Journal of Applied Logic 10 (2012) 330–367

Contents lists available at SciVerse ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Cut elimination for a logic with induction and co-induction

Alwen Tiu [a,*], Alberto Momigliano [b]

[a] Research School of Computer Science, The Australian National University, Australia
[b] Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Italy

ARTICLE INFO

Article history:
Received 1 April 2010
Accepted 26 July 2012
Available online 31 July 2012

Keywords:
Logical frameworks
(Co-)induction
Higher-order abstract syntax
Cut-elimination
Parametric reducibility

ABSTRACT

Proof search has been used to specify a wide range of computation systems. In order to build a framework for reasoning about such specifications, we make use of a sequent calculus involving induction and co-induction. These proof principles are based on a proof-theoretic (rather than set-theoretic) notion of *definition* (Hallnäs, 1991 [18], Eriksson, 1991 [11], Schroeder-Heister, 1993 [38], McDowell and Miller, 2000 [22]). Definitions are akin to logic programs, where the left and right rules for defined atoms allow one to view theories as "closed" or defining fixed points. The use of definitions and free equality makes it possible to reason intensionally about syntax. We add in a consistent way rules for pre- and post-fixed points, thus allowing the user to reason inductively and co-inductively about properties of computational system making full use of higher-order abstract syntax. Consistency is guaranteed via cut-elimination, where we give a direct cut-elimination procedure in the presence of general inductive and co-inductive definitions via the parametric reducibility technique.

1. Introduction

A common approach to specifying computations is via deductive systems. Those are used to specify and reason about various logics, as well as aspects of programming languages such as operational semantics, type theories, abstract machines, etc. Such specifications can be represented as logical theories in a suitably expressive formal logic where *proof-search* can then be used to model the computation. A logic used as a specification language is known as a *logical framework* [33], which comes equipped with a representation methodology. The encoding of the syntax of deductive systems inside formal logic can benefit from the use of *higher-order abstract syntax* (HOAS), a high-level and declarative treatment of object-level bound variables and substitution. At the same time, we want to use such a logic to reason over the *meta-theoretical* properties of object languages, e.g., type preservation in operational semantics [23], soundness and completeness of compilation [29] or congruence of bisimulation in transition systems [24]. Typically this involves reasoning by (structural) induction and, when dealing with infinite behavior, co-induction [20].

The need to support both inductive and co-inductive reasoning and some form of HOAS requires some careful design decisions, since the two are prima facie notoriously incompatible. While any meta-language based on a λ-calculus can be used to specify and animate HOAS encodings, meta-reasoning has traditionally involved (co-)inductive specifications both at the level of the syntax and of the judgments – which are of course unified at the type-theoretic level. Syntax-level HOAS provides crucial freeness properties for (object-level) datatype constructors, while judgment-level HOAS offers principles of case analysis and (co-)induction. The latter is known to be problematic, since HOAS specifications may lead to non-monotone (co-)inductive operators, which by cardinality and consistency reasons are not permitted in inductive logical frameworks.

* Corresponding author. Tel.: +61 02 6125 5992; fax: +61 02 6125 8645.
E-mail addresses: Alwen.Tiu@anu.edu.au (A. Tiu), momigliano@dsi.unimi.it (A. Momigliano).

1570-8683/$ – see front matter
http://dx.doi.org/10.1016/j.jal.2012.07.007

Moreover, even when HOAS is weakened [10] so as to be made compatible with proof assistants such as HOL or Coq, weak HOAS suffers the fate of allowing the existence of too many functions and yielding the so-called *exotic* terms. Those are canonical terms in the signature of an HOAS encoding that do not correspond to any term in the deductive system under study. This causes a loss of adequacy in specifications, which is one of the pillars of formal verification, and it undermines the trust in formal derivations. On the other hand, logics such as LF [19] that are weak by design in order to support this style of syntax are not directly endowed with (co-)induction principles.

