots$, $A^m\{\ \}_{1\ldots n}$ and $B\{\ \}_{1\ldots m}$ stand for skeleton contexts:

- System rDT*, if the $A_i\{\ \}_{1\ldots n}$s and the $A^i\{\ \}_{1\ldots n}$s are rigid conjugates of $A\{\ \}_{1\ldots n}$, and
- System DT*, if the $A_i\{\ \}_{1\ldots n}$s and the $A^i\{\ \}_{1\ldots n}$s are lax conjugates of $A\{\ \}_{1\ldots n}$.

$$\star\{\bar{\star}\}\frac{B\left\{A^j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots m}}{A\left\{B\{x_i^j\}^{1\ldots m}\right\}_{1\ldots n}} \qquad \text{where} \qquad \begin{array}{l} A^1\{\ \}_{1\ldots n},\ \ldots,\ A^m\{\ \}_{1\ldots n} \text{ are} \\ \text{up-conjugates of } A\{\ \}_{1\ldots n} \\ \\ A_1\{\ \}_{1\ldots n},\ \ldots,\ A_m\{\ \}_{1\ldots n} \text{ are} \\ \text{down-conjugates of } A\{\ \}_{1\ldots n} \end{array}$$

$$\star\{\bar{\star}\}\frac{A\left\{B\{x_i^j\}^{1\ldots m}\right\}_{1\ldots n}}{B\left\{A_j\{x_i^j\}_{1\ldots n}\right\}^{1\ldots m}}$$

Figure 5.1: System rDT^*, for rigid conjugates, and System DT^*, for lax conjugates.

Proposition 5.3.5. *Every derivation in* rDT^* *can be turned in polynomial time into a derivation in* rDT *with the same premise and conclusion.*

Proposition 5.3.6. *Every derivation in* DT^* *can be turned in polynomial time into a derivation in* DT *with the same premise and conclusion.*

Systems rDT^* and DT^* are remarkable because they capture a vast number of inference rules for many logics. One way they do that is via rDT for binary connectives, but we can extend saturation and conjugacy to unary connectives, *i.e.*, modalities and quantifiers, and the Eversion Lemma is still valid. Moreover, the inference rules so generated are sound for many logics. For example, in modal logics [32] we get

$$\vee\check{\Box}\frac{\Box(x \vee y)}{\Box x \vee \Diamond y} \qquad \text{and} \qquad \wedge\hat{\Diamond}\frac{\Diamond x \wedge \Box y}{\Diamond(x \wedge y)} \quad ;$$

in linear logic [33] we get

$$\parr\check{!}\frac{!(x \parr y)}{!x \parr ?y} \qquad \text{and} \qquad \otimes\hat{?}\frac{?x \otimes !y}{?(x \otimes y)} \quad ;$$

in classical logic [9] we get

$$\vee\check{\forall}\frac{\forall x.(A \vee B)}{\forall x.A \vee \exists x.B} \qquad \text{and} \qquad \wedge\hat{\exists}\frac{\exists x.A \wedge \forall x.B}{\exists x.(A \wedge B)} \quad .$$

One important difference between rDT^* and DT^* is the treatment of unary sets of conjugates. For example, the formula $\forall x.x$ generates one set containing one rigid down-conjugate, *i.e.*, $\forall x.x$, and one set containing one lax

down-conjugate, *i.e.*, $\exists x.x$. Only the lax conjugate generates sound rules in classical logic, for example

$$\exists\check{\forall}\,\frac{\forall x.\exists y.A}{\exists y.\exists x.A} \qquad \text{and} \qquad \forall\hat{\exists}\,\frac{\forall y.\forall x.A}{\exists x.\forall y.A} \quad .$$

The rigid notion of conjugacy would require each connective to be saturated, generating instead

$$\exists\check{\forall}\,\frac{\forall x.\exists y.A}{\exists y.\forall x.A} \qquad \text{and} \qquad \forall\hat{\exists}\,\frac{\forall y.\exists x.A}{\exists x.\forall y.A} \quad ,$$

which are not sound.

We want to investigate further what role eversion can play in the design of proof systems and the study of their properties, in particular normalisation. We can start from DT^*, define saturation for the language at hand, and then determine which of the rules generated are sound for the intended semantics. We hope that this method will give us a systematic and simplified normalisation theory for several logics.

Chapter 6

Strict Linearity

In the introduction of Chapter 5, we illustrated the problem that we want to solve: achieving strict linearity by eliminating unit-equality inference steps. With eversion and explicit substitutions, this can be achieved with only a polynomial complexity cost in the size of the derivation; both play a vital role here.

The basic idea is that units in a derivation can be padded with structure which mimics their surroundings and making them amenable to being merged. This padding consists only of multiple copies of the unit, and so is invisible to the kdt abstraction map. Moreover, the only subderivations that they are involved in are instances of merges and eversions which do not perform any semantically meaningful logical deduction, and therefore can be discounted in normalisation procedures.

A naive approach to eliminating the unit-equality inference steps in this way will blow up the size of the derivation exponentially, because the padded formulae will be propagated through the derivation and will interfere with one another. In this chapter we show that this can be controlled, so that the unit-equality steps can be eliminated while maintaining structural equivalence and paying only a polynomial price in terms of complexity.

We work inside $\mathsf{KDTEq}^{=}\text{-}\mathsf{ODS}$ and we turn $\mathsf{rKDTEq}\text{-}\mathsf{OD}$ (resp., $\mathsf{KDTEq}\text{-}\mathsf{OD}$) derivations into $\mathsf{rKDT}\text{-}\mathsf{ODS}$ (resp., $\mathsf{KDT}\text{-}\mathsf{ODS}$) ones. In other words, we exchange unit-equality steps for explicit substitutions. Having explicit substitutions does not affect the normalisation properties of any proof system because normalisation can happen inside substitutions the same way as it happens outside.

First we will discuss the specifics of how interference between elimination

steps can blow up the size of the derivation, and show how eversion and explicit substitutions are deployed to control them. We then prove the main result of this chapter, using a construction on the entire given derivation.

We refer back to the unit equalities given in Figure 3.6 and note that they all can be expressed as

$$\pi\left[\dfrac{\boldsymbol{A}}{\boldsymbol{A}\ \alpha\ x}=\right] \qquad \text{or} \qquad \pi\left[\dfrac{\boldsymbol{A}\ \alpha\ x}{\boldsymbol{A}}=\right]$$

for x a variable, $\boldsymbol{A}$ an open formula, and π an actual substitution which puts a unit onto x; without loss of generality, we assume that the unit is always on the right. For the 'bifurcating' unit equalities $\mathsf{u} = \mathsf{u}\,\boldsymbol{a}\,\mathsf{u} = \mathsf{u} \wedge \mathsf{u} = \mathsf{u} \vee \mathsf{u}$, we note that this means making a choice for which occurrence of u we take to be $\boldsymbol{A}$. We can describe those unit-equality inference steps as shown on the left as propagating a unit upwards through the proof; and those as shown on the right as propagating a unit downwards.

This general template for the unit-equality inference steps will allow us to perform the transformations necessary and to treat the unit-equality inference steps in a general way.

Given a derivation $\phi \in \mathsf{KDTEq\text{-}OD}$, we eliminate the unit-equality inference steps in two phases, where the first deals with all those propagating a unit downwards and the second with all those propagating one upwards. In the first phase, the downward-propagating unit-equality inference steps are replaced by an appropriate merge; and in the second phase, the upward-propagating unit-equality inference steps are replaced by an appropriate merge and where necessary, these two merges are resolved with an eversion.

6.1 Compression via Eversion

Consider the problem of eliminating the unit equations from a derivation $\phi \in \mathsf{rKDTEq\text{-}OD}$ without using the eversion lemma, and in particular consider eliminating a pair of unit-equality inference steps as shown here (where we assume that in each section of ψ, x occurs exactly once, so that the two inference steps are connected):

$$\pi\left[\begin{array}{c}\phi\\ \hline \boldsymbol{K}\left\{=\dfrac{\boldsymbol{A}}{\boldsymbol{A}\ \alpha\ x}\right\}\\ \hline \psi\\ \hline \mathbf{H}\left\{=\dfrac{\boldsymbol{B}\ \beta\ x}{\boldsymbol{B}}\right\}\\ \hline \chi\end{array}\right]$$

Eliminating the two unit equalities would result in a substitution $\left[\widehat{\boldsymbol{B}}^{x}\middle|x\right]$ being propagated up, and a substitution $\left[\breve{\boldsymbol{A}}^{x}\middle|x\right]$ being propagated down (where $\boldsymbol{A}^{x}$ and $\boldsymbol{B}^{x}$ stand for the result of substituting the variable x onto every leaf of $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively). These cannot in general be resolved without using eversion. Therefore, in order to eliminate both unit equalities, the entire context around one of them must be duplicated, resulting in something like the following derivation:

$$\pi\left[\begin{array}{c}[v\ \beta\ x|v]_{\underline{\mathbf{H}\{\ \}}}[w\ \alpha\ x|w]_{\underline{\boldsymbol{A}}}\phi\\ \hline [v\ \beta\ x|v]_{\underline{\mathbf{H}\{\ \}}}\boldsymbol{K}\left\{\begin{array}{c}[w\ \alpha\ x|w]_{\underline{\boldsymbol{A}}}\boldsymbol{A}\\ \|\\ \boldsymbol{A}\ \alpha\ \breve{\boldsymbol{A}}^{x}\end{array}\right\}\\ \hline [v\ \beta\ x|v]_{\underline{\mathbf{H}\{\ \}}}\left[\breve{\boldsymbol{A}}^{x}\middle|x\right]\ \psi\\ \hline [v\ \beta\ x|v]_{\underline{\mathbf{H}\{\ \}}}\mathbf{H}\left\{\boldsymbol{B}\ \beta\ \breve{\boldsymbol{A}}^{x}\right\}\\ \|\\ \left[\begin{array}{c}\mathbf{H}\{\boldsymbol{B}\}\\ \hline \chi\end{array}\right]\ \beta\ [x|v]_{\underline{\breve{\mathbf{H}}\{\ \}}}\breve{\mathbf{H}}\{\breve{\boldsymbol{A}}^{x}\}\end{array}\right]\quad.$$

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.

Indeed this is the situation that we would find in a subatomic system for multiplicative linear logic with units and decision trees. It is not the case that $\bot \otimes \bot = \bot$ nor that $1 \parr 1 = 1$, and so unit-equality inference steps

$= \dfrac{\bot \otimes \bot}{\bot}$ and $= \dfrac{1}{1 \parr 1}$ cannot in general be eliminated via local merges. We would instead have to duplicate the context in this way.

In classical logic, by instead using the eversion lemma to resolve the propagated substitutions, the duplication of the context $\mathbf{H}\{\ \}$ is avoided.

6.2 Compression via Explicit Substitutions

Eversion alone is not sufficient to control the potential blow-up of size; we need to take care to describe a construction in which successive eliminations of unit equalities cannot lead to an exponential accumulation of material.

To observe the potential blow-up, consider the following derivation:

$$\pi \left[\begin{array}{c} \kappa_1 \left\{ = \dfrac{\boldsymbol{A}}{\boldsymbol{A}\ \alpha\ x} \right\} \\ \hdotsfor{1} \\ \kappa_2 \left\{ = \dfrac{\boldsymbol{B}\{x\}}{\boldsymbol{B}\{x\}\ \beta\ y} \right\} \\ \hdotsfor{1} \\ \kappa_3 \left\{ = \dfrac{\boldsymbol{C}\{x\}\{y\}}{\boldsymbol{C}\{x\}\{y\}\ \gamma\ z} \right\} \end{array}\right] ,$$

in which $\kappa_1, \kappa_2, \kappa_3$ are derivation contexts.

Eliminating these unit equalities from top to bottom will result in the following derivation (where again, $\boldsymbol{A}^x$ stands for substituting a variable x onto every leaf of a formula $\boldsymbol{A}$):

$$\pi \left[\begin{array}{c} [v\ \gamma\ z|v]_{\underline{\boldsymbol{C}\{\ \}\{\ \}}} [v\ \beta\ (y\ \gamma\ z)|v]_{\underline{\boldsymbol{B}\{\ \}}} \kappa_1 \left\{ \begin{array}{c} [v\ \alpha\ ((x\ \gamma\ z)\ \beta\ (y\ \gamma\ z))|v]_{\underline{\boldsymbol{A}}} \boldsymbol{A} \\ \| \\ \boldsymbol{A}\ \alpha\ [(x\ \gamma\ z)\ \beta\ (y\ \gamma\ z)|v]_{\underline{\breve{\boldsymbol{A}}}} \breve{\boldsymbol{A}} \end{array} \right\} \\ \hdotsfor{1} \\ [v\ \gamma\ z|v]_{\underline{\boldsymbol{C}\{\ \}\{\ \}}} \kappa_2 \left\{ \begin{array}{c} [v\ \beta\ (y\ \gamma\ z)|v]_{\underline{\boldsymbol{B}\{\ \}}} \boldsymbol{B}\{[(x\ \gamma\ z)\ \beta\ (y\ \gamma\ z)|v]_{\underline{\breve{\boldsymbol{A}}}} \breve{\boldsymbol{A}}\} \\ \| \\ \boldsymbol{B}\{[x\ \gamma\ z|v]_{\underline{\breve{\boldsymbol{A}}}} \breve{\boldsymbol{A}}\}\ \beta\ [y\ \gamma\ z|v]_{\underline{\breve{\boldsymbol{B}}}} \breve{\boldsymbol{B}}\{[y\ \gamma\ z|v]_{\underline{\breve{\boldsymbol{A}}}} \breve{\boldsymbol{A}}\} \end{array} \right\} \\ \hdotsfor{1} \\ \kappa_3 \left\{ \begin{array}{c} [v\ \gamma\ z|v]_{\underline{\boldsymbol{C}\{\ \}\{\ \}}} \boldsymbol{C}\{[x\ \gamma\ z|v]_{\underline{\breve{\boldsymbol{A}}}} \breve{\boldsymbol{A}}\}\{[y\ \gamma\ z|v]_{\underline{\breve{\boldsymbol{B}}}} \breve{\boldsymbol{B}}\{[y\ \gamma\ z|v]_{\underline{\breve{\boldsymbol{A}}}} \breve{\boldsymbol{A}}\}\} \\ \| \\ \boldsymbol{C}\{\breve{\boldsymbol{A}}^x\}\{\breve{\boldsymbol{B}}^y\{\breve{\boldsymbol{A}}^y\}\}\ \gamma\ \breve{\boldsymbol{C}}\{\breve{\boldsymbol{A}}^z\}\{\breve{\boldsymbol{B}}^z\{\breve{\boldsymbol{A}}^z\}\} \end{array} \right\} \end{array}\right] .$$

Crucially, we see that in the conclusion, z is replaced by $\breve{\boldsymbol{C}}\{\breve{\boldsymbol{A}}^z\}\{\breve{\boldsymbol{B}}^z\{\breve{\boldsymbol{A}}^z\}\}$, which contains two copies of $\breve{\boldsymbol{A}}^z$: one inherited from x occurring in $\boldsymbol{B}\{x\}$ and one from x occurring in $\boldsymbol{C}\{x\}\{y\}$. This pattern will lead to exponential blow-up for the size of the derivation, but it can be controlled with explicit substitutions. We will factor out the repeated instances of $\breve{\boldsymbol{A}}^z$ and instead replace z by $\left\langle \breve{\boldsymbol{A}}^z \middle| x \right\rangle \left\langle \breve{\boldsymbol{B}}^z\{x\} \middle| y \right\rangle \breve{\boldsymbol{C}}\{x\}\{y\}$. This compression is realised in the corresponding composition by expansion steps inside χ_i and ω_j in Figure 6.3.

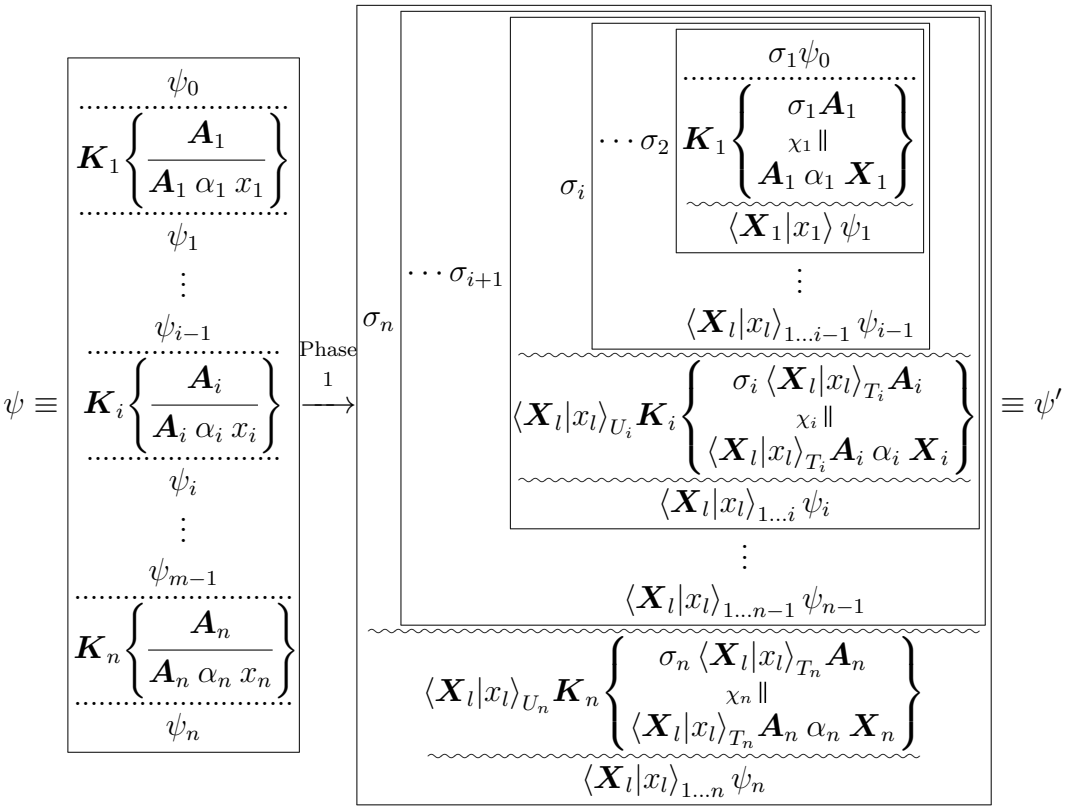
6.3 Main theorem

Theorem 6.3.1. *Given any derivation ϕ in* rKDTEq-OD *(resp.,* KDTEq-OD*), we can build a derivation ϕ' in* rKDT-ODS *(resp.,* KDT-ODS*) such that ϕ and ϕ' are structurally equivalent for the abstraction* kdt *and such that the size of ϕ' is polynomial in the size of ϕ.*

Proof. This proof works the same for rKDTEq-OD and KDTEq-OD, so we assume that it is for rKDTEq-OD. We refer to Figures 6.1 and 6.2. Note that we use Notation 5.1.15.

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 pre-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.

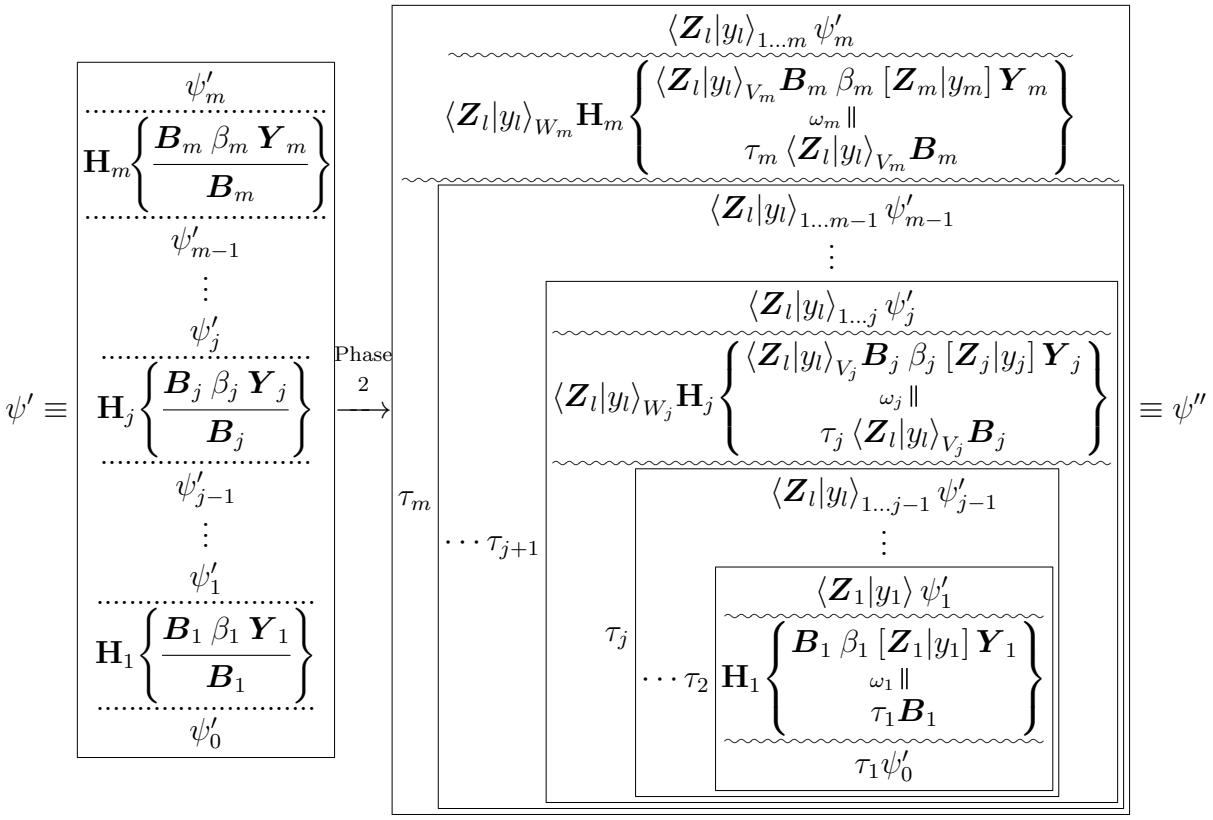
We first consider the equality steps which are instances of $=_1$, $=_3$, $=_5$, $=_7$, $=_9$ or $=_{11}$, as given in Figure 3.6 on page 37; that is, those unit-equality inference steps which create a unit travelling downwards in the derivation. Let $x_1, \ldots, x_n$ be the variables in ψ that correspond 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 6.1 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



Where:

$$
\begin{array}{rcl}
\sigma_i & = & [v\ \alpha_i\ x_i | v]_{\underline{\boldsymbol{A}_i}} \\
\breve{\boldsymbol{A}}_i^{\boldsymbol{C}} & \equiv & [\boldsymbol{C} | v]_{\underline{\boldsymbol{A}_i} \setminus \{x_1, \ldots, x_{i-1}\}}\ \breve{\boldsymbol{A}}_i \\
\boldsymbol{X}_i & \equiv & \left\langle \breve{\boldsymbol{A}}_1^{x_i} \middle| x_1 \right\rangle \cdots \left\langle \breve{\boldsymbol{A}}_{i-1}^{x_i} \middle| x_{i-1} \right\rangle \breve{\boldsymbol{A}}_i^{x_i} \\
T_i & = & \{x_1, \ldots, x_{i-1}\} \cap \underline{\boldsymbol{A}_i} \\
U_i & = & \{x_1, \ldots, x_{i-1}\} \setminus \underline{\boldsymbol{A}_i} \\
\chi_i & & \text{is given in Figure 6.3.}
\end{array}
$$

Figure 6.1: Phase 1 of the construction in Theorem 6.3.1



Where:

$$
\begin{aligned}
\underline{\boldsymbol{Y}_j} &= \{y_j\} \\
\tau_j &= \langle \boldsymbol{Y}_j | y_j \rangle\ [v\ \beta_j\ y_j | v]_{\underline{\boldsymbol{B}_j}} \\
\widehat{\boldsymbol{B}}_j^{\boldsymbol{C}} &\equiv [\boldsymbol{C} | v]_{\underline{\boldsymbol{B}_j} \setminus \{y_1, \ldots, y_{j-1}\}}\ \widehat{\boldsymbol{B}}_j \\
\boldsymbol{Z}_j &\equiv \left\langle \widehat{\boldsymbol{B}}_1^{y_j} \middle| y_1 \right\rangle \cdots \left\langle \widehat{\boldsymbol{B}}_{j-1}^{y_j} \middle| y_{j-1} \right\rangle \widehat{\boldsymbol{B}}_j^{y_j} \\
V_j &= \{y_1, \ldots, y_{j-1}\} \cap \underline{\boldsymbol{B}_j} \\
W_j &= \{y_1, \ldots, y_{j-1}\} \setminus \underline{\boldsymbol{B}_j} \\
\omega_j &\quad \text{is given in Figure 6.3.}
\end{aligned}
$$

Figure 6.2: Phase 2 of the construction in Theorem 6.3.1

invalid inference steps are arranged as in the figure. Under the assumptions on π mentioned above, no variable x_i appears in formulae $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_i$, for $1 \leq i \leq n$; on the other hand, x_i might appear in $\boldsymbol{A}_{i+1}, \ldots, \boldsymbol{A}_n$.