In a previous paper [30] we have introduced the logic Linc$^-$, for a logic with λ-terms, induction and co-induction,[1] which carefully adds principles of induction and co-induction to a first-order intuitionistic logic based on a proof-theoretic notion of *definitions*, following on work (among others) by Hallnäs [18], Eriksson [11], Schroeder-Heister [38] and McDowell and Miller [22]. Definitions are akin to logic programs, but allow us to view theories as "closed" or defining fixed points. This alone permits us to perform case analysis independently from induction principles. Our approach to formalizing induction and co-induction is via the least and greatest solutions of the fixed point equations specified by the definitions. The proof rules for induction and co-induction make use of the notion of *pre-fixed points* and *post-fixed points* respectively. In the inductive case, this corresponds to the induction invariant, while in the co-inductive one to the so-called simulation. Judgments are encoded as definitions accordingly to their informal semantics, either inductive or co-inductive. The simply typed language and the notion of free equality underlying Linc$^-$, enforced via (higher-order) unification in an inference rule, make it possible to reason *intensionally* about syntax. In fact, we can support HOAS encodings of object-level constants and we can *prove* the freeness properties of those constants, namely injectivity, distinctness and case exhaustion. Linc$^-$ is a conservative extension of $FO\lambda^{\Delta\mathbb{N}}$ [22] and a generalization (with a term language based on simply typed λ-calculus) of Martin-Löf first-order theory of iterated inductive definitions [21].

The contribution of this paper is to give a *direct* proof of cut-elimination for such a logic, based on the robust and highly configurable *reducibility-candidate* technique, which dates back to Girard's strong normalization proof for System F [17]. Beyond its intrinsic interest, it is our hope that the cut elimination proof presented in this paper may be used as a springboard towards cut elimination procedures for more expressive (conservative) extensions of Linc$^-$ with the ∇-quantifier, along the lines of [28,41,42,15].

We note that there are other *indirect* methods to show the admissibility of cut in a logic with (co-)inductive definitions, viz.:

1. Encoding (co-)inductive definitions as higher-order formulae. This approach is followed by Baelde and Miller [5] where μMALL, a linear fixed-point logic related to Linc$^-$, is introduced. Cut elimination is proved indirectly by encoding fixed points into second-order linear logic and conjecturing cut-elimination for an extension of LL2.
2. Model-theoretically: this is taken by Brotherston and Simpson [8], which provide a model for a classical first-order logic with inductive definitions: here, cut admissibility follows by the semantical completeness of the cut free fragment.

In particular, one crucial, and orthogonal, extension to Linc$^-$ is the addition of the ∇-quantifier [28,41,42,15], which allows one to reason about the intensional aspects of *names and bindings* in object syntax specifications (see, e.g., [14,43]). However, both aforementioned approaches do not seem to cope effectively with ∇: while the behavior of the quantifier can be operationally simulated in a Linc$^-$-like logic, see the "eigenvariable" encoding of [23], a direct encoding in linear logic is presently unknown. Similarly, it is not obvious how the model-theoretic way can be taken, while the semantics of ∇ itself is not yet fully understood, although there have been some recent attempts, see [26,36].

The present paper is an extended and revised version of [30]. Because of the significant span of time elapsed, we feel it is useful to place the present paper in the time-line of closely related work (more details can be found in Section 7). In the conference paper [30], the co-inductive rule had a technical side condition requiring the given predicate to be used "monotonically" in the simulation. This somewhat ad hoc restriction was imposed by the particular cut elimination proof technique outlined in that paper. That restriction was firstly removed in the 2008 version of the paper [44], thanks to the adoption of the *parametric* reducibility technique. Still, (co-)inductive definitions had to be stratified. This is the argument that Gacek builds on for the cut elimination proof for $\mathcal{G}$ in his dissertation [13], although it is fair to say that not every detail is fully ironed out. In particular, the definition of parametric reducibility is taken "as is" and not generalized w.r.t. nominal abstraction. Simultaneously, and to a great extent independently, Baelde [4] gave his Girard-like proof of cut-elimination for μMALL. To our knowledge, this is the only other work that uses the candidate-reducibility technique directly in the sequent calculus setting. In our paper, the formulation of the rules is inspired by the second-order encoding of least and greatest fixed points used in [5] and, remarkably, neither stratification nor monotonicity is required.

The rest of the paper is organized as follows. Section 2 introduces the sequent calculus for the logic Linc$^-$. Every sequent in Linc$^-$ carries a typing context for eigenvariables, called the *signature* of the sequent. This allows us to be technically more precise in particular accounting for empty types in the logic. However, the presence of signatures complicates the definition of reducibility (candidates) needed to prove cut-elimination. We therefore prove cut elimination for Linc$^-$ by a detour through a system with no signatures (with the consequence that types are not allowed to be empty), called Linc$_i^-$, presented in Section 3. Section 4 presents two transformations of derivations in Linc$_i^-$ that are essential to the cut reduction rules and

[1] The "minus" in the terminology refers to the lack of the ∇-quantifier w.r.t. the eponymous logic in Tiu's thesis [41].

$$\frac{}{\Sigma; C \longrightarrow C}\ init \qquad \frac{\Sigma; B, B, \Gamma \longrightarrow C}{\Sigma; B, \Gamma \longrightarrow C}\ \mathrm{c}\mathcal{L} \qquad \frac{\Sigma; \Gamma \longrightarrow C}{\Sigma; B, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}$$

$$\frac{\Sigma; \Delta_1 \longrightarrow B_1 \quad \cdots \quad \Sigma; \Delta_n \longrightarrow B_n \quad \Sigma; B_1, \ldots, B_n, \Gamma \longrightarrow C}{\Sigma; \Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc, \quad \text{where } n \geqslant 0$$

$$\frac{}{\Sigma; \bot \longrightarrow B}\ \bot\mathcal{L} \qquad \frac{}{\Sigma; \Gamma \longrightarrow \top}\ \top\mathcal{R}$$

$$\frac{\Sigma; B_i, \Gamma \longrightarrow D}{\Sigma; B_1 \wedge B_2, \Gamma \longrightarrow D}\ \wedge\mathcal{L}, \quad i \in \{1,2\} \qquad \frac{\Sigma; \Gamma \longrightarrow B \quad \Sigma; \Gamma \longrightarrow C}{\Sigma; \Gamma \longrightarrow B \wedge C}\ \wedge\mathcal{R}$$

$$\frac{\Sigma; B, \Gamma \longrightarrow D \quad \Sigma; C, \Gamma \longrightarrow D}{\Sigma; B \vee C, \Gamma \longrightarrow D}\ \vee\mathcal{L} \qquad \frac{\Sigma; \Gamma \longrightarrow B_i}{\Sigma; \Gamma \longrightarrow B_1 \vee B_2}\ \vee\mathcal{R}, \quad i \in \{1,2\}$$

$$\frac{\Sigma; \Gamma \longrightarrow B \quad \Sigma; C, \Gamma \longrightarrow D}{\Sigma; B \supset C, \Gamma \longrightarrow D}\ \supset\mathcal{L} \qquad \frac{\Sigma; B, \Gamma \longrightarrow C}{\Sigma; \Gamma \longrightarrow B \supset C}\ \supset\mathcal{R}$$

$$\frac{\Sigma; B\,t, \Gamma \longrightarrow C}{\Sigma; \forall_\tau x.B, \Gamma \longrightarrow C}\ \forall\mathcal{L} \qquad \frac{\Sigma, y_\tau; \Gamma \longrightarrow B\,y}{\Sigma; \Gamma \longrightarrow \forall_\tau x.B\,x}\ \forall\mathcal{R}, \quad y_\tau \notin \Sigma$$

$$\frac{\Sigma, y_\tau; B\,y, \Gamma \longrightarrow C}{\Sigma; \exists_\tau x.B\,x, \Gamma \longrightarrow C}\ \exists\mathcal{L}, \quad y \notin \Sigma \qquad \frac{\Sigma; \Gamma \longrightarrow B\,t}{\Sigma; \Gamma \longrightarrow \exists_\tau x.B\,x}\ \exists\mathcal{R}$$

Equality rules

$$\frac{\{\Sigma[\rho]; \Gamma[\rho] \longrightarrow C[\rho]\}_{\rho \in \mathbb{U}(s,t)}}{\Sigma; s = t, \Gamma \longrightarrow C}\ \mathrm{eq}\,\mathcal{L} \qquad \frac{}{\Sigma; \Gamma \longrightarrow t = t}\ \mathrm{eq}\,\mathcal{R}$$