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:

$$\psi \equiv \boxed{\begin{array}{l} \kappa \left\{ \dfrac{\boldsymbol{A}_i}{\boldsymbol{A}_i\ \alpha_i\ x_i} \right\} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \kappa' \left\{ \dfrac{\boldsymbol{B}_j\ \beta_j\ x_i}{\boldsymbol{B}_j} \right\} \end{array}} \quad .$$

Here, x_i would be assigned an instance of $\breve{\boldsymbol{A}}_i$ for a down-merge at the top and an instance of $\widehat{\boldsymbol{B}}_j$ for an up-merge at the bottom. By Corollary 5.2.5, these conflicting substitutions can be reconciled via the eversion construction

$$\begin{array}{c} \left[\widehat{\boldsymbol{B}}_j \middle| v\right]_{\underline{\boldsymbol{A}_i}} \breve{\boldsymbol{A}}_i \\ \| \\ \left[\breve{\boldsymbol{A}}_i \middle| v\right]_{\underline{\boldsymbol{B}_j}} \widehat{\boldsymbol{B}}_j \end{array} \quad .$$

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 rKDT-ODS derivation χ_i, for $1 \leq i \leq n$, shown in Figure 6.3. Each variable x_i is replaced by a formula $\boldsymbol{X}_i$, whose purpose is to make a down-merge of $\boldsymbol{A}_i$ via $\boldsymbol{A}_i$ and α_i possible. The down-merge is χ'_i, in the version of Proposition 5.1.20. $\boldsymbol{X}_i$ is constituted by the formula $\breve{\boldsymbol{A}}_i$ 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, $\boldsymbol{X}_i$, whose structure matches the surroundings (to be amenable to a merge) but whose value remains that of x_i. Those variables of $\breve{\boldsymbol{A}}_i$ that are not in $\{x_1, \ldots, x_{i-1}\}$ are set to x_i, in $\breve{\boldsymbol{A}}_i^{x_i}$. The other variables of $\breve{\boldsymbol{A}}_i$ must be replaced by substitutions that could match the formulae

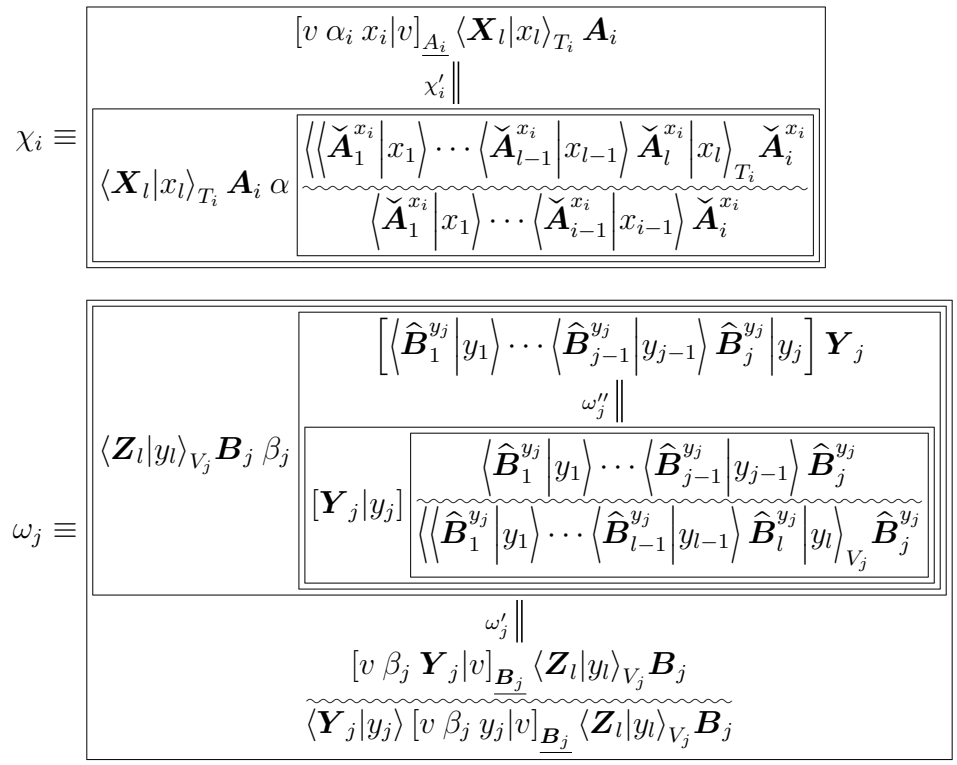


Figure 6.3: Auxiliary derivations for Phases 1 and 2 in Theorem 6.3.1.

generated by the $\chi_1, \ldots, \chi_{i-1}$ above χ_1 in the derivation; those formulae are $\breve{\boldsymbol{A}}_1^{x_1}, \ldots, \breve{\boldsymbol{A}}_{i-1}^{x_{i-1}}$ and are matched by $\breve{\boldsymbol{A}}_1^{x_i}, \ldots, \breve{\boldsymbol{A}}_{i-1}^{x_i}$. At its top, χ_i generates the substitution σ_i, which does not change the value of the variables it applies to, and which is propagated upwards in the derivation. At its bottom, χ_i generates the substitution $\langle \boldsymbol{X}_i | x_i \rangle$, which is propagated downwards in the derivation and which also does not change values because $\pi \boldsymbol{X}_i = \pi x_i$. The rest of the construction in Figures 6.1 and 6.2 is bookkeeping, mainly relying on having maximally renamed apart all variables so that we can move substitutions without capturing any.

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 Figure 3.6 on page 37; that is, those unit-equality inference steps which create a unit travelling upwards in the derivation. For $1 \leq j \leq m$, $\boldsymbol{B}_j$ takes the place of $\boldsymbol{A}_i$ and $\boldsymbol{Y}_j$ that of x_i. One difference is that now $\boldsymbol{Y}_j$ might be one of the formulae $\boldsymbol{X}_i$s, and not just a variable. That said, each $\boldsymbol{Y}_j$ still only contains one variable (potentially in multiple copies), say y_j, and we note that y_j does not appear in $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_j$ and might appear in $\boldsymbol{B}_{j+1}, \ldots, \boldsymbol{B}_m$. In Phase 2, each formula $\boldsymbol{Z}_j$ plays the same role as $\boldsymbol{X}_i$ in Phase 1, and the derivation ω_j, shown in Figure 6.3, 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 $\langle \boldsymbol{Y}_j | y_j \rangle$ but for the rest its role is similar. The result of Phase 2 is a derivation ψ'' in $\mathsf{rKDT\text{-}ODS}$.

Structural Equivalence. We must show that $\mathsf{kdt}\,\phi \equiv \mathsf{kdt}\,\pi\psi = \mathsf{kdt}\,\pi\psi'' \equiv \mathsf{kdt}\,\phi'$. The vertical composition steps added to ψ by the construction above are compositions by expansions and the inference steps of the merge derivations χ'_i and ω'_j and of the eversions ω''_j. Compositions by expansions are turned by kdt into equalities. The merge derivations fall into the remit of Propositions 5.1.20 or 5.1.22; for example, for ω'_j, we have

$$
\begin{aligned}
&[\boldsymbol{Y}_j | y_j] \left\langle \left\langle \widehat{\boldsymbol{B}}_1^{y_j} \middle| y_1 \right\rangle \cdots \left\langle \widehat{\boldsymbol{B}}_{l-1}^{y_j} \middle| y_{l-1} \right\rangle \widehat{\boldsymbol{B}}_l^{y_j} \middle| y_l \right\rangle_{V_j} \widehat{\boldsymbol{B}}_j^{y_j} \\
&\equiv [\boldsymbol{Y}_j | v]_{\underline{\boldsymbol{B}_j}} \left\langle \left\langle \widehat{\boldsymbol{B}}_1^{y_l} \middle| y_1 \right\rangle \cdots \left\langle \widehat{\boldsymbol{B}}_{l-1}^{y_l} \middle| y_{l-1} \right\rangle \widehat{\boldsymbol{B}}_l^{y_l} \middle| y_l \right\rangle_{V_j} \widehat{\boldsymbol{B}}_j \\
&\equiv [\boldsymbol{Y}_j | v]_{\underline{\boldsymbol{B}_j}} \langle \boldsymbol{Z}_l | y_l \rangle_{V_j} \widehat{\boldsymbol{B}}_j
\end{aligned}
$$

and either $\mathsf{kdt}\,\pi[\boldsymbol{Y}_j | v]_{\underline{\boldsymbol{B}_j}} \langle \boldsymbol{Z}_l | y_l \rangle_{V_j} \widehat{\boldsymbol{B}}_j \equiv \mathsf{0}$ or $\mathsf{kdt}\,\pi[\boldsymbol{Y}_j | v]_{\underline{\boldsymbol{B}_j}} \langle \boldsymbol{Z}_l | y_l \rangle_{V_j} \widehat{\boldsymbol{B}}_j \equiv \mathsf{1}$.

Therefore we obtain $\mathsf{kdt}\,\pi\omega'_j \equiv \mathsf{kdt}\,\pi\,\langle \boldsymbol{Z}_l|y_l\rangle_{V_j}\boldsymbol{B}_j \equiv \mathsf{kdt}\,\pi\boldsymbol{B}_j$ (which could be identical to a unit if β_j is in $\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\dots\}$). To the eversions ω''_j we can apply Lemma 4.2.9 because each section of $\pi\omega''_j$ is either equal to $\mathsf{0}$ or to $\mathsf{1}$; this is because the only variable that ω''_j contains is y_j, which derives from a unit in some equality step of ϕ. Therefore either $\mathsf{kdt}\,\pi\omega''_j \equiv \mathsf{0}$ or $\mathsf{kdt}\,\pi\omega''_j \equiv \mathsf{1}$. All this means that the vertical structure of $\pi\psi''$ is not altered by applying kdt. It remains to be checked that all the substitutions applied in various places do not alter the value of the sections of ϕ. Take, for example, $\sigma_i = [v\,\alpha_i\,x_i|v]_{\underline{\boldsymbol{A}_i}}$. Each variable v_l is composed by α_i to x_i: if $\alpha_i \in \{\vee,\wedge\}$ then πx_i is the unit of α_i, and if $\alpha_i \in \{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\dots\}$ then $\pi x_i \equiv \pi v_l$. In both cases, the value of the expression is preserved. In those substitutions where explicit substitutions are present, like τ_j, the explicit substitutions are collapsed by kdt because they are composed exclusively of units.

Complexity.

We establish upper bounds for the width and height of ψ'. The width of ϕ, say w, dominates the size of $A_1, \dots, A_n$, 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 5.1.10). Therefore,

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

Because of Lemma 5.1.10, 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,

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

and this is the width of ϕ'. Because of Lemmas 5.1.10 and 5.2.1, the height of ω_j is also $O(w^7h^5)$, which dominates h', therefore the height of ϕ' is $O(w^7h^5)$. □

Remark 6.3.2. The construction to eliminate the unit-equality inference steps introduces nested atom connectives even if none appear in the given derivation. This happens because of the need for merges and eversions and their propagation via the variables standing for the units to be eliminated. However, those nested atom connectives only belong to subformulae whose value, as detected by kdt, is a unit. Therefore, if no proper decision trees appear in the given derivation, none appear in the transformed one, and a standard Tarskian semantics applies to it.

We can show exactly where the 'merge-eversion' method is turned from exponential to polynomial by the use of explicit substitutions. It is the composition by expansion at the bottom of the merge inside χ_i, and the similar composition inside ω_j. In those expansions, several explicit substitutions are factored out, relying on the fact that the formula involved uses a number of 'formula patterns' – the $\boldsymbol{A}_i$s, – which is linear in the size of the derivation. Without the factorisation, those formulae would accumulate exponentially, as described in Section 6.2: one $\boldsymbol{A}_1$ would become two $\boldsymbol{A}_1$s inside $\boldsymbol{A}_2$, then four inside $\boldsymbol{A}_3$ and so on. However, since those patterns only contain one variable x_i (in multiple copies) they can be factored out because they are all the same formula.

At this point, another question naturally arises: *Given any derivation ϕ in* KDTEq-ODS*, can we build a derivation ϕ' in* KDT-ODS *such that ϕ and ϕ' are structurally equivalent for the abstraction* kdt *and such that the size of ϕ' is polynomial in the size of ϕ?* In other words, can the same result be obtained for derivations that contain explicit substitutions to start with? The answer appears to be negative. The problem is that, when explicit substitutions are present, units can be factored in and out of them. The 'merge-eversion' method relies on substituting units with formulae that depend on the unit-equality steps where the units are created or destroyed. Therefore, sharing the units into an explicit substitution creates conflicts. We consider this an interesting open problem because it might be solved by resorting to a more powerful notion of explicit substitution.

Indeed, as we mention in the introduction, we intend to develop a new notion of substitution by incremental steps that increase its expressive power; we would like these substitutions to be able to operate in proof systems

with negation, duplication, and first- and higher-order quantifiers. We might hope that the eventual notion of substitution achieves a fixed point for the above problem, so that whenever we have a derivation with substitutions and unit equalities, the unit equalities can be eliminated at polynomial cost on the size of the derivation. In Chapter 7, we introduce a more expressive notion of substitution which may solve this problem for the strictly linear, propositional case, but this remains to be fully investigated.

Chapter 7

P-Simulation of Substitution Frege

In this chapter, we investigate the proof complexity properties of a strictly linear system with a notion of explicit substitutions. We do this by way of comparison to substitution Frege systems [15], which are a class of p-equivalent systems for propositional logic with a substitution rule. No system is known to be more powerful than substitution Frege systems in compressing the size of proofs, and this therefore serves as a benchmark.

System rKDTEq-OD is a conservative extension of System SKS-OD, and Bruscoli and Guglielmi have shown that SKS-OD p-simulates Frege systems [6]. Therefore, thanks to Theorem 6.3.1, we know that System rKDT-ODS p-simulates Frege systems. Then, a natural question arises: Given that rKDT-ODS has a notion of substitution, can it simulate substitution Frege systems? We do not know, but that seems unlikely, and it has not been possible with our current understanding of rKDT-ODS.

The difficulty arises from the fact that in substitution Frege systems, compression can be achieved by reusing formulae, so that from a formula A in a substitution Frege proof, both σA and τA can be obtained, where σ and τ are unrelated substitutions. However, in the formalism ODS, substitutions must go all the way through a proof, and so it is not obvious how to achieve this compression: duplicating A will not work, and neither will applying σ and τ in succession.

Instead, we can increase the power of ODS substitutions in a natural way by a generalisation, and that indeed allows us to p-simulate substitution Frege. The generalisation allows for more control over where a substitution

applies, so that only certain occurrences of a variable will be affected by a substitution. This seems to be well-motivated from a mathematical perspective, because it extends our ability to factor out subderivations from those which are identical to those which are similar, varying only in the objects concerned.

This results in a natural overall translation scheme, in particular for the Frege substitution itself. There are intricacies that we must deal with, but they only have to do with strict linearity, to make sure that there is no exponential propagation of variables across the translated proof.

First, we introduce the generalised substitution for open deduction, which we name 'supersubstitution', then, we present a particular substitution Frege system, and then we define a translation that achieves the desired p-simulation.

7.1 Supersubstitution

We introduce a more general notion of substitution than the one in previous sections. The problem we want to solve is the following. Given the formula

$$A\{B\{C_1\}\}\cdots\{B\{C_n\}\} \quad ,$$

how can we factor out $B\{\ \}$ efficiently? We introduce a notion which we call 'supersubstitution', which allows for the substitution of $B\{\ \}$ to be shared as much as possible. With this notion, we can solve the problem with the formula

$$\left\langle C_1\middle|x^{\{1\}}\right\rangle\cdots\left\langle C_n\middle|x^{\{n\}}\right\rangle\langle B(x)|y\rangle\, A\left\{y^{\{1\}}\right\}\cdots\left\{y^{\{n\}}\right\} \quad ,$$

which can be flattened in two successive steps as

$$\left\langle C_1\middle|x^{\{1\}}\right\rangle\cdots\left\langle C_n\middle|x^{\{n\}}\right\rangle A\left\{\ \lceil B\{x\}\rceil^{\{1\}}\ \right\}\cdots\left\{\ \lceil B\{x\}\rceil^{\{n\}}\ \right\} \qquad \text{and}$$
$$A\left\{\ \lceil B\{C_1\}\rceil^{\{1\}}\ \right\}\cdots\left\{\ \lceil B\{C_n\}\rceil^{\{n\}}\ \right\} \quad .$$

The idea is that the labels $\{1\},\ldots,\{n\}$ are inherited by what is substituted into their 'ranges' $\lceil\ \rceil^{\{1\}},\ \ldots,\ \lceil\ \rceil^{\{n\}}$, and then a substitution $\left\langle C_i\middle|x^{\{i\}}\right\rangle$ is applied only inside the range $\lceil\ \rceil^{\{i\}}$.

The introduction of such a notion is motivated from a mathematical perspective. A lemma may be used multiple times in a proof with different inputs and we can compress the proof by allowing these similar subproofs to be factored out.

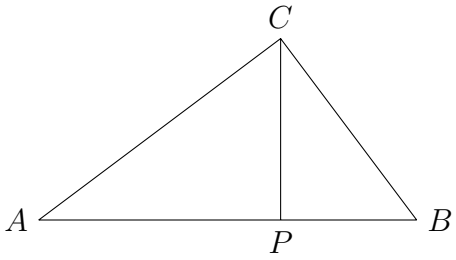


Figure 7.1: The triangle ABC in Example 7.1.1

Example 7.1.1. Consider the following sketch proof of Pythagoras' theorem, which makes use of the similarity of the two smaller triangles $\triangle ACP$ and $\triangle BCP$ to $\triangle ABC$, shown in Figure 7.1.

$$\dfrac{\boxed{\dfrac{\dfrac{\angle ACP = \frac{\pi}{2} - \angle BAC = \angle CBA}{\triangle CBA \sim \triangle ACP}}{|AC|^2 = |AB||AP|}} \wedge \boxed{\dfrac{\dfrac{\angle BCP = \frac{\pi}{2} - \angle ABC = \angle CAB}{\triangle CAB \sim \triangle BCP}}{|BC|^2 = |BA||BP|}}}{\dfrac{|AC|^2 + |BC|^2 = |AB||AP| + |BA||BP|}{|AC|^2 + |BC|^2 = |AB|^2}}$$

The two subderivations shown in boxes are similar, differing only in that where the left has an A the right has a B and vice versa; we consider this a repetition of similar derivations, even though they are not identical. We can consider the derivation

$$\dfrac{\dfrac{\angle xCP = \frac{\pi}{2} - \angle yxC = \angle Cyx}{\triangle Cyx \sim \triangle xCP}}{|xC|^2 = |xy||xP|} \quad ;$$

under the substitution $[A|x, B|y]$ we obtain the derivation on the left and under the substitution $[B|x, A|y]$ we obtain the derivation on the right. We would like to be able to use supersubstitutions to factor out the repeated derivation, and obtain the following:

$$\left\langle A\middle|x^{\{1\}}, B\middle|y^{\{1\}}\right\rangle \left\langle B\middle|x^{\{2\}}, A\middle|y^{\{2\}}\right\rangle \left\langle \dfrac{\dfrac{\angle xCP = \frac{\pi}{2} - \angle yxC = \angle Cyx}{\triangle Cyx \sim \triangle xCP}}{|xC|^2 = |xy||xP|} \middle| z \right\rangle z^{\{1\}} \wedge z^{\{2\}}$$

$$\underbrace{\left\langle A\middle|x^{\{1\}}, B\middle|y^{\{1\}}\right\rangle \left\langle B\middle|x^{\{2\}}, A\middle|y^{\{2\}}\right\rangle \left\lceil |xC|^2 = |xy||xP| \right\rceil^{\{1\}} \wedge \left\lceil |xC|^2 = |xy||xP| \right\rceil^{\{2\}}}$$

$$\dfrac{\dfrac{\left(|AC|^2 = |AB||AP|\right) \wedge \left(|BC|^2 = |BA||BP|\right)}{|AC|^2 + |BC|^2 = |AB||AP| + |BA||BP|}}{|AC|^2 + |BC|^2 = |AB|^2}$$

That is, the names of the points A and B are stored in supersubstitutions so that the derivation can be shared, and the variables $z^{\{1\}}$ and $z^{\{2\}}$ 'remember' which name goes on which variable.