Induction rules

$$\frac{\vec{y}; B\,S\,\vec{y} \longrightarrow S\,\vec{y} \quad \Sigma; \Gamma, S\,\vec{t} \longrightarrow C}{\Sigma; \Gamma, p\,\vec{t} \longrightarrow C}\ \mathrm{I}\mathcal{L}, \quad p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$$

$$\frac{\Sigma; \Gamma \longrightarrow B\,X^p\,\vec{t}}{\Sigma; \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}, \quad p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x} \qquad \frac{\Sigma; \Gamma \longrightarrow B\,X^p\,\vec{t}}{\Sigma; \Gamma \longrightarrow X^p\,\vec{t}}\ \mathrm{I}\mathcal{R}_p, \quad p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$$

Co-induction rules

$$\frac{\Sigma; B\,X^p\,\vec{t}, \Gamma \longrightarrow C}{\Sigma; p\,\vec{t}, \Gamma \longrightarrow C}\ \mathrm{CI}\,\mathcal{L}, \quad p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x} \qquad \frac{\Sigma; B\,X^p\,\vec{t}, \Gamma \longrightarrow C}{\Sigma; X^p\,\vec{t}, \Gamma \longrightarrow C}\ \mathrm{CI}\,\mathcal{L}_p, \quad p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$$

$$\frac{\Sigma; \Gamma \longrightarrow S\,\vec{t} \quad \vec{y}; S\,\vec{y} \longrightarrow B\,S\,\vec{y}}{\Sigma; \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\,\mathcal{R}, \quad p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$$

Fig. 1. The inference rules of Linc$^-$.

the cut elimination proof in subsequent sections. Section 5 is the heart of the paper: we first (Section 5.1) give a (sub)set of reduction rules that transform a Linc$_i^-$ derivation ending with a cut rule to another Linc$_i^-$ derivation. The complete set of reduction rules can be found in Appendix A. We then introduce the crucial notions of *normalizability* (Section 5.2) and of *parametric reducibility* after Girard (Section 5.3). Detailed proofs of the main lemma related to reducibility candidates are in Appendix B. Cut elimination for Linc$_i^-$ is proved in detail in Section 5.4. Cut elimination of Linc$^-$ is then obtained by showing that (Section 6) if a derivation in Linc$_i^-$ can be "decorated" with signatures to obtain a Linc$^-$ derivation, then cut reductions preserve this decoration. Hence, the cut elimination procedure for Linc$_i^-$, when applied to Linc$^-$ derivations, terminates as well. Section 7 surveys related work and concludes the paper.

Finally, we remark that this paper is concerned *only* with the cut elimination proof of Linc$^-$. For examples and applications of Linc$^-$ and its extensions, we refer the interested reader to [41,6,15,14,43].

2. The logic Linc$^-$

The logic Linc$^-$ shares the core fragment of $FO\lambda^{\Delta\mathbb{N}}$, which is an intuitionistic version of Church's Simple Theory of Types. We shall assume that the reader is familiar with Church's simply typed λ-calculus (with both β and η rules), so we shall recall only the basic syntax of the calculus here. A simple type is either a *base type* or a compound type formed using the function-type constructor $\rightarrow$. Types are denoted by σ and τ. We assume an infinite set of typed variables, written x_σ, y_τ, etc. The syntax of λ-terms is given by the following grammar:

$$s, t ::= x_\tau \mid (\lambda x_\tau . t) \mid (s\ t).$$

We consider only well-typed terms in Church's simple type system. To simplify presentation, in the following we shall often omit the type index in variables and λ-abstraction. The notion of free and bound variables are defined as usual. The set of free variables in a term t is denoted by $FV(t)$. This notation generalizes to formulae and multisets of formulae, e.g., $FV(B)$, $FV(\Gamma)$.

We assume a given finite set of base types, which contains a distinguished element o, denoting the type of formulae. Following Church, we shall represent formulae as well-typed λ-terms of type o. We assume a set of typed constants that correspond to logical connectives. The constants $\top : o$ and $\bot : o$ denote 'true' and 'false', respectively. Propositional binary connectives, i.e., $\wedge$, $\vee$, and $\supset$, are assigned the type $o \rightarrow o \rightarrow o$. Quantifiers are represented by indexed families of constants: $\forall_\tau$ and $\exists_\tau$, both are of type $(\tau \rightarrow o) \rightarrow o$. We also assume a family of typed equality symbols $=_\tau : \tau \rightarrow \tau \rightarrow o$. Although we adopt a representation of formulae as λ-terms, we shall use a more traditional notation when writing down formulae. For example, instead of writing $(\wedge\ A\ B)$, we shall use an infix notation $(A \wedge B)$. Similarly, we shall write $\forall_\tau x.P$ instead of $\forall_\tau\ (\lambda x_\tau.P)$. Again, we shall omit the type annotations in quantifiers and equality when they can be inferred from the context of the discussion.

The type τ in quantifiers and the equality predicate are restricted to those simple types that do not contain occurrences of o. Hence our logic is essentially first-order, since we do not allow quantification over predicates. As we shall often refer to this kind of restriction to types, we give the following definition:

Definition 1. A simple type τ is *essentially first-order* (efo) if it is generated by the following grammar:

$$\tau ::= k \mid \tau \rightarrow \tau$$

where k is a base type other than o.

From now on, we shall use τ exclusively for efo-types.

For technical reasons (to present (co-)inductive proof rules, see Section 2.2), we introduce a notion of *parameter* into the syntax of formulae. Intuitively, they play the role of eigenvariables ranging over the recursive call in a fixed point expression. To each predicate symbol p, we associate a countably infinite set $\mathcal{P}_p$, called the *parameter set for p*. Elements of $\mathcal{P}_p$ are denoted by X^p, Y^p, Z^p, etc., and have the same type as p. When we refer to formulae of Linc$^-$, we have in mind simply-typed λ-terms of type o *in $\beta\eta$-long normal form*. Thus formulae of Linc$^-$ can be equivalently defined via the following grammar:

$$F ::= X^p\,\vec{t} \mid s =_\tau t \mid p\,\vec{t} \mid \bot \mid \top \mid F \wedge F \mid F \vee F \mid F \supset F \mid \forall_\tau x.F \mid \exists_\tau x.F.$$

A *substitution* is a type-preserving mapping from variables to terms. We assume the usual notion of capture-avoiding substitutions, denoted by lower-case Greek letters, e.g., θ, ρ and σ. Application of substitution is written in postfix notation, e.g. $t[\theta]$ denotes the term resulting from an application of θ to t. This notation is generalized to applications of a substitution to a formula, e.g., $B[\theta]$ or a multiset of formula, e.g., $\Gamma[\theta]$. To simplify presentation, we write $t[\theta][\sigma]$ to mean $(t[\theta])[\sigma]$. Composition of substitutions, denoted by $\circ$, is defined as $t[\theta \circ \rho] = t[\theta][\rho]$. We use ϵ to denote the identity substitution, i.e., $\epsilon(x) = x$ for every variable x. We denote with $\mathbb{S}$ the set of all substitutions. Given a pair of terms s and t, a *unifier* of (s, t) is a substitution θ such that $s[\theta] = t[\theta]$. We denote with $\mathbb{U}(s, t)$ the set of all unifiers of s and t. Any substitution is a unifier of two identical terms, so obviously, we have $\mathbb{U}(t, t) = \mathbb{S}$.

The sequent calculus of Linc$^-$ is given in Fig. 1. A sequent is denoted by $\Sigma; \Gamma \rightarrow C$ where C is a formula and Γ is a multiset of formulae, each in $\beta\eta$-long normal form, and Σ is a *signature*, i.e., a set of eigenvariables, such that $FV(\Gamma \cup \{C\}) \subseteq \Sigma$. Notice that in the presentation of the rule schemes, we make use of HOAS, e.g., in the application $B\,x$ it is implicit that B has no free occurrence of x. Similarly for the (co-)induction rules. We work modulo α-conversion without further notice. In the $\forall\mathcal{R}$ and $\exists\mathcal{L}$ rules, y is an eigenvariable (of type τ) that is not already in Σ. In the $\exists\mathcal{R}$ and $\forall\mathcal{L}$ rules, t is a term of type τ. The *init* rule can be restricted to the atomic form, i.e., where C is either $(p\,u_1 \ldots u_n)$ or $(X^p\,u_1 \ldots u_n)$. However, we keep the more general form as it simplifies the definition of a certain transformation of derivations (see Definition 8) used in the cut elimination proof. The *mc* rule is a generalization of the cut rule that simplifies the presentation of the cut-elimination proof for intuitionistic systems, i.e., to deal with the case involving permutation of cut over the contraction rule; see [39,22] and the reduction rule $-/c\mathcal{L}$ in Appendix A.