This example goes beyond the current scope of the work with supersubstitutions, because it makes use of predicates and functions and a more complicated proof system. However, it serves to show the intended direction of development.

Definition 7.1.2. We extend the language of pre-derivations as follows.

Let $\mathcal{R}$ be a countable set of elementary ranges and let $\circ \notin \mathcal{R}$ denote the null range; let $\mathcal{R}^+ = \mathcal{R} \cup \{\circ\}$.

We update Definition 2.1.1 of pre-derivations by replacing $\mathcal{V}$ by $\mathcal{V}^{\mathcal{P}(\mathcal{R})}$, so that each variable is associated with a *range*, a (possibly empty) set of elementary ranges.

We also replace the definition of composition by explicit substitution by $\left\langle \mathcal{P} \middle| \mathcal{V}^{\mathcal{R}^+} \right\rangle \mathcal{P}$ so that each explicit substitution is associated with either an elementary range or the null range. If the explicit substitution is associated with an elementary range we call it a *supersubstitution*, and if is associated with the null range we may write $\langle A|x\rangle$ instead of $\left\langle A\middle|x^{\{\circ\}}\right\rangle$; this coincides with the notion of explicit substitution used in previous chapters. If every variable in a formula A is associated with every elementary range r in the set $R \subseteq \mathcal{R}$, we may write $\lceil A \rceil^R$. For a variable $y^{\emptyset}$ whose range is the empty set, we may write y.

If all the free variable occurrences in an pre-derivation $\phi\{x_1^{R_1 \cup R}\} \cdots \{x_h^{R_h \cup R}\}$ contain the same range R, we can denote the pre-derivation as

$$\lceil \phi\{x_1^{R_1}\} \cdots \{x_h^{R_h}\} \rceil^R ;$$

then, x^R and $\lceil x \rceil^R$ are identical expressions, and so are $\lceil \lceil \phi \rceil^R \rceil^{R'}$ and $\lceil \phi \rceil^{R \cup R'}$.

The syntactic identity relation $\equiv$ and the size function take ranges into account: $x^R \equiv x^{R'}$ if and only if $R = R'$, and $\left|x^R\right| = 1 + |R|$.

Establishing that variables have ranges necessitates updating some of the notions presented in previous sections. In particular, we must consider the flattening function and the correctness of instances of composition by expansion.

Example 7.1.3. In general, supersubstitutions cannot be 'precomposed' and simplified the same way as substitutions can be. For example, the substitutions $\langle \mathsf{0}|y\rangle \langle \mathsf{1}|y\rangle \langle y \vee y|x\rangle$ and $\langle \mathsf{1}|y, \mathsf{1} \vee \mathsf{1}|x\rangle$ are such that for any formula A,

$$\mathsf{fl}\, \langle \mathsf{0}|y\rangle \langle \mathsf{1}|y\rangle \langle y \vee y|x\rangle\, A \equiv \mathsf{fl}\, \langle \mathsf{1}|y, \mathsf{1} \vee \mathsf{1}|x\rangle\, A \quad ,$$

so we could simplify the former to the latter purely as an operation on substitutions.

However, if we have a compound supersubstitution $\left\langle \mathsf{0}\middle|y^{\{1\}}\right\rangle \left\langle \mathsf{1}\middle|y^{\{2\}}\right\rangle \langle y \vee y|x\rangle$, we cannot simplify this to $\left\langle \mathsf{1}\middle|y^{\{2\}}, \mathsf{1} \vee \mathsf{1}\middle|x\right\rangle$ or similar, because we will get a different result depending on whether we apply the substitution to $x^{\{1\}}$ or $x^{\{2\}}$: in the first case we would get $\mathsf{0} \vee \mathsf{0}$ and in the second $\mathsf{1} \vee \mathsf{1}$. We therefore need to be careful in our definition of the flattening function.

Definition 7.1.4. Given a formula A, we can obtain its *flat expansion* $\mathsf{fl}\, A$ by applying all of the substitutions:

- If $A \in \mathcal{V}^{\mathcal{R}} \cup \mathcal{U}$ then $\mathsf{fl}\, A \equiv A$.
- If $A \equiv B\, \alpha\, C$ then $\mathsf{fl}\, A \equiv \mathsf{fl}\, B\, \alpha\, \mathsf{fl}\, C$.
- If $A \equiv \left\langle B\middle|x^{\{r\}}\right\rangle x^R$ then $\mathsf{fl}\, A \equiv \lceil \mathsf{fl}\, B \rceil^R$ if either $r \in R$ or $r = \circ$ and $\mathsf{fl}\, A \equiv x^R$ otherwise
- If $A \equiv \left\langle B\middle|x^{\{r\}}\right\rangle (C\ \beta\ D)$ then $\mathsf{fl}\, A \equiv (\mathsf{fl} \left\langle B\middle|x^{\{r\}}\right\rangle C)\ \beta\ (\mathsf{fl} \left\langle B\middle|x^{\{r\}}\right\rangle D)$
- If $A \equiv \left\langle B\middle|x^{\{r\}}\right\rangle \left\langle C\middle|y^{\{q\}}\right\rangle D$, for $q \in \{\circ\} \cup \mathcal{R}$ then $\mathsf{fl}\, A \equiv \mathsf{fl}(\left\langle B\middle|x^{\{r\}}\right\rangle (\mathsf{fl} \left\langle C\middle|y^{\{q\}}\right\rangle D))$.

The correctness of an instance of composition by expansion was originally defined as depending only on the flat expansion of those formulae being composed, so that $\overset{\phi}{\underset{\psi}{\sim}}$ was correct whenever $\mathsf{fl}\, \mathsf{cn}\, \phi \equiv \mathsf{fl}\, \mathsf{pr}\, \psi$. For this, we

needed to show that this equivalence could be checked in polynomial time on $|\mathsf{cn}\,\phi| + |\mathsf{pr}\,\psi|$. However, to do the same for formulae with supersubstitutions is difficult.

The reason for this is that we must keep track of the ranges in order to know which supersubstitutions will apply to a variable; for example, given a formula $\sigma\,\langle A\{y\}|x\rangle\,B$, in order to know which supersubstitutions in σ will apply to y, we need to keep track of the information about every occurrence of x in B. Therefore it's not obvious how to compare two formulae without computing both flat expansions, and these can be exponentially larger than the

The work on supersubstitutions is in the early stages of its development. Future work may lead to a polynomial-time algorithm to decide the syntactic equivalence of $\mathsf{fl}\,A$ and $\mathsf{fl}\,B$ for any formulae A and B, or indeed to a different notion altogether of how to share the repeated structures within a derivation. For now we restrict ourselves to a more limited definition of composition by expansion, which is nevertheless sufficient to prove the result in this chapter.

Definition 7.1.5. A pair of formulae A and B are *composable by expansion* if:

- Their every variable occurrence is associated with the range $\emptyset$ and $\mathsf{fl}\,A \equiv \mathsf{fl}\,B$.
- $A \equiv \langle C|y\rangle\,\sigma K\{y\}$ and $B \equiv \langle C|y\rangle\,\sigma K\{\lceil C\rceil^R\}$, where $K\{\ \}$ is either flat or of the form $\langle H\{\ \}|z\rangle\,D$ for $H\{\ \}$ flat, y does not appear as a substitution variable in σ nor does y occur free in C, and no substitution variable in τ occurs free in C.
- $A \equiv \langle \tau C|x\rangle\,D$ and $B \equiv \tau\,\langle C|x\rangle\,D$, where no substitution variable of τ occurs free in D.
- $A \equiv \left\langle C\middle|x^{\{r\}}\right\rangle \sigma K\{x^R\}$ and $B \equiv \left\langle C\middle|x^{\{r\}}\right\rangle \sigma K\{\lceil C\rceil^R\}$, where no variables appear in C, $r \in R$, $K\{\ \}$ is either flat or of the form $\langle H\{\ \}|z\rangle\,D$ for $H\{\ \}$ flat, and x does not appear as a substitution variable in σ.
- $A \equiv \sigma D$ and $B \equiv D$, where no substitution variable in σ appears in D.

A pre-derivation $\underset{\psi}{\overset{\phi}{\sim}}$ is *correct* if $\mathsf{cn}\,\phi$ and $\mathsf{pr}\,\psi$ are composable by expansion, and we call the formalism consisting of all correct pre-derivations ODSS.

Example 7.1.6. The following pre-derivation is correct, provided that w, x and y occur free in neither A nor B:

$$\dfrac{\left\langle \mathsf{0}\middle|v^{\{3\}}\right\rangle \quad \boxed{\begin{array}{c}\langle A|w\rangle \quad \boxed{\begin{array}{c}\langle B|x\rangle \quad \boxed{\begin{array}{c}\langle A\wedge B|y\rangle \left\langle y^{\{3\}}\middle|z\right\rangle z\\ \hline \langle A\wedge B|y\rangle \left\langle \ulcorner A\wedge B\urcorner^{\{3\}}\middle|z\right\rangle z\end{array}} \\ \hline \langle B|x\rangle \left\langle y^{\{3\}}\middle|z\right\rangle z\end{array}} \\ \hline \langle A|w\rangle \left\langle y^{\{3\}}\middle|z\right\rangle z\end{array}}}{\left\langle \mathsf{0}\middle|v^{\{3\}}\right\rangle \langle A|w\rangle \langle B|x\rangle \langle A\wedge B|y\rangle \left\langle \ulcorner[\mathsf{0}|v]\,(A\wedge B)\urcorner^{\{3\}}\middle|z\right\rangle z}$$

We limit ourselves here to only those substitution reduction steps which we will use in the construction in the proof of Theorem 7.3.4; other reduction steps could be added to these but we leave extending the formalism to future investigations into supersubstitutions. This means that not every $\mathcal{S}$-ODS derivation is a $\mathcal{S}$-ODSS derivation, because it may make use of the more free composition by expansion permitted in the ODS formalism.

Soundness follows from the fact that ranges are inherited, so that they go all the way through the derivation, and that derivations are only identical if their ranges are identical.

Example 7.1.7. The composition $\dfrac{x}{\langle x|y^{\{1\}}\rangle\, y^{\{1\}}}$ is not correct: applying $\langle x|y^{\{1\}}\rangle$ to $y^{\{1\}}$ results in $x^{\{1\}}$, which is not x. This therefore precludes unsound derivations such as

$$\langle \mathsf{1}|x\rangle \,\langle \mathsf{0}|x^{\{1\}}\rangle \,\dfrac{x}{\langle x|y^{\{1\}}\rangle\, y^{\{1\}}} \quad ,$$

whose premise is equal to $\mathsf{1}$ and whose conclusion is equal to $\mathsf{0}$.

7.2 Substitution Frege

We introduce some standard notions in proof complexity. Good references for this material are [13] and [14]. We can choose any convenient system among the many available Frege systems without prejudice to the complexity class; they are all equivalent modulo polynomial-time translations. We use one of Łukasiewicz's set of axioms [23, 26] because it only has three axioms that neatly confine the particular challenges of the translation we must perform,

namely dealing with weakening and contraction by maintaining strict linearity. To those, we add axioms for the units, which are convenient for the substitution rule.

$$\mathsf{F}_1 \equiv x \to (y \to x) \qquad \mathsf{F}_4 \equiv \mathsf{t}$$
$$\mathsf{F}_2 \equiv (x \to (y \to z)) \to ((x \to y) \to (x \to z)) \qquad \mathsf{F}_5 \equiv \neg\mathsf{f}$$
$$\mathsf{F}_3 \equiv (\neg x \to \neg y) \to (y \to x)$$

$$\mathsf{mp}\ \frac{x \quad x \to y}{y}$$

$$\mathsf{usf}\ \frac{A}{[\mathsf{f}|x]\,A} \qquad \mathsf{ust}\ \frac{A}{[\mathsf{t}|x]\,A}$$

Figure 7.2: System sFrege.

Definition 7.2.1. A *Frege proof system* is a finite collection of axioms and inference rules. A *Frege proof* of A_n is a sequence of formulae $A_1, \ldots, A_n$ such that each A_i is either an axiom or is the conclusion of an inference step with premises from $A_1, \ldots, A_{i-1}$. A *substitution Frege proof system* is a Frege proof system in which there is a rule $\dfrac{A}{\sigma A}$, where σ is a substitution. The substitution Frege system sFrege is given in Figure 7.2, where F_1, F_2, F_3, F_4 and F_5 are axioms, the rule mp is *modus ponens*, and the rule schemes usf and ust are called *unit substitutions*. System Frege is obtained from sFrege by removing the rules usf and ust.

Remark 7.2.2. System Frege, and therefore System sFrege, are sound and complete for propositional classical logic.

The connectives in sFrege are different from the connectives in rKDT. However, polynomial-time translations can be established in both directions via De Morgan's laws, when rKDT atoms are not nested. The restriction to unnested atoms is not a limitation because we are only interested in standard propositional logic. This choice of language and axioms for sFrege has two

advantages: we only must translate three non-trivial axioms, and the translation makes it clear where strict linearity needs to be dealt with. Indeed, axiom F_1 realises weakening (on y) and axiom F_2 realises contraction (on the negative occurrence of x).

Definition 7.2.3. A proof system $\mathcal{S}'$ *p-simulates* a proof system $\mathcal{S}$ if a proof of A in $\mathcal{S}$ can be transformed into a proof of A in $\mathcal{S}'$ in polynomial time.

One interesting feature of sFrege is that it only substitutes (formulae equivalent to) units in the usf and ust rules. It is shown in Lemma 17 of [12] that substituting units is enough to achieve the same power as substituting formulae. Only having to deal with unit substitutions exempts us from introducing technical details that obscure the translation, which is already complicated.

Remark 7.2.4. Because of Reckhow's theorem, System sFrege p-simulates any substitution Frege system.

7.3 P-Simulation

The language of rKDT is richer than that of sFrege. Still, we can obtain a natural correspondence between the two languages if we do not express any decision tree in rKDT, *i.e.*, if we abolish nested atoms. There is a natural isomorphism between sFrege formulae and the so restricted rKDT ones, obtained from $(x \to y) \leftrightsquigarrow (\bar{x} \vee y)$ and $\neg x \leftrightsquigarrow \bar{x}$ and De Morgan's laws.

To obtain the desired p-simulation, we must use the Merge Lemma by mimicking arbitrary formulae without changing the semantic value of a given formula. To do this, we use variables to control the various constructions. We start by translating sFrege formulae into open rKDT formulae, and then we rename apart all the variables to achieve the necessary freedom.

Much of the technicality in the proof of Theorem 7.3.4 arises from the condition of strict linearity. Without this condition we could simulate an instance of modus ponens via the $\mathsf{KDTEq}^{=}$-OD derivation

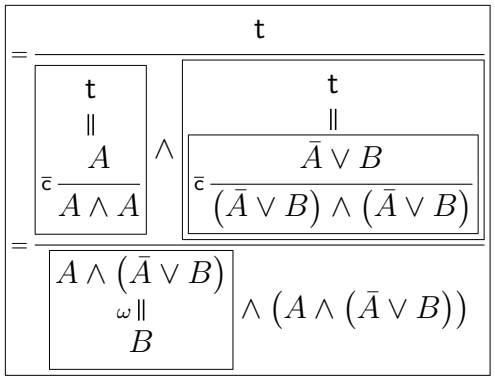
$$\vee\check{\wedge}\,\frac{\boldsymbol{A}\wedge\left(\bar{\boldsymbol{A}}\vee\boldsymbol{B}\right)}{\begin{array}{c}\boxed{\begin{array}{c}\boldsymbol{A}\wedge\bar{\boldsymbol{A}}\\ \chi\|\\ \rho\check{\boldsymbol{A}}\\ \|\,\mathsf{KEq}\\ \mathsf{0}\end{array}}\vee\boldsymbol{B}\\ =\\ \boldsymbol{B}\end{array}}\quad,$$

where χ is a merge as in Lemma 5.1.17 and ρ is an actual substitution which puts either $\mathsf{0}\wedge\mathsf{1}$ or $\mathsf{1}\wedge\mathsf{0}$ onto every leaf of $\check{A}$. Indeed this is the core structure of the simulation that we give, but under the constraints of strict linearity we are obliged to construct the following:

$$\psi\equiv\vee\check{\wedge}\,\frac{\begin{array}{c}\boxed{\begin{array}{c}[v\vee\mathsf{0}|v]_{\underline{\boldsymbol{A}}}\boldsymbol{A}\\ \|\\ \boldsymbol{A}\vee\check{\boldsymbol{A}}^{\mathsf{0}}\end{array}}\wedge\left(\bar{\boldsymbol{A}}\vee\boldsymbol{B}\right)\\ \\ \boxed{\begin{array}{c}\boldsymbol{A}\wedge\bar{\boldsymbol{A}}\\ \chi\|\\ \rho\check{\boldsymbol{A}}\end{array}}\vee\left(\check{\boldsymbol{A}}^{\mathsf{0}}\vee\boldsymbol{B}\right)\end{array}}\quad.$$

This propagates structures through the overall proof, which we can control with explicit substitutions.

In addition, we avoid having to duplicate and rearrange formulae, both of which are cumbersome in a strictly linear system, by means of explicit substitutions. In [6], it is shown that the proof system SKSg (in the calculus of structures formalism) p-simulates a Frege system. Translated to SKSg-OD, instances of modus ponens are simulated in this proof inside the following construction:



where ω is the derivation which does the modus ponens. This duplicates the formulae involved in the modus ponens so that they can be used multiple times in the proof. Using explicit substitutions, in rKDT-ODSS, this becomes:

$$\frac{\left\langle [v \vee \mathsf{0}|v]_{\underline{\boldsymbol{A}}}\boldsymbol{A}\middle|x\right\rangle \left\langle \bar{\boldsymbol{A}} \vee \boldsymbol{B}\middle|y\right\rangle \langle x \wedge y|z\rangle\, z}{\left\langle [v \vee \mathsf{0}|v]_{\underline{\boldsymbol{A}}}\boldsymbol{A}\middle|x\right\rangle \left\langle \bar{\boldsymbol{A}} \vee \boldsymbol{B}\middle|y\right\rangle \left\langle \begin{array}{c} \left([v \vee \mathsf{0}|v]_{\underline{\boldsymbol{A}}}\boldsymbol{A}\right) \wedge \left(\bar{\boldsymbol{A}} \vee \boldsymbol{B}\right) \\ {\scriptstyle\psi}\| \\ \rho\check{\boldsymbol{A}} \vee \left(\check{\boldsymbol{A}}^{\mathsf{0}} \vee \boldsymbol{B}\right) \end{array}\middle| z\right\rangle z} \quad ;$$

and the details of how we deal with what is propagated here is dealt with fully in the proof of Theorem 7.3.4.

By contrast to modus ponens, having introduced the mechanism of supersubstitution, instances of unit substitutions are comparatively simple to simulate; we need only take care that variables are properly distinguished in the translation from sFrege formulae to rKDT-ODSS formulae.

We suppose that we are given an sFrege proof $A, [\mathsf{f}|x]\, A, [\mathsf{t}|x]\, A, \ldots, B$ and we assume that we have substitutions σ_1 and σ_2 which can locally simulate the substitutions $[\mathsf{f}|x]$ and $[\mathsf{t}|x]$. We then simulate these instances of unit substitution as follows:

$$\frac{\sigma_2^{\{2\}}\sigma_1^{\{1\}} \langle A|w\rangle \left\langle w^{\{1\}}\middle|x\right\rangle \left\langle w^{\{2\}}\middle|y\right\rangle \cdots z}{\langle A|w\rangle \left\langle \lceil \sigma_1^{\{1\}} A \rceil^{\{1\}}\middle|x\right\rangle \left\langle \lceil \sigma_2^{\{2\}} A \rceil^{\{2\}}\middle|y\right\rangle \cdots z} \quad ;$$

that is, we restrict the ranges of σ_1 and σ_2 so that they cannot interfere with one another.

We now show in detail how these constructions can be put together to p-simulate any sFrege proof.

Definition 7.3.1. Let VF be the set that contains all the variables appearing in any sFrege formula. Given an sFrege formula A, we obtain the *corresponding open formula* $\lceil A \rceil^+$ in rKDT as follows:

$$\lceil \mathsf{f} \rceil^+ \equiv \lceil \mathsf{t} \rceil^- \equiv f \qquad \lceil x \rceil^+ \equiv (u_{x0} \,\mathbf{x}\, u_{x1}) \qquad \lceil \neg B \rceil^+ \equiv \lceil B \rceil^-$$

$$\lceil \mathsf{t} \rceil^+ \equiv \lceil \mathsf{f} \rceil^- \equiv t \qquad \lceil x \rceil^- \equiv (u_{\bar{x}1} \,\mathbf{x}\, u_{\bar{x}0}) \qquad \lceil \neg B \rceil^- \equiv \lceil B \rceil^+$$

$$\lceil B \to C \rceil^+ \equiv (\lceil B \rceil^- \vee \lceil C \rceil^+) \qquad \lceil B \to C \rceil^- \equiv (\lceil B \rceil^+ \wedge \lceil C \rceil^-)$$

where f and t are variables and u_{x0}, u_{x1}, $u_{\bar{x}0}$ and $u_{\bar{x}1}$ are also variables, for every $x \in \mathsf{VF}$. We call

$$\Sigma = \langle \mathsf{0}|f, \mathsf{1}|t\rangle \, \langle \mathsf{0}|u_{x0}, \mathsf{1}|u_{x1}, \mathsf{0}|u_{\bar{x}0}, \mathsf{1}|u_{\bar{x}1}\rangle_{x \in \mathsf{VF}}$$

the *standard value substitution*. We call

$$\Pi^{\mathsf{0}}_{\mathbf{x}} = \langle \mathsf{0}|u_{x0}, \mathsf{0}|u_{x1}, \mathsf{1}|u_{\bar{x}0}, \mathsf{1}|u_{\bar{x}1}\rangle \qquad \text{and} \qquad \Pi^{\mathsf{1}}_{\mathbf{x}} = \langle \mathsf{1}|u_{x0}, \mathsf{1}|u_{x1}, \mathsf{0}|u_{\bar{x}0}, \mathsf{0}|u_{\bar{x}1}\rangle$$

the *false-for-x* and the *true-for-x substitutions*, respectively. We call

$$\Delta = [t|f, f|t, u_{\bar{x}1}|u_{x0}, u_{\bar{x}0}|u_{x1}, u_{x1}|u_{\bar{x}0}, u_{x0}|u_{\bar{x}1}]$$

the *dualising substitution*. The notation $\widetilde{\boldsymbol{A}}$ denotes an rKDT formula $\boldsymbol{A}$ where $\vee$ is replaced by $\wedge$ and $\wedge$ is replaced by $\vee$.