The rules for equality and fixed points are discussed in Sections 2.1 and 2.2, respectively.

2.1. Equality

The right introduction rule for equality is reflexivity, that is, it recognizes that two terms are syntactically equal. The left introduction rule is more interesting. The substitution ρ in eq$\mathcal{L}$ is a unifier of s and t. Note that we specify the premise of eq$\mathcal{L}$ as a set, with the intention that every sequent in the set is a premise of the rule. This set is of course infinite; every unifier of (s,t) can be extend to another one (e.g., by adding substitution pairs for variables not in the terms). However, in many cases, it is sufficient to consider a particular set of unifiers, which is often called a *complete set of unifiers (CSU)* [2], from which any unifier can be obtained by composing a member of the CSU set with a substitution. In the case where the terms are first-order terms, or higher-order terms with the pattern restriction [27], the set CSU is either empty or a singleton. In eq$\mathcal{L}$, the signature $\Sigma[\rho]$ in the premises is defined as follows:

$$\Sigma[\rho] = \bigcup \big\{ FV\big(x[\rho]\big) \bigm| x \in \Sigma \big\}.$$

Our rules for equality actually encompasses the notion of *free equality* as commonly found in logic programming, in the form of Clark's equality theory [9]: injectivity of function symbols, inequality between distinct function symbols, and the "occurs-check" follow from the eq$\mathcal{L}$-rule. For instance, given a base type nt (for natural numbers) and the constants $z : nt$ (zero) and $s : nt \to nt$ (successor), we can derive $\forall x.\ z = (s\ x) \supset \bot$ as follows:

$$\dfrac{\dfrac{\dfrac{}{y; z = (s\ y) \longrightarrow \bot}\ \mathrm{eq}\mathcal{L}}{y; \cdot \longrightarrow z = (s\ y) \supset \bot}\ \supset\mathcal{R}}{\cdot \longrightarrow \forall x.\ z = (s\ x) \supset \bot}\ \forall\mathcal{R}.$$

Since z and $s\ y$ are not unifiable, the eq$\mathcal{L}$ rule above has empty premise, thus concluding the derivation. A similar derivation establishes the occur-check property, e.g., $\forall x.\ x = (s\ x) \supset \bot$. We can also prove the injectivity of the successor function, i.e. $\forall x \forall y.(s\ x) = (s\ y) \supset x = y$.

This proof-theoretic notion of equality has been considered in several previous works, e.g. by Schroeder-Heister [38], and McDowell and Miller [22].

2.2. Induction and co-induction

One way of adding induction and co-induction to a logic is to introduce fixed point expressions and their associated introduction and elimination rules, i.e. using the μ and ν operators of the (first-order) μ-calculus. This is essentially what we shall follow here, but with a different notation. Instead of using a "nameless" notation with μ and ν to express fixed points, we associate a fixed point equation with an atomic formula. That is, we associate certain designated predicates with a *definition*. This notation is clearer and more convenient as far as our applications are concerned. For a proof system using nameless notations for (co-)inductive predicates, the interested reader is referred to [5,4].