Remark 7.3.2. An sFrege formula A is semantically equivalent to $\Sigma\lceil A \rceil^+$. The reason we use the four variables u_{x0}, u_{x1}, $u_{\bar{x}0}$ and $u_{\bar{x}1}$ for each sFrege variable x is to simulate sFrege unit substitution. For example, the corresponding open formula to the sFrege formula $A \equiv y \to (x \to x)$ is

$$\lceil A \rceil^+ \equiv (u_{\bar{y}1} \,\mathbf{y}\, u_{\bar{y}0}) \vee ((u_{\bar{x}1} \,\mathbf{x}\, u_{\bar{x}0}) \vee (u_{x0} \,\mathbf{x}\, u_{x1})) \quad ;$$

we simulate $[\mathsf{f}|x]\, A \equiv y \to (\mathsf{f} \to \mathsf{f})$ by

$$\mathsf{fl}\, \Pi^{\mathsf{0}}_{\mathbf{x}} \lceil A \rceil^+ \equiv (u_{\bar{y}1} \,\mathbf{y}\, u_{\bar{y}0}) \vee ((\mathsf{1} \,\mathbf{x}\, \mathsf{1}) \vee (\mathsf{0} \,\mathbf{x}\, \mathsf{0})) \quad .$$

Note how four rKDT variables u_{x0}, u_{x1}, $u_{\bar{x}0}$ and $u_{\bar{x}1}$ for each x are necessary for this to work.

Remark 7.3.3. For every sFrege formula A, we have $\lceil \neg A \rceil^+ \equiv \Delta \widetilde{\lceil A \rceil^+}$.

Theorem 7.3.4. *System* rKDT-ODSS *p-simulates* sFrege *and System* rKDT-ODS *p-simulates* Frege *in cubic time on the size of a given proof.*

Proof. Given an sFrege proof $A_1, \ldots, A_n$, we will build the following rKDT-ODSS proof:

$$\phi \equiv \Sigma\sigma_n \cdots \sigma_1 \left[\begin{array}{lll}
 & \langle \boldsymbol{B}_1|y_1\rangle & \langle \boldsymbol{B}_2|y_2\rangle \cdots \quad \langle \boldsymbol{B}_n|y_n\rangle\, y_n \\
\sim & & \\
 & \langle \psi_1|y_1\rangle & \langle \boldsymbol{B}_2|y_2\rangle \cdots \quad \langle \boldsymbol{B}_n|y_n\rangle\, y_n \\
\sim & & \\
\tau_1 & \langle \boldsymbol{K}_1\{\boldsymbol{A}_1\}|y_1\rangle & \langle \psi_2|y_2\rangle \cdots \quad \langle \boldsymbol{B}_n|y_n\rangle\, y_n \\
\sim & & \\
 & \vdots & \\
\sim & & \\
\tau_1 \cdots \tau_{n-2} & \langle \boldsymbol{K}_1\{\boldsymbol{A}_1\}|y_1\rangle\, \langle \boldsymbol{K}_2\{\boldsymbol{A}_2\}|y_2\rangle \cdots \quad \langle \boldsymbol{B}_n|y_n\rangle\, y_n \\
\sim & & \\
\tau_1 \cdots \tau_{n-2}\tau_{n-1} & \langle \boldsymbol{K}_1\{\boldsymbol{A}_1\}|y_1\rangle\, \langle \boldsymbol{K}_2\{\boldsymbol{A}_2\}|y_2\rangle \cdots \quad \langle \psi_n|y_n\rangle\, y_n \\
\sim & & \\
\tau_1 \cdots \tau_{n-2}\tau_{n-1}\tau_n & \langle \boldsymbol{K}_1\{\boldsymbol{A}_1\}|y_1\rangle\, \langle \boldsymbol{K}_2\{\boldsymbol{A}_2\}|y_2\rangle \cdots \langle \boldsymbol{K}_n\{\boldsymbol{A}_n\}|y_n\rangle\, y_n \\
\sim & & \\
 & \tau_1 \cdots \tau_n \boldsymbol{K}_n\{\boldsymbol{A}_n\} &
\end{array}\right],$$

in which for $1 \leq i \leq n$ and some actual substitution ρ^i, the following properties hold:

1. $\Sigma\, \langle \boldsymbol{B}_1|y_1\rangle \cdots \langle \boldsymbol{B}_i|y_i\rangle\, y_i = \mathsf{1}$

2. $\Sigma\tau_1 \cdots \tau_i \boldsymbol{K}_i\{\ \} = \{\ \}$

3. $\boldsymbol{A}_i \equiv [\rho^i v \vee \boldsymbol{Z}_v|v]_{\underline{\langle\!\langle A_i \rangle\!\rangle}} \langle\!\langle A_i \rangle\!\rangle$, where if where if $\pi v \notin \{f, t\}$, then $\boldsymbol{Z}_v \equiv \mathsf{0}$, otherwise $\boldsymbol{Z}_v \equiv \mathsf{0}$ or $\boldsymbol{Z}_v \equiv \mathsf{0}\, \mathbf{x}_v\, \mathsf{0}$, for some $\mathbf{x}_v$.

4. $\forall v \in \underline{\langle\!\langle A_i \rangle\!\rangle}$, either $\Big(\underline{\rho^i v} = \{\pi v\}$ and $\rho^i v = \pi v\Big)$, or $\Big(\underline{\rho^i v} = \emptyset$, $\pi v \in \{f, t\}$ and $\Sigma\rho^i v = \Sigma\pi v\Big)$

5. σ_i does not apply to $\boldsymbol{K}_1\{\boldsymbol{A}_1\}$, …, $\boldsymbol{K}_n\{\boldsymbol{A}_n\}$

6. $\tau_i = \langle \boldsymbol{C}_1 | z_1^i, \ldots, \boldsymbol{C}_{d_i} | z_{d_i}^i \rangle$ and $\boldsymbol{K}_i\{\ \} \equiv (z_1^i \vee \cdots (z_{d_i}^i \vee \{\ \}) \cdots)$, where the z_j^is are fresh and distinct and no y_i appears in any $\boldsymbol{C}_j$

7. $\forall x \in \mathsf{VF}$, all instances of $u_{x\mathsf{0}}$, $u_{x\mathsf{1}}$, $u_{\bar{x}\mathsf{0}}$ and $u_{\bar{x}\mathsf{1}}$ in ϕ are in the scope of $\mathbf{x}$, and every subformula $\boldsymbol{C} \equiv \boldsymbol{D}\,\mathbf{x}\,\boldsymbol{E}$ of $\boldsymbol{B}_i$ or $\tau_i\boldsymbol{K}_i\{\ \}$ is such that $\Sigma\boldsymbol{C} = \Sigma\Pi^{\mathsf{0}}_{\mathbf{x}}\boldsymbol{C} = \Sigma\Pi^{\mathsf{1}}_{\mathbf{x}}\boldsymbol{C}$

8. $\sigma_i = \begin{cases} \Pi^{\mathsf{0}}_{\mathbf{x}} \text{ restricted to range } i & \text{if } \psi_i \text{ simulates } \mathsf{usf} \text{ for } x \\ \Pi^{\mathsf{1}}_{\mathbf{x}} \text{ restricted to range } i & \text{if } \psi_i \text{ simulates } \mathsf{ust} \text{ for } x \\ \epsilon & \text{otherwise} \end{cases}$

The flat derivations $\psi_1, \ldots, \psi_n$ correspond each to an instance of an axiom or an application of mp, usf or ust; their role is to make available to the rest of the proof, via the fresh and distinct variables $y_1, \ldots, y_n$, the flat formulae $\boldsymbol{A}_1$, …, $\boldsymbol{A}_n$, which simulate $A_1, \ldots, A_n$. Σ is the standard value substitution. Each substitution σ_i is empty except when it contributes to simulate an sFrege unit substitution rule; in that case, σ_i is an explicit supersubstitution $\Pi^{\mathsf{0}}_{\mathbf{x}}$ or $\Pi^{\mathsf{1}}_{\mathbf{x}}$, for some x, whose substitution variables are restricted to the range i; the range i, informally, corresponds to the i-th 'column' in ϕ; the details are in the part of this proof about simulating substitution. Note that ϕ only uses supersubstitutions if the given proof uses substitutions. We note that what is denoted in the construction as composition by expansion may in fact abbreviate up to three instances of composition by expansion, in the case where A_i and A_{i+1} are both obtained by modus ponens, but this does not affect the order of the height of the construction.

If ψ_i simulates an axiom, the flat context $\boldsymbol{K}_i\{\ \}$ is identical to $\{\ \}$; otherwise, if ψ_i simulates a modus ponens between A_h and $A_h \to A_k$, then $\boldsymbol{K}_i\{\ \}$ is a disjunction of $\mathsf{0}$-valued formulae that, intuitively, collect the variables involved in the rKDT realisation of a cut between A_h and $\neg A_h$. As a consequence, $\boldsymbol{K}_i\{\ \}$ might be different from $\{\ \}$ when ψ_i simulates a unit substitution. The τ_is are explicit substitutions that prevent the modus-ponens

contexts $\boldsymbol{K}_i\{\ \}$s from growing exponentially by factoring out certain formulae.

This is how the premise of ψ_i is related to the rest of the proof: if ψ_i simulates an axiom, then $\mathsf{pr}\,\psi_i \equiv \boldsymbol{B}_i$; if ψ_i simulates a modus ponens between A_h and A_k, then $\boldsymbol{B}_i$ is built around a conjunction of y_h and y_k, and $\mathsf{pr}\,\psi_i \equiv \mathsf{fl}\,\langle \boldsymbol{K}_h\{\boldsymbol{A}_h\}|y_h\rangle\,\langle \boldsymbol{K}_k\{\boldsymbol{A}_k\}|y_k\rangle\,\boldsymbol{B}_i$; if ψ_i simulates a unit substitution for A_h, then $\boldsymbol{B}_i \equiv y_h^{\{i\}}$ and $\mathsf{pr}\,\psi_i \equiv \mathsf{fl}\,\sigma_i\,\langle \boldsymbol{K}_h\{\boldsymbol{A}_h\}|y_h\rangle\,\boldsymbol{B}_i$. Concerning the conclusion of ψ_i, in every case we have $\mathsf{cn}\,\psi_i \equiv \mathsf{fl}\,\tau_i\boldsymbol{K}_i\{\boldsymbol{A}_i\}$.

We assume to have an actual renaming substitution π and, for every i, an open rKDT formula $\langle\!\langle A_i \rangle\!\rangle$ such that

$$\pi\langle\!\langle A_i \rangle\!\rangle \equiv \lceil A_i \rceil^+$$

and all the variables in $\langle\!\langle A_1 \rangle\!\rangle$, ..., $\langle\!\langle A_n \rangle\!\rangle$ are maximally renamed apart and fresh. Note that, for every variable v appearing in any of $\langle\!\langle A_1 \rangle\!\rangle$, ..., $\langle\!\langle A_n \rangle\!\rangle$ we have either $\Sigma\pi v = \mathsf{0}$ or $\Sigma\pi v = \mathsf{1}$. We use the $\langle\!\langle A_i \rangle\!\rangle$s to build the derivations simulating the axioms and modus ponens – renaming apart the variables allows us to treat different occurrences of the same variable differently.

In the following case analysis, we prove the properties given above; they help to understand the construction and guide the proof of its correctness. Properties 1, 2, 3 and 4 are proved by induction on n. Properties 5, 6, 7 and 8 are guaranteed by the construction. Those properties ensure that ϕ is a simulation of the given sFrege proof.

Property 1 guarantees that each ψ_i simulation could be flattened into a derivation having premise equal to $\mathsf{1}$.

Property 7 guarantees that the value of every subformula in the premise of ϕ that corresponds to an sFrege variable x remains unchanged by any false-for-x or true-for-x substitution; it guarantees the same also for the part of the conclusion of ϕ that collects the variables involved in cuts.

Property 8 guarantees that the only possible substitutions are of the false-for-x or true-for-x kind.

Together, Properties 1, 7 and 8 entail

$$\mathsf{pr}\,\phi \equiv \Sigma\sigma_n \cdots \sigma_1\,\langle \boldsymbol{B}_1|y_1\rangle \cdots \langle \boldsymbol{B}_n|y_n\rangle\,y_n = \mathsf{1} \quad ;$$

note that the order of the σ_is in $\sigma_n \cdots \sigma_1$ does not matter.

Property 2 guarantees that the modus-ponens context $\boldsymbol{K}_i$ and its related explicit substitution τ_i do not alter the value of $\boldsymbol{A}_i$.

Property 3 models the structure of each $\boldsymbol{A}_i$ formula in such a way that the simulations are strictly linear; Property 4 guarantees that the extra structure needed for that (which mostly goes into the actual substitution ρ^i) does not alter the value of $\boldsymbol{A}_i$.

Properties 5 and 6 are non-trivial only in the case of modus ponens and substitution and are further clarified below.

Properties 7, 5, 2, 6, 3 and 4 entail

$$\begin{aligned}\mathsf{cn}\,\phi &\equiv \Sigma\sigma_n\cdots\sigma_1\tau_1\cdots\tau_n\boldsymbol{K}_n\{\boldsymbol{A}_n\}\\ &= \Sigma\boldsymbol{A}_n\\ &\equiv \Sigma\,[\rho^n v\vee \boldsymbol{Z}_v|v]\,\underline{\langle\!\langle A_n\rangle\!\rangle}\langle\!\langle A_n\rangle\!\rangle\\ &= \Sigma\pi\langle\!\langle A_n\rangle\!\rangle\\ &\equiv \Sigma\lceil A_n\rceil^+\end{aligned}\quad,$$

which is semantically equivalent to A_n.

We now see in detail how each ψ_i is built, based on its nature.

Axioms. If A_i is an instance of an axiom F_1, F_2 or F_3, we build a flat derivation ω_i according to the scheme in Figure 7.3, 7.4 or 7.5, respectively. If A_i is an instance of F_4 or F_5, we set $\omega_i \equiv t$. In an effort to simplify the notation, in all the constructions for axioms, we assume that the formulae $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{E}$, with various decorations, are local to those constructions (*i.e.*, a $\boldsymbol{C}_1$ for one axiom is not the same as a $\boldsymbol{C}_1$ for another one). In every case, we let $\sigma_i = \tau_i = \epsilon$, therefore satisfying Properties 5 and 8.

We set $\boldsymbol{K}_i\{\ \} \equiv \{\ \}$ (therefore $d_i = 0$), which implies $\mathsf{cn}\,\psi_i \equiv \boldsymbol{A}_i$, so satisfying Properties 2 and 6.

We are given $\langle\!\langle A_i\rangle\!\rangle$, obtained from $\lceil A_i\rceil^+$ by maximally renaming fresh variables apart such that $\pi\langle\!\langle A_i\rangle\!\rangle \equiv \lceil A_i\rceil^+$. $\langle\!\langle A_i\rangle\!\rangle$ contains subformulae that might be merged inside ω_i to build its premise and conclusion. For example, for F_1 in Figure 7.3, $\langle\!\langle A_i\rangle\!\rangle$ is decomposed into $\boldsymbol{C}_1$, $\boldsymbol{C}_2$ and $\boldsymbol{D}_1$ and these three formulae determine the formulae $\boldsymbol{C}_1^1$, $\boldsymbol{C}_3$, $\boldsymbol{C}_4$ and $\boldsymbol{D}_2$; the derivations χ_j^is are obtained from the Merge Lemma 5.1.10 or its variant 5.1.17. The derivation ψ_i is obtained from ω_i via a merge construction by Lemma 5.1.17:

$$\psi_i \equiv \boxed{\begin{array}{c}\omega_i \vee [\mathsf{0}|v]_{\underline{\langle\!\langle A_i\rangle\!\rangle}}\langle\!\langle A_i\rangle\!\rangle\\ \|\\ [\rho^i v\vee \mathsf{0}|v]_{\underline{\langle\!\langle A_i\rangle\!\rangle}}\langle\!\langle A_i\rangle\!\rangle\end{array}}\quad.$$

This construction relies on the identity

$$\mathsf{cn}\,\omega_i \equiv \rho^i\langle\!\langle A_i \rangle\!\rangle \quad , \tag{7.1}$$

where ρ^i is some actual substitution; we build ρ^i and verify the above identity for each axiom in the following. Identity 7.1 immediately yields Property 3 in the case of axioms. Note that $\mathsf{cn}\,\psi_i$ is semantically equivalent to $\mathsf{cn}\,\omega_i$ – it just contains some 'padding' with $\mathsf{0}$s. We do this construction to extract those $\mathsf{0}$s in the modus ponens simulation, as required by strict linearity.

For every axiom instance we set

$$\boldsymbol{B}_i \equiv \mathsf{pr}\,\omega_i \vee [\mathsf{0}|v]_{\underline{\langle\!\langle A_i \rangle\!\rangle}}\langle\!\langle A_i \rangle\!\rangle \equiv \mathsf{pr}\,\psi_i \quad .$$

In the case of axioms, Properties 1, 2, 3 and 4 do not rely on the induction hypothesis. Since ϕ_1 must simulate an axiom, establishing those properties for axioms serves as the base case for each of them. Having established Properties 2, 5, 6 and 8, in the following we only must verify Properties 1, 3, 4 and 7 for each possible axiom.

Axiom F_1. We simulate $A_i \equiv C \to (D \to C)$ and we refer to Figure 7.3. The substitution into $\boldsymbol{D}_1$ realises the weakening of formula D by weakening each unit that appears in $\mathsf{fl}\,\Sigma\pi\boldsymbol{D}_1$. The derivation χ_1^i realises the identity $C \to C$; note that $\boldsymbol{C}_1$ and $\widetilde{\boldsymbol{C}}_1$ are down-conjugates of $\widehat{\boldsymbol{C}}_1$, therefore we can use the Merge Lemma 5.1.10 with some renaming substitution ρ that depends on the renaming substitution π. The derivation χ_2^i helps to maintain the strict linearity of ω_i by merging $\boldsymbol{C}_1$ with an up-conjugate of itself set to be equal to $\mathsf{1}$.

We verify Property 1:

$$\begin{aligned}
\Sigma\,\langle \boldsymbol{B}_1|y_1\rangle \cdots \langle \boldsymbol{B}_i|y_i\rangle\, y_i &= \Sigma\,\langle \boldsymbol{B}_i|y_i\rangle\, y_i \\
&= \Sigma\pi\left(\left(\left(\boldsymbol{C}_1^1 \vee [(\mathsf{0}\wedge v)\vee(v\wedge\mathsf{0})|v]_{\underline{\boldsymbol{D}_1}}\boldsymbol{D}_1\right)\wedge\boldsymbol{C}_3\right)\vee\ [\mathsf{0}|v]_{\underline{\langle\!\langle A_i \rangle\!\rangle}}\langle\!\langle A_i \rangle\!\rangle\right) \\
&= \Sigma\pi[v\vee\rho v|v]_{\underline{\boldsymbol{C}_1}}\widehat{\boldsymbol{C}}_1 = [\Sigma\pi v\vee\Sigma\Delta\pi v|v]_{\underline{\boldsymbol{C}_1}}\widehat{\boldsymbol{C}}_1 = \mathsf{1} \quad ,
\end{aligned}$$

because $y_1, \ldots, y_{i-1}$ do not appear in $\boldsymbol{B}_i$, and because $\Sigma\pi v$ and $\Sigma\Delta\pi v$ have dual values (see Remark 7.3.3).

We verify Property 3 by verifying Identity 7.1 and at the same time assigning a value to ρ^i:

$$\begin{aligned}
\mathsf{cn}\,\omega_i &\equiv \pi(\boldsymbol{C}_4 \vee (\boldsymbol{D}_2 \vee \boldsymbol{C}_2)) \\
&\equiv \pi\left([\mathsf{1}\wedge v|v]_{\underline{\boldsymbol{C}_1}}\boldsymbol{C}_1 \vee \left([(\mathsf{0}\vee v)\wedge(v\vee\mathsf{0})|v]_{\underline{\boldsymbol{D}_1}}\boldsymbol{D}_1 \vee \boldsymbol{C}_2\right)\right) \\
&\equiv \pi[\mathsf{1}\wedge v|v]_{\underline{\boldsymbol{C}_1}}[(\mathsf{0}\vee v)\wedge(v\vee\mathsf{0})|v]_{\underline{\boldsymbol{D}_1}}\langle\!\langle A_i \rangle\!\rangle \equiv \rho^i\langle\!\langle A_i \rangle\!\rangle \quad ;
\end{aligned}$$

$$\omega_i \equiv \pi \boxed{\vee\check{\wedge}\ \frac{\left(\boldsymbol{C}_1^{\mathsf{1}} \vee \left[\wedge\check{\vee}\ \frac{(\mathsf{0}\wedge v)\vee(v\wedge\mathsf{0})}{(\mathsf{0}\vee v)\wedge(v\vee\mathsf{0})}\middle| v\right]_{\underline{\boldsymbol{D}_1}} \boldsymbol{D}_1\right) \wedge \boxed{\begin{array}{c}\boldsymbol{C}_3\\ \chi_1^i \,\|\\ \boldsymbol{C}_1\vee\boldsymbol{C}_2\end{array}}}{\boxed{\begin{array}{c}\boldsymbol{C}_1^{\mathsf{1}}\wedge\boldsymbol{C}_1\\ \chi_2^i\,\|\\ \boldsymbol{C}_4\end{array}} \vee (\boldsymbol{D}_2\vee\boldsymbol{C}_2)}}$$

$$\begin{aligned}
\langle\!\langle A_i \rangle\!\rangle &\equiv \boldsymbol{C}_1 \vee (\boldsymbol{D}_1 \vee \boldsymbol{C}_2)\\
\boldsymbol{C}_2 &\equiv \rho\widetilde{\boldsymbol{C}}_1\\
\boldsymbol{C}_1^{\mathsf{1}} &\equiv [\mathsf{1}|v]_{\underline{\boldsymbol{C}_1}}\widehat{\boldsymbol{C}}_1\\
\boldsymbol{C}_3 &\equiv [v\vee\rho v|v]_{\underline{\boldsymbol{C}_1}}\widehat{\boldsymbol{C}}_1\\
\boldsymbol{C}_4 &\equiv [\mathsf{1}\wedge v|v]_{\underline{\boldsymbol{C}_1}}\boldsymbol{C}_1\\
\boldsymbol{D}_2 &\equiv [(\mathsf{0}\vee v)\wedge(v\vee\mathsf{0})|v]_{\underline{\boldsymbol{D}_1}}\boldsymbol{D}_1
\end{aligned}$$

Figure 7.3: Simulation of F_1 instance $A_i \equiv C \to (D \to C)$ in Theorem 7.3.4.

we exploit the fact that the variables of $\langle\!\langle A_i \rangle\!\rangle$ are maximally renamed apart.