Definition 2. An *inductive definition clause* is written $\forall \vec{x}.\ p\,\vec{x} \stackrel{\mu}{=} B\,\vec{x}$, where p is a predicate constant. The atomic formula $p\,\vec{x}$ is the *head* of the clause, and the formula $B\,\vec{x}$, where B is a closed term containing no occurrences of parameters, is the *body*. Similarly, a *co-inductive definition clause* is written $\forall \vec{x}.\ p\,\vec{x} \stackrel{\nu}{=} B\,\vec{x}$. The symbols $\stackrel{\mu}{=}$ and $\stackrel{\nu}{=}$ are used simply to indicate a definition clause: they are not a logical connective. We shall write $\forall \vec{x}.\ p\,\vec{x} \stackrel{\triangle}{=} B\,\vec{x}$ to denote a definition clause generally, i.e., when we are not interested in the details of whether it is an inductive or a co-inductive definition. A *definition* is a finite set of definition clauses. A predicate may occur only at most once in the heads of the clauses of a definition. We shall restrict to *non-mutually recursive* definitions. That is, given two clauses $\forall \vec{x}.\ p\,\vec{x} \stackrel{\triangle}{=} B\,\vec{x}$ and $\forall \vec{y}.\ q\,\vec{y} \stackrel{\triangle}{=} C\,\vec{y}$ in a definition, where $p \neq q$, if p occurs in C then q does not occur in B, and vice versa.

One can encode mutual recursion between definitions of the same kind (inductive or co-inductive) with a single predicate with an extra argument. However, it is in general not possible to encode mixed mutual recursion, e.g., mutual recursion between an inductive and a co-inductive definitions, in Linc$^-$. In this sense, our notion of fixed points is more restricted than what could be encoded using μ and ν as in [4].

For technical convenience we also bundle up all the definitional clauses for a given predicate in a single clause, following the same principles of the *iff-completion* in logic programming [37]. Further, in order to simplify the presentation of rules that involve predicate substitutions, we denote a definition using an abstraction over predicates, that is

$$\forall \vec{x}.\ p\vec{x} \stackrel{\triangle}{=} B\ p\,\vec{x}$$

where B is an abstraction with no free occurrence of predicate symbol p and variables $\vec{x}$. Substitution of p in the body of the clause with a formula S can then be written simply as $B\,S\,\vec{x}$. When writing definition clauses, we omit the outermost universal quantifiers, with the assumption that free variables in a clause are universally quantified. For example even numbers are defined as:

$$ev\ x \stackrel{\mu}{=} (x = z) \vee \big(\exists y.\ x = \big(s\ (s\ y)\big) \wedge ev\ y\big)$$

where in this case B is of the form $\lambda p \lambda w.\ (w = z) \vee (\exists y.w = (s\ (s\ y)) \wedge p\ y)$.

The left and right rules for (co-)inductively defined atoms are given at the bottom of Fig. 1. In rules I$\mathcal{L}$ and CI$\mathcal{R}$, the abstraction S is an invariant of the (co-)induction rule. The variables $\vec{y}$ are new eigenvariables and X^p is a new parameter not occurring in the lower sequent. For the rule $\mathrm{I}\mathcal{L}$, S denotes a pre-fixed point of the underlying fixed point operator. Similarly, for the co-induction rule $\mathrm{CI}\mathcal{R}$, S can be seen as denoting a post-fixed point of the same operator. Here, we use a characterization of induction and co-induction proof rules as, respectively, the least and the greatest solutions to a fixed point equation.

Notice that the right-introduction rules for inductive predicates and the left-introduction rules for co-inductive predicates are slightly different from the corresponding rules in Linc-like logics (see Remark 1 in page 6). These rules can be better understood by the usual interpretation of (co-)inductive definitions in second-order logic [34,32] – for simplicity, we show only the propositional case here:

$$p \stackrel{\mu}{=} B\,p \quad \rightsquigarrow \quad \forall p.(B\,p \supset p) \supset p,$$
$$p \stackrel{\nu}{=} B\,p \quad \rightsquigarrow \quad \exists p.p \wedge (p \supset B\,p).$$

Then the right-introduction rule for inductively defined predicates will involve an implicit universal quantification over predicates. As standard in sequent calculus, such a universal quantified predicate will be replaced by a new eigenvariable (in this case, a new parameter), reading the rule bottom up. Note that if we were to follow the above second-order interpretation literally, an alternative rule for inductive predicates could be:

$$\frac{B\,X^p \supset X^p, \Gamma \longrightarrow X^p}{\Gamma \longrightarrow p}\ \mathrm{I}\mathcal{R}, \quad p \stackrel{\mu}{=} B\,p.$$

Then there would be no need to add the $\mathrm{I}\mathcal{R}_p$-rule since it would be derivable, using the clause $B\,X^p \supset X^p$ in the left-hand side of the sequent. Our presentation has the advantage that it simplifies the cut-elimination argument in the subsequent sections. The left-introduction rule for co-inductively defined predicates can be explained dually.