We then use ρ^i to verify Property 4, given that $\underline{\rho^i v} = \{\pi v\}$ for $v \in \underline{\langle\!\langle A_i \rangle\!\rangle}$:

$$\pi[1 \wedge v|v]_{\underline{\boldsymbol{C}_1}}[(0 \vee v) \wedge (v \vee 0)|v]_{\underline{\boldsymbol{D}_1}} v = \pi v \quad .$$

Property 7 holds because, for all $x \in \mathsf{VF}$, all instances of u_{x0}, u_{x1}, $u_{\bar{x}0}$ and $u_{\bar{x}1}$ in ϕ are in the scope of $\mathbf{x}$ in $\pi\langle\!\langle A \rangle\!\rangle_i$ and this fact is not changed by any of the substitutions and inference steps in ω_i. Moreover, the value of $\Sigma\pi\boldsymbol{C}_1^1$ and $\Sigma\pi[(0 \wedge v) \vee (v \wedge 0)|v]_{\underline{\boldsymbol{D}_1}}\boldsymbol{D}_1$ is not affected by any substitution. Finally, concerning the subformulae of $\pi\boldsymbol{C}_3$, for $v \in \underline{\boldsymbol{C}_1}$, if $\pi v \equiv u_{x0}$ then $\Delta\pi v \equiv u_{\bar{x}1}$, which means $\Pi^0_{\mathbf{x}}\pi v \vee \Pi^0_{\mathbf{x}}\Delta\pi v = 0 \vee 1 = 1 = 1 \vee 0 = \Pi^1_{\mathbf{x}}\pi v \vee \Pi^1_{\mathbf{x}}\Delta\pi v$, and similarly if $\pi v \equiv u_{x1}$, $\pi v \equiv u_{\bar{x}1}$ and $\pi v \equiv u_{\bar{x}0}$.

Axiom F_2. We simulate $A_i \equiv (C \to (D \to E)) \to ((C \to D) \to (C \to E))$ and we refer to Figure 7.4. The construction and the way we prove its properties are similar to those for Axiom F_1. There is no weakening here, but there is a contraction on C, realised by derivation χ^i_9. The derivations χ^i_1, χ^i_2, χ^i_3 and χ^i_4 realise the identities $(\neg C \to \neg C)$, $(\neg C \to \neg C)$, $(\neg D \to \neg D)$ and $(E \to E)$, respectively. The derivations χ^i_5, χ^i_6, χ^i_7, χ^i_8 and χ^i_{10} help to maintain the strict linearity of ω_i. The substitution ρ^i can be computed by composing the substitutions to be applied to $\langle\!\langle A_i \rangle\!\rangle$ to get $\mathsf{cn}\,\omega_i$, *i.e.*,

$$\rho^i = \pi[1 \wedge (0 \vee v)|v]_{\underline{\boldsymbol{C}_1}}[0 \vee v|v]_{\underline{(\boldsymbol{D}_1 \wedge \boldsymbol{E}_1)}}[v \wedge 1|v]_{\underline{\boldsymbol{C}_2}}[0 \vee v|v]_{\underline{\boldsymbol{D}_2}}[v \vee v|v]_{\underline{\boldsymbol{C}_3}}[0 \vee v|v]_{\underline{\boldsymbol{E}_2}} \quad .$$

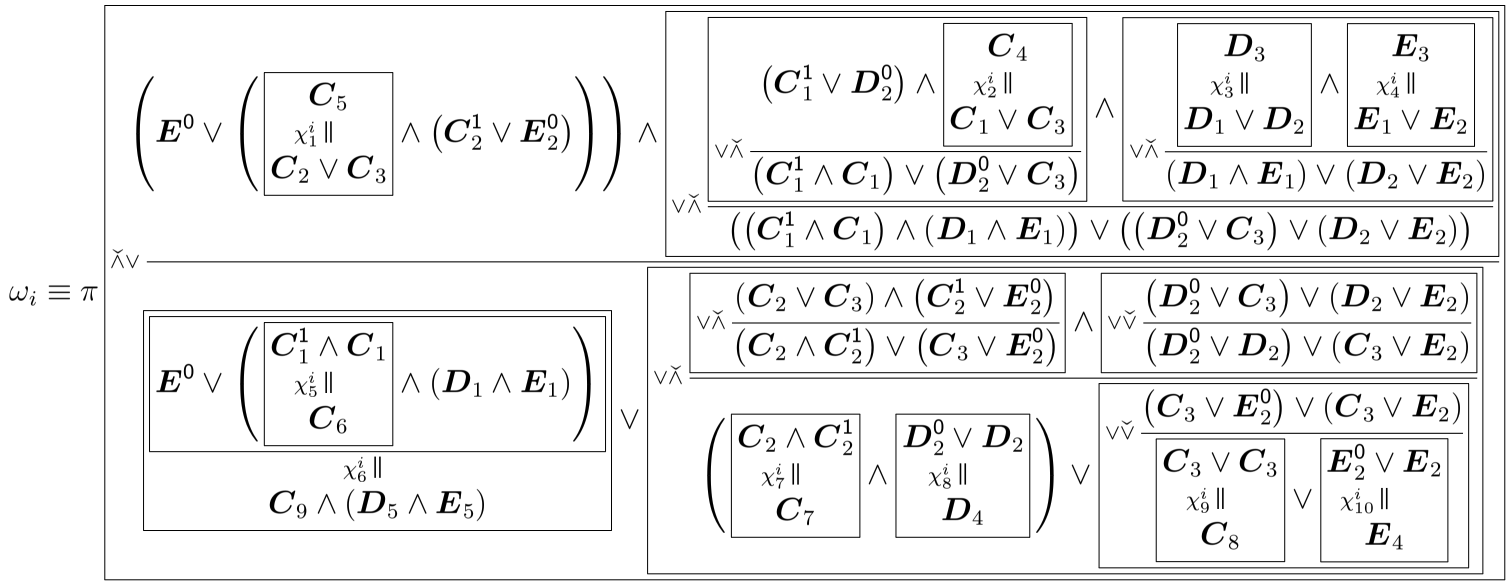
Verifying Properties 1, 3, 4 and 7 is similar to what we did for Axiom F_1.

Axiom F_3. We simulate $(\neg C \to \neg D) \to (D \to C)$ and we refer to Figure 7.5. The construction and the way we prove its properties are similar to those for Axioms F_1 and F_2. There is no weakening and no contraction here. The derivations χ^i_1 and χ^i_2 realise the identities $C \to C$ and $\neg D \to \neg D$, respectively. The derivations χ^i_3, χ^i_4, χ^i_5 and χ^i_6 help to maintain the strict linearity of ω_i. The substitution ρ^i can be computed by composing the substitutions to be applied to $\langle\!\langle A_i \rangle\!\rangle$ to get $\mathsf{cn}\,\omega_i$, and verifying Properties 1, 3, 4 and 7 is similar to what we did for Axioms F_1 and F_2.

Axioms F_4 ***and*** F_5. Property 1 is immediate. Properties 3 and 4 follow from $\langle\!\langle A_i \rangle\!\rangle \equiv v$, for some variable v, such that $\pi v \equiv t$ and $\boldsymbol{A}_i \equiv [t \vee 0|v]_{\underline{\langle\!\langle A_i \rangle\!\rangle}}\langle\!\langle A_i \rangle\!\rangle \equiv t \vee 0$. Property 7 is trivial because ψ_i contains no atom connectives.

Modus Ponens. We simulate

$$\mathsf{mp}\,\frac{A_h \quad A_h \to A_i}{A_i}$$



$$\langle\!\langle A_i \rangle\!\rangle \equiv (\boldsymbol{C}_1 \wedge (\boldsymbol{D}_1 \wedge \boldsymbol{E}_1)) \vee ((\boldsymbol{C}_2 \wedge \boldsymbol{D}_2) \vee (\boldsymbol{C}_3 \vee \boldsymbol{E}_2))$$

$$\boldsymbol{C}_1 \equiv \rho_1 \widetilde{\boldsymbol{C}}_3 \qquad \boldsymbol{D}_1 \equiv \rho_3 \widetilde{\boldsymbol{D}}_2$$

$$\boldsymbol{C}_2 \equiv \rho_2 \widetilde{\boldsymbol{C}}_3 \qquad \boldsymbol{E}_2 \equiv \rho_4 \widetilde{\boldsymbol{E}}_1$$

$$\boldsymbol{C}_1^1 \equiv [1|v]_{\underline{\boldsymbol{C}_1}} \widehat{\boldsymbol{C}}_1 \qquad \boldsymbol{D}_2^0 \equiv [0|v]_{\underline{\boldsymbol{D}_2}} \boldsymbol{D}_2$$

$$\boldsymbol{C}_2^1 \equiv [1|v]_{\underline{\boldsymbol{C}_2}} \widehat{\boldsymbol{C}}_2 \qquad \boldsymbol{E}_2^0 \equiv [0|v]_{\underline{\boldsymbol{E}_2}} \boldsymbol{E}_2$$

$$\boldsymbol{C}_4 \equiv [\rho_1 v \vee v|v]_{\underline{\boldsymbol{C}_3}} \widehat{\boldsymbol{C}}_3 \qquad \boldsymbol{D}_3 \equiv [\rho_3 v \vee v|v]_{\underline{\boldsymbol{D}_2}} \widehat{\boldsymbol{D}}_2$$

$$\boldsymbol{C}_5 \equiv [\rho_2 v \vee v|v]_{\underline{\boldsymbol{C}_3}} \widehat{\boldsymbol{C}}_3 \qquad \boldsymbol{E}_3 \equiv [v \vee \rho_4 v|v]_{\underline{\boldsymbol{E}_1}} \widehat{\boldsymbol{E}}_1$$

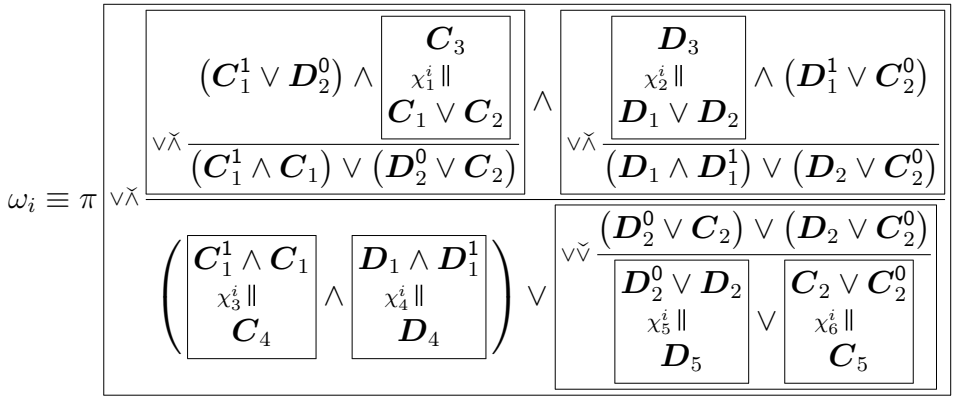
$$\boldsymbol{C}_6 \equiv [1 \wedge v|v]_{\underline{\boldsymbol{C}_1}} \boldsymbol{C}_1 \qquad \boldsymbol{D}_4 \equiv [0 \vee v|v]_{\underline{\boldsymbol{D}_2}} \boldsymbol{D}_2$$

$$\boldsymbol{C}_7 \equiv [v \wedge 1|v]_{\underline{\boldsymbol{C}_2}} \boldsymbol{C}_2 \qquad \boldsymbol{E}_4 \equiv [0 \vee v|v]_{\underline{\boldsymbol{E}_2}} \boldsymbol{E}_2$$

$$\boldsymbol{C}_8 \equiv [v \vee v|v]_{\underline{\boldsymbol{C}_3}} \boldsymbol{C}_3 \qquad \boldsymbol{E}^0 \equiv [0|v]_{\underline{\boldsymbol{C}_6 \wedge (\boldsymbol{D}_1 \wedge \boldsymbol{E}_1)}} (\boldsymbol{C}_6 \wedge (\boldsymbol{D}_1 \wedge \boldsymbol{E}_1))$$

$$\boldsymbol{C}_9 \wedge (\boldsymbol{D}_5 \wedge \boldsymbol{E}_5) \equiv [0 \vee v|v]_{\underline{\boldsymbol{C}_6 \wedge (\boldsymbol{D}_1 \wedge \boldsymbol{E}_1)}} (\boldsymbol{C}_6 \wedge (\boldsymbol{D}_1 \wedge \boldsymbol{E}_1))$$

Figure 7.4: Simulation of F_2 instance $A_i \equiv (C \to (D \to E)) \to ((C \to D) \to (C \to E))$ in Theorem 7.3.4.



$$\langle\!\langle A_i \rangle\!\rangle \equiv (\boldsymbol{C}_1 \wedge \boldsymbol{D}_1) \vee (\boldsymbol{D}_2 \vee \boldsymbol{C}_2)$$

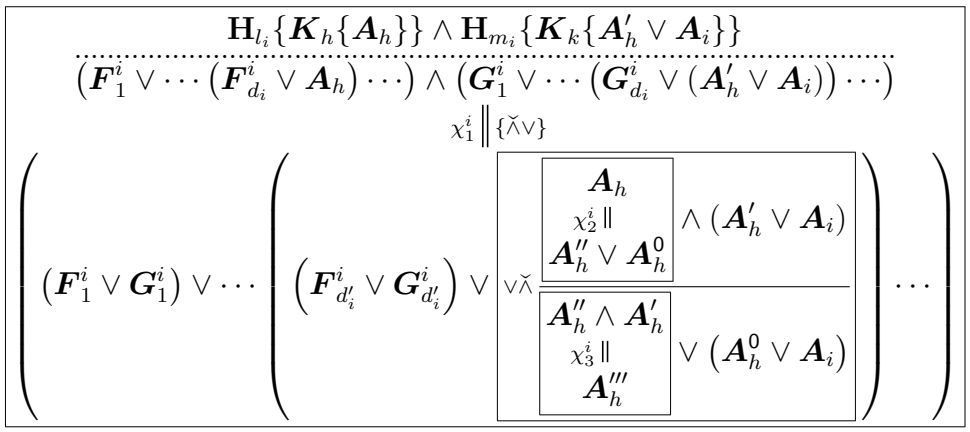
$$\boldsymbol{C}_2 \equiv \rho_1 \widetilde{\boldsymbol{C}}_1 \qquad \boldsymbol{C}_2^0 \equiv [0|v]_{\underline{\boldsymbol{C}_2}} \boldsymbol{C}_2 \qquad \boldsymbol{C}_4 \equiv [1 \wedge v|v]_{\underline{\boldsymbol{C}_1}} \boldsymbol{C}_1$$

$$\boldsymbol{D}_1 \equiv \rho_2 \widetilde{\boldsymbol{D}}_2 \qquad \boldsymbol{D}_2^0 \equiv [0|v]_{\underline{\boldsymbol{D}_2}} \boldsymbol{D}_2 \qquad \boldsymbol{D}_4 \equiv [v \wedge 1|v]_{\underline{\boldsymbol{D}_1}} \boldsymbol{D}_1$$

$$\boldsymbol{C}_1^1 \equiv [1|v]_{\underline{\boldsymbol{C}_1}} \widehat{\boldsymbol{C}}_1 \qquad \boldsymbol{C}_3 \equiv [v \vee \rho_1 v|v]_{\underline{\boldsymbol{C}_1}} \widehat{\boldsymbol{C}}_1 \qquad \boldsymbol{C}_5 \equiv [v \vee 0|v]_{\underline{\boldsymbol{C}_2}} \boldsymbol{C}_2$$

$$\boldsymbol{D}_1^1 \equiv [1|v]_{\underline{\boldsymbol{D}_1}} \widehat{\boldsymbol{D}}_1 \qquad \boldsymbol{D}_3 \equiv [\rho_2 v \vee v|v]_{\underline{\boldsymbol{D}_2}} \widehat{\boldsymbol{D}}_2 \qquad \boldsymbol{D}_5 \equiv [0 \vee v|v]_{\underline{\boldsymbol{D}_2}} \boldsymbol{D}_2$$

Figure 7.5: Simulation of F_3 instance $A_i \equiv (\neg C \to \neg D) \to (D \to C)$ in Theorem 7.3.4.



$$\sigma_i = \epsilon \qquad \boldsymbol{A}_h \equiv \big[\rho^h v \vee \boldsymbol{Z}_v \big| v\big]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \langle\!\langle A_h \rangle\!\rangle$$

$$d'_i = \max(d_h, d_k) = d_i - 2 \qquad \boldsymbol{A}_k \equiv \big[\rho^k v \vee \boldsymbol{Z}_v \big| v\big]_{\underline{\langle\!\langle A_k \rangle\!\rangle}} \langle\!\langle A_k \rangle\!\rangle \equiv \boldsymbol{A}'_h \vee \boldsymbol{A}_i$$

$$l_i = d'_i - d_h \qquad \boldsymbol{A}'_h \equiv \rho_k \widetilde{\langle\!\langle A_h \rangle\!\rangle}$$

$$m_i = d'_i - d_k \qquad \boldsymbol{A}''_h \equiv \rho^h \langle\!\langle A_h \rangle\!\rangle$$

$$\mathbf{H}_j\{\ \} \equiv \underbrace{(\mathsf{0} \vee \cdots (\mathsf{0} \vee}_{j} \{\ \}) \cdots) \qquad \boldsymbol{A}^{\mathsf{0}}_h \equiv [\mathsf{0}|v]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \widetilde{\langle\!\langle A_h \rangle\!\rangle}$$

$$\boldsymbol{A}'''_h \equiv \big[\rho^h v \wedge \rho_k v \big| v\big]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \widetilde{\langle\!\langle A_h \rangle\!\rangle}$$

$$\boldsymbol{B}_i \equiv \mathbf{H}_{l_i}\{y_h\} \wedge \mathbf{H}_{m_i}\{y_k\} \qquad \boldsymbol{K}_i\{\ \} \equiv (z^i_1 \vee \cdots (z^i_{d_i} \vee \{\ \}) \cdots)$$

$$\tau_i = \big\langle \boldsymbol{F}^i_j \vee \boldsymbol{G}^i_j \big| z^i_j \big\rangle_{1 \le j \le d'_i} \Big\langle \boldsymbol{A}'''_h \Big| z^i_{d'_i+1}, \boldsymbol{A}^{\mathsf{0}}_h \Big| z^i_{d'_i+2} \Big\rangle$$

Figure 7.6: Simulation of modus ponens instance in Theorem 7.3.4.

and we refer to Figure 7.6. We set $\sigma_i = \epsilon$ and $\boldsymbol{B}_i \equiv \mathbf{H}_{l_i}\{y_h\} \wedge \mathbf{H}_{m_i}\{y_k\}$, where the contexts $\mathbf{H}_{l_i}\{\ \}$ and $\mathbf{H}_{m_i}\{\ \}$ (one of which is identical to $\{\ \}$) make sure that the two conjuncts in $\boldsymbol{B}_i$ have disjunctions of 0-valued formulae of equal 'depth', *i.e.*, $l_i + d_h = m_i + d_k = d'_i$. The 'depth' of context $\boldsymbol{K}_i\{\ \}$ is $d_i = d'_i + 2$. The formulae $\boldsymbol{F}^i_1, \ldots \boldsymbol{F}^i_{d_i}$ and $\boldsymbol{G}^i_1, \ldots \boldsymbol{G}^i_{d_i}$ are either $\mathsf{0}$ or some variable 'z' from $\boldsymbol{K}_h\{\ \}$ or $\boldsymbol{K}_k\{\ \}$. We are given $\boldsymbol{A}_h$ and $\boldsymbol{A}_k$. We assume that $\boldsymbol{A}_k \equiv \boldsymbol{A}'_h \vee \boldsymbol{A}_i$ simulates $A_k \equiv A_h \to A_i$. From $\boldsymbol{A}_k$ we extract $\boldsymbol{A}'_h \equiv \rho_k \widetilde{\langle\!\langle A_h \rangle\!\rangle}$ for some substitution ρ_k obtained from ρ^k. We build the derivation χ^i_1 by repeatedly applying $\bar{\wedge}\vee$. χ^i_2 and χ^i_3 are two merge derivations for the formulae defined in the figure. The ability to perform χ^i_2 is why we 'pad' with $\mathsf{0}$s the conclusion of ω_h in the simulations of axioms.

Property 1, *i.e.*, $\Sigma\, \langle \boldsymbol{B}_1 | y_1 \rangle \cdots \langle \boldsymbol{B}_{i-1} | y_{i-1} \rangle \, \langle \mathbf{H}_{l_i}\{y_h\} \wedge \mathbf{H}_{m_i}\{y_k\} | y_i \rangle \, y_i = \mathsf{1}$, follows directly from the induction hypothesis.

Property 2, *i.e.*, $\Sigma\tau_1 \cdots \tau_i \boldsymbol{K}_i\{\ \} = \{\ \}$, follows from the following facts:

1. By the induction hypothesis, $\Sigma\tau_1 \cdots \tau_h \boldsymbol{K}_h\{\ \} = \Sigma\tau_1 \cdots \tau_k \boldsymbol{K}_k\{\ \} = \{\ \}$, which implies

$$\Sigma\tau_1 \cdots \tau_h \boldsymbol{F}^i_j = \Sigma\tau_1 \cdots \tau_k \boldsymbol{G}^i_j = \mathsf{0} \quad , \qquad \text{for } 1 \leq j \leq d'_i.$$

2. Since

   - by Property 6, τ_1, …, τ_i do not apply to $\boldsymbol{A}'''_h$ because it only contains variables from $\boldsymbol{A}_h$ and $\boldsymbol{A}_k$,
   - by Property 4, for $v \in \underline{\langle\!\langle A_h \rangle\!\rangle}$, we have $\Sigma\rho^h v = \Sigma\pi v$,
   - $\boldsymbol{A}'_h \equiv \rho_k \widetilde{\langle\!\langle A_h \rangle\!\rangle} \equiv \big[\rho^k \rho'_k v \vee \boldsymbol{Z}_v \big| v\big]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \widetilde{\langle\!\langle A_h \rangle\!\rangle}$, for some renaming substitution ρ'_k, and then, for $v \in \underline{\langle\!\langle A_h \rangle\!\rangle}$ and by Property 4, we have $\Sigma\rho_k v = \Sigma\rho^k\rho'_k v = \Sigma\pi\rho'_k v = \Sigma\Delta\pi v$, whose value is dual to that of $\Sigma\pi v$ (see Remark 7.3.3),

   we obtain

$$\Sigma\tau_1 \cdots \tau_i \boldsymbol{A}'''_h = \Sigma\big[\rho^h v \wedge \rho_k v \big| v\big]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \widetilde{\langle\!\langle A_h \rangle\!\rangle} = [\Sigma\pi v \wedge \Sigma\Delta\pi v | v]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \widetilde{\langle\!\langle A_h \rangle\!\rangle} = \mathsf{0} \quad .$$

3. $\Sigma\tau_1 \cdots \tau_i \boldsymbol{A}^{\mathsf{0}}_h = \mathsf{0}$.

Property 3 is proved by induction by taking $\rho^i = \left[\rho^k \rho v \middle| v\right]_{\underline{\langle\!\langle A_i \rangle\!\rangle}}$, for some renaming substitution ρ that maps $\underline{\langle\!\langle A_i \rangle\!\rangle}$ into $\underline{\langle\!\langle A_k \rangle\!\rangle}$, such that

$$\boldsymbol{A}_i \equiv \left[\rho^k \rho v \vee \boldsymbol{Z}_{\rho v} \middle| v\right]_{\underline{\langle\!\langle A_i \rangle\!\rangle}} \langle\!\langle A_i \rangle\!\rangle \qquad \text{and} \qquad \pi\rho\langle\!\langle A_i \rangle\!\rangle \equiv \pi\langle\!\langle A_i \rangle\!\rangle \equiv \lceil A_i \rceil^+ \quad ;$$

note that $\pi v \equiv \pi\rho v$ and $\boldsymbol{Z}_v \equiv \boldsymbol{Z}_{\rho v}$, for all $v \in \underline{\boldsymbol{A}_i}$.

Property 4 holds because, for all $v \in \underline{\langle\!\langle A_i \rangle\!\rangle}$ and by the induction hypothesis, we have $\underline{\rho^i v} = \underline{\rho^k \rho v}$ and $\pi v \equiv \pi\rho v$, and

$$\rho^i v \equiv \rho^k \rho v = \pi\rho v \equiv \pi v \qquad \text{or} \qquad \Sigma\rho^i v \equiv \Sigma\rho^k \rho v = \Sigma\pi\rho v \equiv \Sigma\pi v \quad ,$$

according to whether $\underline{\rho^i v}$ is nonempty or empty.

Property 7 trivially holds for $\boldsymbol{B}_i$. Regarding $\tau_i \boldsymbol{K}_i\{\ \}$, the only non-trivial check concerns $\boldsymbol{A}'''_h$, for which we have (see Case 2 above):

$$\Sigma \boldsymbol{A}'''_h = \Sigma[\pi v \wedge \Delta\pi v | v]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \widetilde{\langle\!\langle A_h \rangle\!\rangle} = \mathsf{0} \quad .$$

From $\Sigma(u_{x\mathsf{0}} \wedge \Delta u_{x\mathsf{0}}) = \mathsf{0}$ and

$$\Pi^{\mathsf{0}}_{\mathbf{x}}(u_{x\mathsf{0}} \wedge \Delta u_{x\mathsf{0}}) \equiv \Pi^{\mathsf{0}}_{\mathbf{x}}(u_{x\mathsf{0}} \wedge u_{\bar{x}\mathsf{1}}) = \mathsf{0} \wedge \mathsf{1} = \mathsf{0} = \mathsf{1} \wedge \mathsf{0} = \Pi^{\mathsf{1}}_{\mathbf{x}}(u_{x\mathsf{0}} \wedge u_{\bar{x}\mathsf{1}}) \equiv \Pi^{\mathsf{1}}_{\mathbf{x}}(u_{x\mathsf{0}} \wedge \Delta u_{x\mathsf{0}}) \quad ,$$

and from the similar cases of $u_{x\mathsf{1}}$, $u_{\bar{x}\mathsf{0}}$ and $u_{\bar{x}\mathsf{1}}$, we conclude that Property 7 holds.

Properties 5, 6 and 8 are true by construction.

Unit substitution. We only show the simulation of

$$\mathsf{usf}\,\frac{A_h}{[\mathsf{f}|x]\,A_h} \quad ,$$

the case of ust being similar.

We set

$$\sigma_i = \left\langle \mathsf{0} \middle| u^{\{i\}}_{x\mathsf{0}}, \mathsf{0} \middle| u^{\{i\}}_{x\mathsf{1}}, \mathsf{1} \middle| u^{\{i\}}_{\bar{x}\mathsf{0}}, \mathsf{1} \middle| u^{\{i\}}_{\bar{x}\mathsf{1}} \right\rangle \qquad \text{and} \qquad \boldsymbol{B}_i \equiv y^{\{i\}}_h \quad .$$

Let ρ be an actual substitution such that $\langle\!\langle A_h \rangle\!\rangle \equiv \rho\langle\!\langle A_i \rangle\!\rangle \equiv \rho\langle\!\langle [\mathsf{f}|x]\,A_h \rangle\!\rangle$. ρ is a renaming substitution for all variables except for those in the set $X \subseteq \underline{\langle\!\langle A_i \rangle\!\rangle}$ that collects the (renamed apart) results of instances of x that got substituted, and which is so defined: $v \in X$ if and only if $\rho v \equiv v^{\mathsf{L}}\ \mathbf{x}\ v^{\mathsf{R}}$, for some variables v^{L} and v^{R} that depend on v. Note that if $v \in X$ then either

$\pi v \equiv f$ and $\pi\rho v \equiv u_{x0}\,\mathbf{x}\,u_{x1}$, or $\pi v \equiv t$ and $\pi\rho v \equiv u_{\bar{x}1}\,\mathbf{x}\,u_{\bar{x}0}$. Given ρ^h, we get

$$\boldsymbol{A}_h \equiv \left[\rho^h v \vee \boldsymbol{Z}_v \middle| v\right]_{\underline{\langle\!\langle A_h \rangle\!\rangle}} \langle\!\langle A_h \rangle\!\rangle \equiv \left[\rho^h \rho v \vee \boldsymbol{Z}_{\rho v} \middle| v\right]_{\underline{\langle\!\langle A_i \rangle\!\rangle}} \langle\!\langle A_i \rangle\!\rangle \quad .$$

The simulation is performed by

$$\psi_i \equiv \mathsf{fl}\left[\boldsymbol{K}_h\left\{\left[\Pi^0_{\mathbf{x}}\boxed{\vee_{\check{\mathbf{x}}}\frac{(\rho^h v^{\mathsf{L}} \vee 0)\,\mathbf{x}\,(\rho^h v^{\mathsf{R}} \vee 0)}{(\rho^h v^{\mathsf{L}}\,\mathbf{x}\,\rho^h v^{\mathsf{R}}) \vee (0\,\mathbf{x}\,0)}}\middle| v\right]_X \left[\rho^h \rho v \vee \boldsymbol{Z}_{\rho v} \middle| v\right]_{\underline{\langle\!\langle A_i \rangle\!\rangle}\setminus X} \langle\!\langle A_i \rangle\!\rangle\right\}\right]^{\{i\}} \quad ,$$

where

$$\mathsf{pr}\,\psi_i \equiv \mathsf{fl}\,\sigma_i\,\langle \boldsymbol{K}_h\{\boldsymbol{A}_h\}|y_h\rangle\,\boldsymbol{B}_i$$

as required. Note that, for all $v \in X$, we have $\boldsymbol{Z}_{v^{\mathsf{L}}} \equiv \boldsymbol{Z}_{v^{\mathsf{R}}} \equiv 0$ because $\pi v^{\mathsf{L}} \notin \{f,t\}$ and $\pi v^{\mathsf{R}} \notin \{f,t\}$. We define

$$\rho^i = \left[\mathsf{fl}\,\Pi^0_{\mathbf{x}}\rho^h\rho v \middle| v\right]_{\underline{\langle\!\langle A_i \rangle\!\rangle}} \qquad \text{and} \qquad \boldsymbol{A}_i \equiv \left[\left\lceil \rho^i v \right\rceil^{\{i\}} \vee \boldsymbol{Z}_v \middle| v\right]_{\underline{\langle\!\langle A_i \rangle\!\rangle}} \langle\!\langle A_i \rangle\!\rangle \quad ,$$

which satisfies Property 3 because for all $v \in X$, we have $\pi v \in \{f,t\}$ and $\boldsymbol{Z}_v \equiv 0\,\mathbf{x}\,0$. We set $d_i = d_h$ and

$$\tau_i = \left\langle z_1^{h\{i\}} \middle| z_1^i, \ldots, z_{d_h}^{h\{i\}} \middle| z_{d_i}^i \right\rangle \qquad \text{and} \qquad \boldsymbol{K}_i\{\,\} \equiv \left\lceil (z_1^i \vee \cdots (z_{d_i}^i \vee \{\,\}) \cdots) \right\rceil^{\{i\}} \quad ,$$

therefore we have

$$\mathsf{cn}\,\psi_i \equiv \mathsf{fl}\,\tau_i \boldsymbol{K}_i\{\boldsymbol{A}_i\}$$

as required.

Property 1 follows from the induction hypothesis:

$$\Sigma\,\langle \boldsymbol{B}_1|y_1\rangle \cdots \langle \boldsymbol{B}_{i-1}|y_{i-1}\rangle \left\langle y_h^{\{i\}} \middle| y_i \right\rangle y_i = \left\lceil \Sigma\,\langle \boldsymbol{B}_1|y_1\rangle \cdots \langle \boldsymbol{B}_h|y_h\rangle\, y_h \right\rceil^{\{i\}} = 1 \quad .$$

Property 2 also follows from the induction hypothesis:

$$\begin{aligned}
\Sigma\tau_1 \cdots \tau_i \boldsymbol{K}_i\{\,\} &\equiv \Sigma\tau_1 \cdots \tau_{i-1} \left\langle z_1^{h\{i\}} \middle| z_1^i, \ldots, z_{d_h}^{h\{i\}} \middle| z_{d_i}^i \right\rangle \left\lceil (z_1^i \vee \cdots (z_{d_i}^i \vee \{\,\}) \cdots) \right\rceil^{\{i\}} \\
&= \Sigma\tau_1 \cdots \tau_h \left\lceil (z_1^h \vee \cdots (z_{d_h}^h \vee \{\,\}) \cdots) \right\rceil^{\{i\}} \\
&= \left\lceil \Sigma\tau_1 \cdots \tau_h \boldsymbol{K}_h\{\,\} \right\rceil^{\{i\}} = \{\,\} \quad .
\end{aligned}$$

To prove Property 4 we use the induction hypothesis and we distinguish four cases:

1. If $v \in \underline{\langle\!\langle A_i \rangle\!\rangle} \setminus X$ and $\underline{\rho^h \rho v} = \{\pi\rho v\}$, then $\underline{\rho^i v} = \{\pi\rho v\} = \{\pi v\}$ and

$$\rho^i v \equiv \Pi^0_{\mathbf{x}} \rho^h \rho v \equiv \rho^h \rho v = \pi\rho v \equiv \pi v \quad ,$$

because $\Pi^0_{\mathbf{x}}$ does not apply.

2. If $v \in \underline{\langle\!\langle A_i \rangle\!\rangle} \setminus X$ and $\underline{\rho^h \rho v} = \emptyset$ and $\pi\rho v \in \{f, t\}$, then $\underline{\rho^i v} = \emptyset$ and $\pi v \equiv \pi\rho v$ and

$$\Sigma\rho^i v \equiv \Sigma\Pi^0_{\mathbf{x}} \rho^h \rho v \equiv \Sigma\rho^h \rho v = \Sigma\pi\rho v \equiv \Sigma\pi v \quad .$$

3. If $v \in X$ and $\pi v = f$ we have:

$$\begin{aligned} \rho^i v &\equiv \mathsf{fl}\,\Pi^0_{\mathbf{x}} \rho^h \rho v \equiv \mathsf{fl}\,\Pi^0_{\mathbf{x}} \rho^h (v^{\mathsf{L}} \,\mathbf{x}\, v^{\mathsf{R}}) \equiv \mathsf{fl}\,\Pi^0_{\mathbf{x}} (\rho^h v^{\mathsf{L}} \,\mathbf{x}\, \rho^h v^{\mathsf{R}}) \\ &= \mathsf{fl}\,\Pi^0_{\mathbf{x}} (\pi v^{\mathsf{L}} \,\mathbf{x}\, \pi v^{\mathsf{R}}) \equiv \mathsf{fl}\,\Pi^0_{\mathbf{x}} (u_{x0} \,\mathbf{x}\, u_{x1}) \equiv \mathsf{0} \,\mathbf{x}\, \mathsf{0} \quad , \end{aligned}$$

where we exploited the fact that $\pi v^{\mathsf{L}}, \pi v^{\mathsf{R}} \notin \{f, t\}$; note that $\underline{\rho^i v} = \emptyset$; then we have

$$\Sigma\rho^i v = \mathsf{0} \,\mathbf{x}\, \mathsf{0} = \mathsf{0} = \Sigma f = \Sigma\pi v \quad .$$

4. If $v \in X$ and $\pi v = t$ we proceed as for the previous case.

Properties 5, 6, 7 and 8 are satisfied by the construction.

Complexity. In the worst case, the height of a ψ_i is linear in the size of the given proof and its width quadratic; this might happen for a modus ponens

$$\mathsf{mp}\,\frac{A_h \quad A_h \to A_i}{A_i}$$

with a large A_h and where variables have accumulated many ranges. The height of all the ψ_is (except those simulating substitutions, which is constant) is dominated by the height of merge derivations, which is in turn linear on their width. Given that there is a linear number of ψ_is, the size of ϕ is at most quartic in the size of the given sFrege proof. Checking that $\mathsf{pr}\,\phi = 1$ and $\mathsf{cn}\,\phi = \Sigma\lceil A_n \rceil^+$ can be done in linear time on their sizes. □

This therefore shows that rKDT-ODSS is as powerful as sFrege. We do not show the converse here, that sFrege p-simulates rKDT-ODSS, but we have no reason to think that this is not the case.

Chapter 8

Cut Elimination

We now turn to the normalisation of strictly linear proofs by showing a cut elimination procedure. As we mention in the introduction, the motivation for developing a theory of strictly linear derivations is that 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, translate it to a strictly linear setting, normalise inside that system, and then project back to the original system without too much difficulty.

In this chapter, we show that we can give a straightforward and direct proof of cut elimination for a certain class of proofs, containing in particular all those obtained from an SKS-OD proof. We do this by applying the method from [5] of eliminating cuts via projections and committing to the lax notion of conjugacy, working within the system KDT-ODS.

A projection on a derivation is a pair of maps that, for some atom $\boldsymbol{a}$, produce the derivations in which that atom is either true or false. A decision tree $\boldsymbol{A}\,\boldsymbol{a}\,\boldsymbol{B}$ can be understood semantically as "if $\boldsymbol{a}$ is false then $\boldsymbol{A}$; otherwise $\boldsymbol{B}$", and so the projection maps work by only keeping what is in one side of the scope of the atom and discarding the rest. For example, the left-projection of $(\mathsf{0} \wedge \mathsf{1})\,\boldsymbol{a}\,(\mathsf{1} \wedge \mathsf{0})$ on $\boldsymbol{a}$ is $\mathsf{0} \wedge \mathsf{1}$ and its right-projection on $\boldsymbol{a}$ is $\mathsf{1} \wedge \mathsf{0}$.

The precise notion of projection which we use in this thesis differs from that used in [5]; in that paper, the projections operate on formulae with nested atoms such as $\boldsymbol{A}\,\boldsymbol{a}\,(\boldsymbol{B}\,\boldsymbol{a}\,\boldsymbol{C})$ by discarding everything which is in both the left- and right-scope of an atom, such as $\boldsymbol{B}$; whatever is discarded can then be reconstructed. In a strictly linear system we do not have the same freedom to discard and reconstruct logical material, and so we take care to

not discard anything.

We do this by working primarily on 'regular' derivations, which are those derivations which do not contain any such nested occurrences of an atom. Decision trees such as $\boldsymbol{A}\,\boldsymbol{a}\,(\boldsymbol{B}\,\boldsymbol{b}\,\boldsymbol{C})$ are permitted (assuming that none of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ themselves contain occurrences of $\boldsymbol{a}$ or $\boldsymbol{b}$), but not $\boldsymbol{A}\,\boldsymbol{a}\,(\boldsymbol{B}\,\boldsymbol{a}\,\boldsymbol{C})$. We show in this section that a regular derivation can be obtained from any SKS-OD derivation (and from any regular KDTEq-OD derivation), and so this includes all objects of interest.

We work in the lax system KDT-ODS because this gives us access to the mix rule $\check{\wedge}\dfrac{\boldsymbol{A}\wedge\boldsymbol{B}}{\boldsymbol{A}\vee\boldsymbol{B}}$, so that when we retain only what is in the left-scope of the atom $\boldsymbol{a}$ in the rule $\check{\wedge}_{\boldsymbol{a}}\dfrac{(\boldsymbol{A}\,\boldsymbol{a}\,\boldsymbol{B})\wedge(\boldsymbol{C}\,\boldsymbol{a}\,\boldsymbol{D})}{(\boldsymbol{A}\vee\boldsymbol{C})\,\boldsymbol{a}\,(\boldsymbol{B}\wedge\boldsymbol{D})}$ we get an instance of this rule. We do not prove any complexity result relating to cut elimination here, but it is hoped that this will be the subject of future work because deep inference systems in general possess good complexity properties with respect to cut elimination [7] and we expect that supersubstitutions will play a role in these investigations.

Definition 8.0.1. A *cut on* $\boldsymbol{a}$ in KDTEq-OD is any instance of the rule

$$\wedge^{\widehat{\boldsymbol{a}}}\frac{(\boldsymbol{A}\,\boldsymbol{a}\,\boldsymbol{B})\wedge(\boldsymbol{C}\,\boldsymbol{a}\,\boldsymbol{D})}{(\boldsymbol{A}\wedge\boldsymbol{C})\,\boldsymbol{a}\,(\boldsymbol{B}\wedge\boldsymbol{D})}$$

such that $\boldsymbol{A}=\mathsf{0}=\boldsymbol{D}$ and $\boldsymbol{B}=\mathsf{1}=\boldsymbol{C}$, or $\boldsymbol{A}=\mathsf{1}=\boldsymbol{D}$ and $\boldsymbol{B}=\mathsf{0}=\boldsymbol{C}$. In the system KDT-ODSS, we take explicit substitutions in the context into account, and so a *cut on* $\boldsymbol{a}$ is any subderivation

$$\boldsymbol{K}\left\{\wedge^{\widehat{\boldsymbol{a}}}\frac{\boldsymbol{A}}{\boldsymbol{B}}\right\}$$

inside a derivation such that $\mathsf{fl}\,\boldsymbol{K}\{\boldsymbol{A}\}$ and $\mathsf{fl}\,\boldsymbol{K}\{\boldsymbol{B}\}$ when vertically composed form a cut on $\boldsymbol{a}$.

We note that when we translate an SKS-OD derivation to KDTEq-OD, as described in Remark 3.3.8, instances of the cut rule $\bar{\mathsf{i}}$ become cuts as described here; and when we simulate an instance of modus ponens in rKDT-ODSS, as shown in Figure 7.6, the subderivation χ_3^i contains such a cut for every atom

$\boldsymbol{a}$ appearing in $\boldsymbol{A}_h$. The projection to SKS-OD of a derivation in KDTEq-OD containing such a cut will exhibit a cut $\bar{\mathsf{i}}$ in the same location.

By projecting a cut to the left on the atom $\boldsymbol{a}$, we will get $\dfrac{\boldsymbol{A} \wedge \boldsymbol{C}}{\boldsymbol{A} \wedge \boldsymbol{C}}$; and projecting to the right on $\boldsymbol{a}$ we will get $\dfrac{\boldsymbol{B} \wedge \boldsymbol{D}}{\boldsymbol{B} \wedge \boldsymbol{D}}$. Both of these are equal to $\mathsf{0}$ under the condition on the value of the formulae in the cut.

8.1 Cut Elimination via KDTEq-OD

In [5], it is shown that cuts can be eliminated from proofs in KDTEq-OD using projections. That is, given a proof $\phi \in$ KDTEq-OD, it is shown that there exists a proof

$$= \dfrac{1}{\boxed{\begin{array}{c}1\\ \mathsf{l}_{\boldsymbol{a}}\,\phi \,\|\\ \mathsf{l}_{\boldsymbol{a}}\,\mathsf{cn}\,\phi\end{array}}\ \boldsymbol{a}\ \boxed{\begin{array}{c}1\\ \mathsf{r}_{\boldsymbol{a}}\,\phi \,\|\\ \mathsf{r}_{\boldsymbol{a}}\,\mathsf{cn}\,\phi\end{array}}\qquad \|\qquad \mathsf{cn}\,\phi}\ ,$$

where $\mathsf{l}_{\boldsymbol{a}}\,\phi$ and $\mathsf{r}_{\boldsymbol{a}}\,\phi$ are obtained from ϕ by projecting to the left and right respectively.

Therefore, given a proof $\phi \in$ rKDT-ODS, we can flatten all of the substitutions to obtain a proof in rKDTEq-OD, perform the cut elimination procedure to obtain a cut-free proof in KDTEq-OD, and then eliminate the created unit equalities to obtain a proof in rKDT-OD $\subset$ rKDT-ODS. Any substitutions in $\mathsf{cn}\,\phi$ can then be factored out of the conclusion of the resulting proof. It suffices to show that the procedure for eliminating unit equalities given in Chapter 6 does not create any cuts, which is straightforward by examining the constructed merges and eversions.

Proposition 8.1.1. *For any cut-free proof $\phi \in$ rKDTEq-OD, the structurally equivalent proof $\phi' \in$ rKDT-ODS constructed as described in Theorem 6.3.1 is cut-free.*

Proof. The only inference steps created in the construction are in the up- and down-merge constructions and the eversion constructions.

All of the leaves in an eversion get the same unit substituted onto them, so no inference step created in the eversion can satisfy the condition on the value of the cut.

All of the inference steps created in a down-merge are saturated in the conclusion, and so cannot be a cut.

If an inference step $\wedge\hat{\boldsymbol{a}}\,\dfrac{(\boldsymbol{A}\,\boldsymbol{a}\,\boldsymbol{B})\wedge(\boldsymbol{C}\,\boldsymbol{a}\,\boldsymbol{D})}{(\boldsymbol{A}\wedge\boldsymbol{C})\,\boldsymbol{a}\,(\boldsymbol{B}\wedge\boldsymbol{D})}$ is created in an up-merge, then $C = D$, corresponding to the value of the unit (or equivalently $A = B$, when the unit is on the left in the unit-equality inference step).

Therefore no cuts are created in the construction. □

This procedure works for all proofs in KDT-ODS, so this is a stronger result than the one shown in the next section. However, this procedure necessitates applying all substitutions and leaving the strictly linear proof system. Therefore, when possible, we prefer to eliminate the cuts as shown in the next section, and we show that this procedure applies to all the proofs obtained from a standard proof in SKS-OD.

8.2 Cut Elimination for Regular Proofs

This section consists of two parts: first we define *regularity* and show how we can modify the transformation in Chapter 6 to preserve this property; second we show how the cuts can be eliminated from any regular proof via projections. The modification of the transformation in Chapter 6 is not necessary in order to give a direct proof of cut elimination for those KDT-ODS proofs obtained from SKS-OD, but it does make this proof more straightforward and allows us to characterise cut elimination for a larger class of proofs which includes those regular KDT-ODS proofs which are not obtained from a proof at the standard level.

Definition 8.2.1. We call a KDT-ODS derivation *regular* if there is no section whose flat expansion is of the form $\boldsymbol{K}\{\mathbf{H}\{\boldsymbol{A}\,\boldsymbol{a}\,\boldsymbol{B}\}\,\boldsymbol{a}\,\boldsymbol{C}\}$ or $\boldsymbol{K}\{\boldsymbol{A}\,\boldsymbol{a}\,\{\mathbf{H}\{\boldsymbol{B}\,\boldsymbol{a}\,\boldsymbol{C}\}\}$.

Definition 8.2.2. For a regular derivation $\phi \in$ KDT-ODS and an atom $\boldsymbol{a}$ 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 $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \mathsf{I}_{\boldsymbol{a}} \, \psi$.
- If $\phi \equiv \psi \, \beta \, \chi$ for $\beta \not\equiv \boldsymbol{a}$ then $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \mathsf{I}_{\boldsymbol{a}} \, \psi \, \beta \, \mathsf{I}_{\boldsymbol{a}} \, \chi$.
- If $\phi \equiv \langle \psi | x \rangle \, \chi$ then $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \big\langle \mathsf{I}_{\boldsymbol{a}} \, \psi \big| x \big\rangle \, \mathsf{I}_{\boldsymbol{a}} \, \chi$.
- If $\phi \equiv \boxed{\begin{array}{c} \psi \\ \sim \\ \chi \end{array}}$ then $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \boxed{\begin{array}{c} \mathsf{I}_{\boldsymbol{a}} \, \psi \\ \sim \\ \mathsf{I}_{\boldsymbol{a}} \, \chi \end{array}}$.
- If $\phi \equiv \boxed{\check{\wedge}_{\boldsymbol{a}} \dfrac{\psi}{\dfrac{(\boldsymbol{A} \, \boldsymbol{a} \, \boldsymbol{B}) \wedge (\boldsymbol{C} \, \boldsymbol{a} \, \boldsymbol{D})}{(\boldsymbol{A} \vee \boldsymbol{C}) \, \boldsymbol{a} \, (\boldsymbol{B} \wedge \boldsymbol{D})}}{\chi}}$ or $\phi \equiv \boxed{\hat{\vee}_{\boldsymbol{a}} \dfrac{\psi}{\dfrac{(\boldsymbol{A} \wedge \boldsymbol{C}) \, \boldsymbol{a} \, (\boldsymbol{B} \vee \boldsymbol{D})}{(\boldsymbol{A} \, \boldsymbol{a} \, \boldsymbol{B}) \vee (\boldsymbol{C} \, \boldsymbol{a} \, \boldsymbol{D})}}{\chi}}$ then $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \boxed{\check{\wedge} \dfrac{\dfrac{\mathsf{I}_{\boldsymbol{a}} \, \psi}{\mathsf{I}_{\boldsymbol{a}} \, \boldsymbol{A} \wedge \mathsf{I}_{\boldsymbol{a}} \, \boldsymbol{C}}}{\dfrac{\mathsf{I}_{\boldsymbol{a}} \, \boldsymbol{A} \vee \mathsf{I}_{\boldsymbol{a}} \, \boldsymbol{C}}{\mathsf{I}_{\boldsymbol{a}} \, \chi}}}$.
- If $\phi \equiv \boxed{\check{\boldsymbol{a}}_{\beta} \dfrac{\psi}{\dfrac{(\boldsymbol{A} \, \beta \, \boldsymbol{B}) \, \boldsymbol{a} \, (\boldsymbol{C} \, \beta \, \boldsymbol{D})}{(\boldsymbol{A} \, \boldsymbol{a} \, \boldsymbol{C}) \, \beta \, (\boldsymbol{B} \, \boldsymbol{a} \, \boldsymbol{D})}}{\chi}}$ or $\phi \equiv \boxed{\hat{\boldsymbol{a}}_{\beta} \dfrac{\psi}{\dfrac{(\boldsymbol{A} \, \boldsymbol{a} \, \boldsymbol{C}) \, \beta \, (\boldsymbol{B} \, \boldsymbol{a} \, \boldsymbol{D})}{(\boldsymbol{A} \, \beta \, \boldsymbol{B}) \, \boldsymbol{a} \, (\boldsymbol{C} \, \beta \, \boldsymbol{D})}}{\chi}}$ then $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \boxed{\begin{array}{c} \mathsf{I}_{\boldsymbol{a}} \, \psi \\ \cdots\cdots \\ \mathsf{I}_{\boldsymbol{a}} \, \chi \end{array}}$, and similarly for $\boldsymbol{a}\check{\wedge}$ and $\boldsymbol{a}\hat{\vee}$. Note that here β cannot be $\boldsymbol{a}$ due to the assumption that ϕ is regular.
- If $\phi \equiv \boxed{r \dfrac{\psi}{\chi}}$ in any other case, then $\mathsf{I}_{\boldsymbol{a}} \, \phi \equiv \boxed{r \dfrac{\mathsf{I}_{\boldsymbol{a}} \, \psi}{\mathsf{I}_{\boldsymbol{a}} \, \chi}}$; note here again that r cannot be $\boldsymbol{a}\check{\boldsymbol{a}}$ due to the assumption that ϕ is regular.

The *right-projection on* $\boldsymbol{a}$ of a regular derivation ϕ is defined analogously and denoted by $\mathsf{r}_{\boldsymbol{a}} \, \phi$.

Remark 8.2.3. For any atom $\boldsymbol{a}$ and any regular derivation $\phi \in \mathsf{KDT\text{-}ODS}$, $\mathsf{I}_{\boldsymbol{a}} \, \phi$ and $\mathsf{r}_{\boldsymbol{a}} \, \phi$ are uniquely determined. For any distinct atoms $\boldsymbol{a}$ and $\boldsymbol{b}$,

$\pi_{\boldsymbol{a}} \in \{\mathsf{l}_{\boldsymbol{a}}, \mathsf{r}_{\boldsymbol{a}}\}$ and $\pi_{\boldsymbol{b}} \in \{\mathsf{l}_{\boldsymbol{b}}, \mathsf{r}_{\boldsymbol{b}}\}$ commute. 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 8.2.4. For any regular derivation ϕ and any atom $\boldsymbol{a}$, the projections $\mathsf{l}_{\boldsymbol{a}}\,\phi$ and $\mathsf{r}_{\boldsymbol{a}}\,\phi$ contain no occurrences of $\boldsymbol{a}$, and so neither contains any cuts on $\boldsymbol{a}$.

Example 8.2.5. Let

$$\phi \equiv \boldsymbol{a}\check{\boldsymbol{b}}\dfrac{\wedge\hat{\boldsymbol{a}}\dfrac{(0\,\boldsymbol{a}\,1)\wedge(1\,\boldsymbol{a}\,0)}{(0\wedge 1)\,\boldsymbol{a}\,(0\wedge 1)}\;\boldsymbol{b}\,(x\,\boldsymbol{a}\,y)}{((0\wedge 1)\,\boldsymbol{b}\,x)\,\boldsymbol{a}\,((0\wedge 1)\,\boldsymbol{b}\,y)} \quad .$$

Then

$$\mathsf{l}_{\boldsymbol{a}}\,\phi \equiv \dfrac{\begin{array}{c}0\wedge 1\\ \cdots\cdots\\ 0\wedge 1\end{array}\;\boldsymbol{b}\,x}{(0\wedge 1)\,\boldsymbol{b}\,x} \qquad \mathsf{r}_{\boldsymbol{a}}\,\phi \equiv \dfrac{\begin{array}{c}1\wedge 0\\ \cdots\cdots\\ 0\wedge 1\end{array}\;\boldsymbol{b}\,y}{(0\wedge 1)\,\boldsymbol{b}\,y}$$

$$\mathsf{l}_{\boldsymbol{b}}\,\phi \equiv \dfrac{\wedge\hat{\boldsymbol{a}}\dfrac{(0\,\boldsymbol{a}\,1)\wedge(1\,\boldsymbol{a}\,0)}{(0\wedge 1)\,\boldsymbol{a}\,(0\wedge 1)}}{(0\wedge 1)\,\boldsymbol{a}\,(0\wedge 1)} \qquad \mathsf{r}_{\boldsymbol{b}}\,\phi \equiv \dfrac{x\,\boldsymbol{a}\,y}{x\,\boldsymbol{a}\,y}$$

Although ϕ contains a cut on $\boldsymbol{a}$, neither $\mathsf{l}_{\boldsymbol{a}}\,\phi$ nor $\mathsf{r}_{\boldsymbol{a}}\,\phi$ do.

It can be the case that eliminating the unit-equality inference steps from a regular derivation results in irregularity.

Example 8.2.6. Let

$$\phi \equiv \dfrac{=\dfrac{x\,\boldsymbol{a}\,y}{(x\,\boldsymbol{a}\,y)\wedge 1}}{=\dfrac{(x\,\boldsymbol{a}\,y)\wedge 1}{x\,\boldsymbol{a}\,y}} \quad .$$

Applying Phase 1 of the construction, given in Figure 6.1, results in the following pre-derivation:

$$\phi' \equiv \dfrac{\wedge\check{\boldsymbol{a}}\dfrac{(x\wedge 1)\,\boldsymbol{a}\,(y\wedge 1)}{(x\,\boldsymbol{a}\,y)\wedge(1\,\boldsymbol{a}\,1)}}{\langle 1\wedge 1|u\rangle\dfrac{(x\,\boldsymbol{a}\,y)\wedge u}{x\,\boldsymbol{a}\,y}} \quad .$$

Phase 2 of the construction, given in Figure 6.2, does not take into account that all of the necessary logical material is already present, resulting in the following derivation:

$$\phi'' \equiv \boxed{\begin{array}{c} \langle 1\,\boldsymbol{a}\,1|u\rangle \boxed{\wedge\breve{\boldsymbol{a}}\,\dfrac{(x\wedge u)\,\boldsymbol{a}\,(y\wedge u)}{(x\,\boldsymbol{a}\,y)\wedge(u\,\boldsymbol{a}\,u)}} \\ \hline \left\langle \boldsymbol{a}\breve{\boldsymbol{a}}\,\dfrac{(1\,\boldsymbol{a}\,1)\,\boldsymbol{a}\,(1\,\boldsymbol{a}\,1)}{(1\,\boldsymbol{a}\,1)\,\boldsymbol{a}\,(1\,\boldsymbol{a}\,1)} \middle| u \right\rangle ((x\,\boldsymbol{a}\,y)\wedge u) \\ \hline \langle 1\,\boldsymbol{a}\,1|u\rangle \boxed{\wedge\hat{\boldsymbol{a}}\,\dfrac{(x\,\boldsymbol{a}\,y)\wedge(u\,\boldsymbol{a}\,u)}{(x\wedge u)\,\boldsymbol{a}\,(y\wedge u)}} \end{array}} \quad ,$$

which shows irregularity inside the eversion.

Any irregularity created in this way is redundant in the construction. To see this, consider the procedure for eliminating a pair of unit equations such as in the following derivation:

$$\pi\,\boxed{\begin{array}{c} \phi \\ \hdashline \boldsymbol{K}\left\{ = \dfrac{\boldsymbol{A}\{w\,\boldsymbol{a}\,x\}}{\boldsymbol{A}\{w\,\boldsymbol{a}\,x\}\;\alpha\;u} \right\} \\ \hdashline \psi \\ \hdashline \mathbf{H}\left\{ = \dfrac{\boldsymbol{B}\{y\,\boldsymbol{a}\,z\}\;\beta\;u}{\boldsymbol{B}\{y\,\boldsymbol{a}\,z\}} \right\} \\ \hdashline \chi \end{array}}$$

The merge constructions by which we simulate the unit-equality inference steps propagate upwards a substitution $\left\langle \breve{\boldsymbol{A}}^u\{u\,\boldsymbol{a}\,u\} \middle| u \right\rangle$ and downwards a substitution $\left\langle \breve{\boldsymbol{B}}^u\{u\,\boldsymbol{a}\,u\} \middle| u \right\rangle$, which are resolved by an eversion which contains 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. We can alter the construction so that this material is created only once and shared between the merge constructions, replacing the inference step $\boldsymbol{a}\breve{\boldsymbol{a}}$ by the formula $(u\,\boldsymbol{a}\,u)$.

Example 8.2.7. The derivation

$$\phi \equiv \left[\begin{array}{c} = \dfrac{x \; \boldsymbol{a} \; y}{(x \; \boldsymbol{a} \; y) \wedge 1} \\ = \dfrac{(x \; \boldsymbol{a} \; y) \wedge 1}{x \; \boldsymbol{a} \; y} \end{array}\right]$$

can be simulated by the regular derivation

$$\phi'' \equiv \left[\begin{array}{c} \wedge\check{\boldsymbol{a}} \dfrac{(x \wedge 1) \; \boldsymbol{a} \; (y \wedge 1)}{(x \; \boldsymbol{a} \; y) \wedge (1 \; \boldsymbol{a} \; 1)} \\ \underbrace{\langle 1 \; \boldsymbol{a} \; 1 | u \rangle \; ((x \; \boldsymbol{a} \; y) \wedge u)} \\ \wedge\hat{\boldsymbol{a}} \dfrac{(x \; \boldsymbol{a} \; y) \wedge (1 \; \boldsymbol{a} \; 1)}{(x \wedge 1) \; \boldsymbol{a} \; (y \wedge 1)} \end{array}\right] \quad ,$$

The previous example only considers the case where there is a single atom shared between the formulae in the unit-equality inference steps, but in general the formulae may have many atoms in common and we must address all of them in order to ensure regularity. We can do this by taking composite projections of the formulae to be propagated in such a way that we create only the necessary material and avoid substituting any atom into its own scope.

Notation 8.2.8. Let $\boldsymbol{A}$ be an open and regular formula with variables $\{x_1, \ldots, x_m\}$ such that each variable occurs exactly once in $\boldsymbol{A}$. We denote by $\pi_i^{\boldsymbol{A}}$ the projection induced by x_i in $\boldsymbol{A}$. That is, for each atom $\boldsymbol{a}$, $\mathsf{l}_{\boldsymbol{a}}$ is in $\pi_i^{\boldsymbol{A}}$ if x_i occurs in $\boldsymbol{A}$ in the left-scope of the atom $\boldsymbol{a}$, $\mathsf{r}_{\boldsymbol{a}}$ is in $\pi_i^{\boldsymbol{A}}$ if x_i occurs in $\boldsymbol{A}$ in the right-scope of the atom $\boldsymbol{a}$, and neither is in $\pi_i^{\boldsymbol{A}}$ if x_i does not occur in $\boldsymbol{A}$ in the scope of the atom $\boldsymbol{a}$. By the assumptions of regularity and that each variable occurs exactly once, these three cases are mutually exclusive.

For example, if $\boldsymbol{A} \equiv (x_1 \alpha x_2) \boldsymbol{a} ((x_3 \boldsymbol{b} x_4) \wedge x_5)$ then $\pi_1^{\boldsymbol{A}} = \mathsf{l}_{\boldsymbol{a}}$ and $\pi_4^{\boldsymbol{A}} = \mathsf{r}_{\boldsymbol{a}} \, \mathsf{r}_{\boldsymbol{b}}$.

Lemma 8.2.9. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be open and regular* KDT *formulae; let $\underline{\boldsymbol{A}} = \{x_1, \ldots, x_m\}$ and $\underline{\boldsymbol{B}} = \{y_1, \ldots, y_n\}$; and suppose that each variable x_i occurs exactly once in $\boldsymbol{A}$ and each variable y_j occurs exactly once in $\boldsymbol{B}$. Then there exist regular derivations in* KDT-ODS*:*

$$\begin{array}{ccc} \left[\pi_j^{\boldsymbol{B}} \widehat{\boldsymbol{A}} \middle| y_j\right]_{\underline{\boldsymbol{B}}} \check{\boldsymbol{B}} & & \left[\pi_j^{\boldsymbol{B}} \widehat{\boldsymbol{A}}^{y_j} \middle| y_j\right]_{\underline{\boldsymbol{B}}} \check{\boldsymbol{B}} \\ \| & & \| \\ \left[\pi_i^{\boldsymbol{A}} \check{\boldsymbol{B}}^{x_i} \middle| x_i\right]_{\underline{\boldsymbol{A}}} \widehat{\boldsymbol{A}} & & \left[\pi_i^{\boldsymbol{A}} \check{\boldsymbol{B}} \middle| x_i\right]_{\underline{\boldsymbol{A}}} \widehat{\boldsymbol{A}} \end{array} \quad ,$$

where $\boldsymbol{B}^{x_i} \equiv [x_i|y_j]_{\underline{\boldsymbol{B}}}\boldsymbol{B}$ *and* $\boldsymbol{A}^{y_j} \equiv [y_j|x_i]_{\underline{\boldsymbol{A}}}\boldsymbol{A}$.

Proof. We consider the derivation on the left, the one on the right being analogous. The proof follows the same induction as the proof of the Eversion Lemma 5.2.1, differing only in the case where $\boldsymbol{A} \equiv \boldsymbol{A}_1 \boldsymbol{a} \boldsymbol{A}_2$ and $\boldsymbol{B} \equiv \boldsymbol{B}_1 \boldsymbol{a} \boldsymbol{B}_2$. In this case we build the following derivation:

$$\dfrac{\left[\pi_j^{\boldsymbol{B}}\left(\widehat{\boldsymbol{A}}_1 \,\boldsymbol{a}\, \widehat{\boldsymbol{A}}_2\right)\middle| y_j\right]_{\underline{\boldsymbol{B}}}\left(\breve{\boldsymbol{B}}_1 \,\boldsymbol{a}\, \breve{\boldsymbol{B}}_2\right)}{\left[\pi_i^{\boldsymbol{A}}\left(\breve{\boldsymbol{B}}_1 \,\boldsymbol{a}\, \breve{\boldsymbol{B}}_2\right)\middle| x_i\right]_{\underline{\boldsymbol{A}}}\left(\widehat{\boldsymbol{A}}_1 \,\boldsymbol{a}\, \widehat{\boldsymbol{A}}_2\right)}$$

$$\dfrac{\dfrac{\left[\mathsf{l}_{\boldsymbol{a}}\,\pi_j^{\boldsymbol{B}_1}\left(\widehat{\boldsymbol{A}}_1 \,\boldsymbol{a}\, \widehat{\boldsymbol{A}}_2\right)\middle| y_j\right]_{\underline{\boldsymbol{B}_1}}\breve{\boldsymbol{B}}_1}{\left[\pi_i^{\boldsymbol{A}_1}\breve{\boldsymbol{B}}_1^{x_i}\middle| x_i\right]_{\underline{\boldsymbol{A}_1}}\widehat{\boldsymbol{A}}_1}\mathsf{IH}\,\|}{\left[\mathsf{l}_{\boldsymbol{a}}\,\pi_i^{\boldsymbol{A}}\left(\breve{\boldsymbol{B}}_1^{x_i} \,\boldsymbol{a}\, \breve{\boldsymbol{B}}_2^{x_i}\right)\right]} \;\boldsymbol{a}\; \dfrac{\dfrac{\left[\mathsf{r}_{\boldsymbol{a}}\,\pi_j^{\boldsymbol{B}_2}\left(\widehat{\boldsymbol{A}}_1 \,\boldsymbol{a}\, \widehat{\boldsymbol{A}}_2\right)\middle| y_j\right]_{\underline{\boldsymbol{B}_2}}\breve{\boldsymbol{B}}_2}{\left[\pi_i^{\boldsymbol{A}_2}\breve{\boldsymbol{B}}_2^{x_i}\middle| x_i\right]_{\underline{\boldsymbol{A}_2}}\widehat{\boldsymbol{A}}_2}\mathsf{IH}\,\|}{\left[\mathsf{r}_{\boldsymbol{a}}\,\pi_i^{\boldsymbol{A}}\left(\breve{\boldsymbol{B}}_1^{x_i} \,\boldsymbol{a}\, \breve{\boldsymbol{B}}_2^{x_i}\right)\right]}$$

□

This construction coincides with the normal eversion shown in Lemma 5.2.1 whenever at most one of $\boldsymbol{A}$ and $\boldsymbol{B}$ contains the atom $\boldsymbol{a}$. When both $\boldsymbol{A}$ and $\boldsymbol{B}$ contain the atom $\boldsymbol{a}$, this construction avoids creating more material than necessary, essentially replacing the inference step $\boldsymbol{a}\breve{\boldsymbol{a}}\dfrac{(x\,\boldsymbol{a}\,y)\,\boldsymbol{a}\,(x\,\boldsymbol{a}\,y)}{(x\,\boldsymbol{a}\,x)\,\boldsymbol{a}\,(y\,\boldsymbol{a}\,y)}$ in the normal eversion by $\dfrac{x\,\boldsymbol{a}\,y}{x\,\boldsymbol{a}\,y}$ and propagating accordingly, leaving the construction otherwise unchanged.

We note that in general this transformation would affect the value of parts of the derivation, because $x \neq x\,\boldsymbol{a}\,y \neq y$. However, according to our semantic interpretation of decision trees, any subderivation which occurs in both the left- and right-scope of an atom $\boldsymbol{a}$ does not affect the value of the overall derivation. Moreover, in using this construction to eliminate unit equalities, we will substitute the same unit onto every leaf of the derivation so the value will be unchanged.

Example 8.2.10. Let $\widehat{\boldsymbol{A}} \equiv (x_1 \,\boldsymbol{a}\, (x_2 \,\boldsymbol{b}\, x_3)) \wedge x_4$ and $\breve{\boldsymbol{B}} \equiv ((y_1 \,\boldsymbol{a}\, y_2) \vee y_3) \,\boldsymbol{b}\, y_4$. We then obtain the following induced projections:

$$\pi_1^{\boldsymbol{B}}\widehat{\boldsymbol{A}} \equiv \mathsf{l}_{\boldsymbol{a}}\,\mathsf{l}_{\boldsymbol{b}}\,\widehat{\boldsymbol{A}} \equiv x_1 \wedge x_4 \qquad \pi_1^{\boldsymbol{A}}\check{\boldsymbol{B}} \equiv \mathsf{l}_{\boldsymbol{a}}\,\check{\boldsymbol{B}} \equiv (y_1 \vee y_3)\,\boldsymbol{b}\,y_4$$

$$\pi_2^{\boldsymbol{B}}\widehat{\boldsymbol{A}} \equiv \mathsf{r}_{\boldsymbol{a}}\,\mathsf{l}_{\boldsymbol{b}}\,\widehat{\boldsymbol{A}} \equiv x_2 \wedge x_4 \qquad \pi_2^{\boldsymbol{A}}\check{\boldsymbol{B}} \equiv \mathsf{l}_{\boldsymbol{b}}\,\mathsf{r}_{\boldsymbol{a}}\,\check{\boldsymbol{B}} \equiv y_2 \vee y_3$$

$$\pi_3^{\boldsymbol{B}}\widehat{\boldsymbol{A}} \equiv \mathsf{l}_{\boldsymbol{b}}\,\widehat{\boldsymbol{A}} \equiv (x_1\,\boldsymbol{a}\,x_2) \wedge x_4 \qquad \pi_3^{\boldsymbol{A}}\check{\boldsymbol{B}} \equiv \mathsf{r}_{\boldsymbol{b}}\,\mathsf{r}_{\boldsymbol{a}}\,\check{\boldsymbol{B}} \equiv y_4$$

$$\pi_4^{\boldsymbol{B}}\widehat{\boldsymbol{A}} \equiv \mathsf{r}_{\boldsymbol{b}}\,\widehat{\boldsymbol{A}} \equiv (x_1\,\boldsymbol{a}\,x_3) \wedge x_4 \qquad \pi_4^{\boldsymbol{A}}\check{\boldsymbol{B}} \equiv \check{\boldsymbol{B}} \equiv ((y_1\,\boldsymbol{a}\,y_2) \vee y_3)\,\boldsymbol{b}\,y_4$$

We construct the following regular eversion with premise $\left[\pi_j^{\boldsymbol{B}}\widehat{\boldsymbol{A}}\middle|y_j\right]_{\underline{\boldsymbol{B}}}\check{\boldsymbol{B}}$ and conclusion $\left[\pi_i^{\boldsymbol{A}}\check{\boldsymbol{B}}^{x_i}\middle|x_i\right]_{\underline{\boldsymbol{A}}}\widehat{\boldsymbol{A}}$:

$$\wedge\check{\boldsymbol{b}}\,\dfrac{\boxed{\wedge\check{\boldsymbol{a}}\,\dfrac{\boxed{(x_1 \wedge x_4)\,\boldsymbol{a}\,(x_2 \wedge x_4)}}{(x_1\,\boldsymbol{a}\,x_2) \wedge (x_4\,\boldsymbol{a}\,x_4)} \vee ((x_1\,\boldsymbol{a}\,x_2) \wedge x_4)}{\wedge\check{\vee}\,\dfrac{\boxed{\phantom{(x_1 \wedge x_4)\,\boldsymbol{a}\,(x_2 \wedge x_4)}}}{((x_1\,\boldsymbol{a}\,x_2) \vee (x_1\,\boldsymbol{a}\,x_2)) \wedge ((x_4\,\boldsymbol{a}\,x_4) \vee x_4)}}\,\boldsymbol{b}\,((x_1\,\boldsymbol{a}\,x_3) \wedge x_4)}{\boxed{\boldsymbol{a}\check{\boldsymbol{b}}\,\dfrac{\boxed{\boldsymbol{a}\check{\vee}\,\dfrac{\boxed{(x_1\,\boldsymbol{a}\,x_2) \vee (x_1\,\boldsymbol{a}\,x_2)}}{(x_1 \vee x_1)\,\boldsymbol{a}\,(x_2 \vee x_2)}\,\boldsymbol{b}\,(x_1\,\boldsymbol{a}\,x_3)}}{((x_1 \vee x_1)\,\boldsymbol{b}\,x_1)\,\boldsymbol{a}\,((x_2 \vee x_2)\,\boldsymbol{b}\,x_3)}} \wedge (((x_4\,\boldsymbol{a}\,x_4) \vee x_4)\,\boldsymbol{b}\,x_4)}\,.$$

Corollary 8.2.11. *Given any regular derivation ϕ in* KDTEq-OD*, we can build a regular derivation ϕ' in* KDT-ODS *such that ϕ and ϕ' are structurally equivalent for the abstraction* kdt*.*

Proof. We proceed as in the proof of Theorem 6.3.1. However, we replace the eversion in each ω_j in Figure 6.2 by the regular eversion whose existence is proven in Lemma 8.2.9, and propagate the corresponding substitutions though the construction.

The same argument for structural preservation applies in this case: when we substitute the same unit onto every leaf in the regular eversion, the abstraction map kdt will collapse the eversion to that unit; and the propagated substitutions will behave in the same way. □

Corollary 8.2.12. *Given any derivation ϕ of A in* SKS-OD*, we can build a corresponding regular derivation ϕ' of $\boldsymbol{A}'$ in* KDT-ODS*, where A is equal to the* SKS *interpretation of $\boldsymbol{A}'$.*

Proof. We translate the SKS-OD derivation into KDTEq-OD as shown in Remark 3.3.8. This translation does not result in any nesting of atoms, so the resulting derivation is regular. We can then eliminate the unit equalities as shown in Corollary 8.2.11 to obtain a regular derivation in KDT-ODS. □

In Chapters 4 and 6, we emphasised that the structural equivalence between a KDTEq-OD derivation and the rKDT-ODS derivation obtained by eliminating its unit-equality inference steps was sufficient to ensure that normalisation procedures would not be affected by this transformation. In this chapter, we have modified the procedure for eliminating the unit-equality inference steps. This may seem to be contradictory.

In fact we could have left the procedure intact. We show in Proposition 8.1.1 that it never creates any cuts, so we could have defined the projection maps to ignore any material created by it. However, it seems that an inevitable consequence of doing so would be that such a notion of projection could only apply to those KDT-ODSS derivations obtained from regular KDTEq-OD derivations and not to the larger class of all regular KDT-ODSS. We prefer to give a definition of projection which is more straightforward and leads to obtaining a proof of cut elimination for a larger class of proofs.

Moreover, it is striking that the Eversion Lemma can be modified to preserve a property of interest such as regularity in this way. Neither of Lemma 5.2.1 and Lemma 8.2.9 imply the other, although they do coincide when the formulae concerned are regular and have no overlap between their atoms, suggesting that they might both be examples of a more general construction.

The following proposition shows how to reconstruct a formula given its two projections. Its proof follows the same scheme as that of the Merge Lemma 5.1.17. For simplicity, and without loss of generality, we state it for open formulae, *i.e.*, formulae where no units appear. It is a special case of Lemma 8.2.9, but we show the full induction in the proof to show that no cuts are created.

Proposition 8.2.13. *For every atom* $\boldsymbol{a}$ *and every open and regular* KDT *formula* $\boldsymbol{A}$ *such that all its variables are maximally renamed apart, there exists a cut-free derivation*

$$\chi_{\boldsymbol{a}}(\boldsymbol{A}) \equiv \boxed{\begin{array}{c} \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{A}\ \boldsymbol{a}\ \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{A} \\ \Vdots\ \mathsf{KDT\text{-}ODS} \\ [v\ \boldsymbol{a}\ v|v]_V\, \boldsymbol{A} \end{array}} \quad ,$$

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

Proof. We proceed by induction on the structure of $\boldsymbol{A}$.

- If $\boldsymbol{A} \in \mathcal{V}$ then $\chi_{\boldsymbol{a}}(\boldsymbol{A}) \equiv \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{A}\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{A} \equiv \boldsymbol{A}\, \boldsymbol{a}\, \boldsymbol{A} \equiv [v\, \boldsymbol{a}\, v|v]_{\{\boldsymbol{A}\}}\, \boldsymbol{A}$.
- If $\boldsymbol{A} \equiv \boldsymbol{B}\, \boldsymbol{a}\, \boldsymbol{C}$ then, because $\boldsymbol{A}$ is regular, $\chi_{\boldsymbol{a}}(\boldsymbol{A}) \equiv \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{A}\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{A} \equiv \boldsymbol{B}\, \boldsymbol{a}\, \boldsymbol{C} \equiv [v\, \boldsymbol{a}\, v|v]_{\emptyset}\, \boldsymbol{A}$.
- If $\boldsymbol{A} \equiv \boldsymbol{B}\, \beta\, \boldsymbol{C}$ for $\beta \not\equiv \boldsymbol{a}$ then we build

$$\chi_{\boldsymbol{a}}\boldsymbol{A} \equiv \boxed{{}^{\beta\breve{\boldsymbol{a}}}\frac{\left(\mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{B}\, \beta\, \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{C}\right)\, \boldsymbol{a}\, \left(\mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{B}\, \beta\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{C}\right)}{\boxed{\begin{array}{l}\mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{B}\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{B} \\ \chi_{\boldsymbol{a}}(\boldsymbol{B})\,\| \\ [v\, \boldsymbol{a}\, v|v]_U\, \boldsymbol{B}\end{array}}\, \beta\, \boxed{\begin{array}{l}\mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{C}\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{C} \\ \chi_{\boldsymbol{a}}(\boldsymbol{C})\,\| \\ [v\, \boldsymbol{a}\, v|v]_W\, \boldsymbol{C}\end{array}}}} \quad ;$$

  we take $V = U \cup W$; the condition on variables being maximally renamed apart ensures that $U \cap \underline{\boldsymbol{C}} = \emptyset = W \cap \underline{\boldsymbol{B}}$, therefore $[v\, \boldsymbol{a}\, v|v]_V\, \boldsymbol{A} \equiv [v\, \boldsymbol{a}\, v|v]_U\, \boldsymbol{B}\, \beta\, [v\, \boldsymbol{a}\, v|v]_W\, \boldsymbol{C}$.
- If $\boldsymbol{A} \equiv \langle \boldsymbol{B}|x\rangle\, \boldsymbol{C}$ then we build

$$\chi_{\boldsymbol{a}}\boldsymbol{A} \equiv \boxed{\begin{array}{c}\underbrace{\left\langle \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{B}\middle|x\right\rangle\, \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{C}\, \boldsymbol{a}\, \left\langle \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{B}\middle|x\right\rangle\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{C}}_{\left\langle \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{B}\middle|y\right\rangle\, \left\langle \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{B}\middle|z\right\rangle\ \boxed{\begin{array}{c}[y|x]\, \mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{C}\, \boldsymbol{a}\, [z|x]\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{C} \\ \chi'_{\boldsymbol{a}}(\boldsymbol{C})\,\| \\ [y\, \boldsymbol{a}\, z|x]\, [v\, \boldsymbol{a}\, v|v]_U\, \boldsymbol{C}\end{array}}} \\ \left\langle \begin{array}{l}\mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{B}\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{B} \\ \chi_{\boldsymbol{a}}(\boldsymbol{B})\,\| \\ [v\, \boldsymbol{a}\, v|v]_W\, \boldsymbol{B}\end{array}\middle| x\right\rangle\, [v\, \boldsymbol{a}\, v|v]_U\, \boldsymbol{C}\end{array}} \quad ,$$

  where, with y and z fresh variables, $\chi'_{\boldsymbol{a}}(\boldsymbol{C})$ is obtained from $\chi_{\boldsymbol{a}}(\boldsymbol{C})$ by renaming the occurrence of x in $\mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{C}$ by y and that of x in $\mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{C}$ by z and propagating the renamings; we take $V = U \cup W$ and the condition on variables being maximally renamed apart ensures that $U \cap \underline{\boldsymbol{C}} = \emptyset = W \cap \underline{\boldsymbol{B}}$, therefore $[v\, \boldsymbol{a}\, v|v]_V\, \boldsymbol{A} \equiv \langle [v\, \boldsymbol{a}\, v|v]_W\, \boldsymbol{B}|x\rangle\, [v\, \boldsymbol{a}\, v|v]_U\, \boldsymbol{C}$.

No cuts are generated in any step because all inference steps are of the the form $\beta\breve{\boldsymbol{a}}$, for β a connective in $\boldsymbol{A}$. □

Remark 8.2.14. By Remark 3.3.5, a proof in KDT-ODS cannot contain any free variable. Moreover, if $\boldsymbol{A} = 1$ then $\mathsf{l}_{\boldsymbol{a}}\, \boldsymbol{A}\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}}\, \boldsymbol{A} = 1$.

We can eliminate cuts by building a proof that explores all the assignments of truth values to atoms, like in a truth table. Each assignment generates two projections, and the projections are kept together by the atom connective corresponding to the atom being tested. The order by which values are assigned is arbitrary.

Theorem 8.2.15 (Cut Elimination)**.** *In* KDT-ODS*, for every regular proof of* $\boldsymbol{A}$ *we can build a cut-free proof of* $\boldsymbol{A}'$*, such that* $\boldsymbol{A} = \boldsymbol{A}'$*.*

Proof. Given a KDT-ODS proof ϕ_0, we extract all the units into a substitution π_0, such that $\phi_0 \equiv \pi_0\psi_0'$ and the variables in ψ_0' are maximally renamed apart. We enumerate $\boldsymbol{a}_1, \cdots \boldsymbol{a}_n$ those atoms on which there is a cut in ψ_0. By Proposition 8.2.13 we build

$$\phi_1 \equiv \boxed{\begin{array}{c} \mathsf{l}_{\boldsymbol{a}_1}\, \psi_0\, \boldsymbol{a}\, \mathsf{r}_{\boldsymbol{a}_1}\, \psi_0 \\ \chi_{\boldsymbol{a}_1}(\mathsf{cn}\, \psi_0) \,\Big\|\, \mathsf{KDT\text{-}ODS} \\ [v\ \boldsymbol{a}_1\ v|v]_{V_0}\, \mathsf{cn}\, \psi_0 \end{array}} \quad .$$

We make $\psi_1 \equiv \pi_1\phi_1$ such that all its variables are maximally renamed apart and we repeat the construction on ψ_1, ψ_2 and so on until we obtain ϕ_n. The derivation $\phi \equiv \pi_0\pi_1 \cdots \pi_{n-1}\phi_n$ is cut-free, by Remark 8.2.4, and $\mathsf{cn}\, \phi = \mathsf{cn}\, \phi_0$. By Remark 8.2.14, we have that $\mathsf{pr}\, \phi = 1$. □

Corollary 8.2.16. *For every proof* ϕ *of* A *in* SKS-OD*, we can build a cut-free proof* ϕ' *of* $\boldsymbol{A}'$ *in* KDT-ODS *such that the interpretation of* ϕ' *into* SKS-OD *is structurally equivalent to* ϕ *and the interpretation* $\boldsymbol{A}'$ *into* SKS *is equal to* A*.*

Proof. By Corollary 8.2.12, we can construct a corresponding regular proof in KDT-ODS; by Theorem 8.2.15, we can eliminate the cuts from that proof to obtain a cut-free proof ϕ' of a suitable $\boldsymbol{A}'$. We then apply the abstraction map kdt followed by the standard interpretation map given in [2] to ϕ'. □

This procedure for cut elimination does not make any use of the explicit substitutions within the system to compress the resulting cut-free proof. An interesting avenue for future work would be to evaluate and compare the compression afforded by cuts and explicit substitutions, following the line

of [27] and [35] in which it is shown that the deep inference formalism of the *calculus of structures* allows for a cut-free system with substitutions to be directly compared to a system with cuts and without substitutions, and speculated that substitutions may be as powerful as the cut.

Chapter 9

Conclusions

We believe that the work presented here brings us one step closer to the objective of having a good semantics of proofs. One essential test for that semantics will be its ability to express a non-degenerate and useful notion of *identity of proofs* (which is known as Hilbert's 25th problem [36]). The reason we think that this work contributes is that we envisage the following progression:

1. Based on the strictly linear proof systems and supersubstitution developed in this thesis, we define a notion of substitution for standard (deep-inference) proof systems that works with structural rules and, especially, identity-cut cycles.

2. We extend explicit substitution to the first order, in particular, we deal with quantification as a special case of substitution (there is preliminary, unpublished research about that).

3. We attempt to define the identity of two proofs as their being factorisable to the same proof via that notion of substitution (plus some equality relation decidable in polynomial time).

This is our highest priority for future research.

Bibliography

[1] Martin Abadi, Luca Cardelli, P-L Curien, and J-J Lévy. Explicit substitutions. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 31–46, 1989.

[2] Andrea Aler Tubella. *A Study of Normalisation Through Subatomic Logic*. PhD thesis, University of Bath, 2017.

[3] Andrea Aler Tubella and Alessio Guglielmi. Subatomic proof systems: Splittable systems. *ACM Transactions on Computational Logic*, 19(1):5:1–33, 2018.

[4] 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.

[5] Chris Barrett and Alessio Guglielmi. A subatomic proof system for decision trees. *ACM Transactions on Computational Logic*, 23(4):26:1–25, 2022.

[6] Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):14:1–34, 2009.

[7] 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.

[8] Kai Brünnler. Two restrictions on contraction. *Logic Journal of the IGPL*, 11(5):525–529, 2003.

[9] Kai Brünnler. Locality for classical logic. *Notre Dame Journal of Formal Logic*, 47(4):557–580, 2006.

[10] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. Technical Report WV-01-02, Technische Universität Dresden, 2001.

[11] 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.

[12] Samuel R. Buss. Some remarks on lengths of propositional proofs. *Archive for Mathematical Logic*, 34(6):377–394, 1995.

[13] Samuel R. Buss. Propositional proof complexity – An introduction. In Ulrich Berger and Helmut Schwichtenberg, editors, *Computational Logic*, volume 165 of *NATO ASI series. Series F. Computer and Systems Sciences*, pages 127–178. Springer, 1999.

[14] Peter Clote and Evangelos Kranakis. *Boolean Functions and Computation Models*. Springer, 2002.

[15] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[16] Gerhard Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.

[17] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

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

[19] Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1):9:1–36, 2008.

[20] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A proof calculus which reduces syntactic bureaucracy. In Christopher Lynch, editor, *21st International Conference on Rewriting Techniques and Applications (RTA)*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 135–150. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.

[21] 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.

[22] 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.

[23] Yasuyuki Imai and Kiyoshi Iséki. On axiom systems of propositional calculi. I. *Proceedings of the Japan Academy*, 41(6):436–439, 1965.

[24] Emil Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 19(2):323–339, 2009.

[25] Delia Kesner. A theory of explicit substitutions with safe and full composition. *Logical Methods in Computer Science*, Volume 5, Issue 3, July 2009.

[26] Jan Łukasiewicz. *Elements of Mathematical Logic*, volume 31 of *Pure and Applied Mathematics*. Pergamon Press, 1963.

[27] Novak Novaković and Lutz Straßburger. On the power of substitution in the calculus of structures. *ACM Transactions on Computational Logic*, 16(3):19:1–20, 2015.

[28] M.S. Paterson and M.N. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16(2):158–167, 1978.

[29] Detlef Plump. Term graph rewriting. In Hartmut Ehrig, Gregor Engels, Hans-Joerg Kreowski, and Grzegorz Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation, Volume*

2: Applications, Languages and Tools, volume 2, pages 3–61. World Scientific Publishing, Singapore, 1999.

[30] Benjamin Ralph. *Modular Normalisation of Classical Proofs*. PhD thesis, University of Bath, 2019.

[31] Alessio Santamaria. *Towards a Godement calculus for dinatural transformations*. PhD thesis, University of Bath, Somerset, UK, 2019.

[32] Charles Stewart and Phiniki Stouppa. A systematic proof theory for several modal logics. In Renate Schmidt, Ian Pratt-Hartmann, Mark Reynolds, and Heinrich Wansing, editors, *Advances in Modal Logic (AiML)*, volume 5, pages 309–333. King's College Publications, 2005.

[33] Lutz Straßburger. A local system for linear logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2514 of *Lecture Notes in Computer Science*, pages 388–402. Springer, 2002.

[34] Lutz Straßburger. From deep inference to proof nets via cut elimination. *Journal of Logic and Computation*, 21(4):589–624, 2011.

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

[36] Rüdiger Thiele. Hilbert's twenty-fourth problem. *American Mathematical Monthly*, 110:1–24, 2003.

[37] 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.

[38] Ingo Wegener. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2000.