Remark 1 *(Fixed point unfolding).* A commonly used form of introduction rules for definitions and fixed points uses an unfolding of the definitions. This kind of rules is followed in several related logics, e.g., $FO\lambda^{\Delta\mathbb{N}}$ [22], Linc [30,41] and μMALL [5]. The right-introduction rule for inductive definitions, for instance, takes the form:

$$\frac{\Sigma; \Gamma \longrightarrow B\,p\,\vec{t}}{\Sigma; \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}', \quad p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}.$$

The logic Linc, like $FO\lambda^{\Delta\mathbb{N}}$, imposes a stratification on definitions, which amounts to a strict positivity condition: the head of a definition does not appear to the left of an implication. Let us call such a definition a *stratified definition*. For stratified definitions, the rule $\mathrm{I}\mathcal{R}'$ can be derived in Linc^- as follows:

$$\dfrac{\Sigma; \Gamma \longrightarrow B\,p\,\vec{t} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vec{x}; B\,X^p\,\vec{x} \longrightarrow B\,X^p\,\vec{x}}\ init}{\vec{x}; B\,X^p\,\vec{x} \longrightarrow X^p\,\vec{x}}\ \mathrm{I}\mathcal{R}_p \qquad \dfrac{}{\Sigma'; X^p\,\vec{u} \longrightarrow X^p\,\vec{u}}\ init}{\Sigma'; p\,\vec{u} \longrightarrow X^p\,\vec{u}}\ \mathrm{I}\mathcal{L}}{\vdots}}{\Sigma; B\,p\,\vec{t} \longrightarrow B\,X^p\,\vec{t}}}{\Sigma; B\,p\,\vec{t} \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}}{\Sigma; \Gamma \longrightarrow p\,\vec{t}}\ mc$$

where the 'dots' denote a derivation composed using left and right introduction rules for logical connectives in B. Notice that all leaves of the form $\Sigma'; p\,\vec{u} \longrightarrow X^p\,\vec{u}$ can be proved by using the $\mathrm{I}\mathcal{L}$ rule, with X^p as the inductive invariant. This suggests that when restricted to stratified definitions, any formula (not containing ∇) provable in Linc is also provable in Linc^-. Conversely, given a stratified definition, any proof in Linc^- using that definition can be transformed into a proof of Linc simply by replacing X^p with p.

A dual argument applies to co-inductive definitions, of course.

Since a defined atomic formula can be unfolded via its introduction rules, the definition of the *size* of a formula needs to take into account this possible unfolding. This is done by assigning a positive integer to each predicate symbol, which we refer to as its *level*. A similar notion of level of a predicate was introduced for $FO\lambda^{\Delta\mathbb{N}}$ [22]. However, in $FO\lambda^{\Delta\mathbb{N}}$, the level of a predicate is only used to guarantee *monotonicity* of definitions.

Definition 3 *(Size of formulae).* To each predicate p we associate a natural number $\mathrm{lvl}(p)$, the *level* of p. Given a formula B, its *size* $|B|$ is defined as follows:

1. $|X^p\,\vec{t}| = 1$, for any X^p and any $\vec{t}$.
2. $|p\,\vec{t}| = \mathrm{lvl}(p)$.

3. $|\bot| = |\top| = |(s = t)| = 1$.
4. $|B \wedge C| = |B \vee C| = |B \supset C| = |B| + |C| + 1$.
5. $|\forall x.\ B\,x| = |\exists x.\ B\,x| = |B\,x| + 1$.

Note that in this definition, we do not specify precisely any particular level assignment to predicates. We show next that there is a level assignment that has a property that will be useful later in proving cut elimination, see Definition 15 and Lemma 14.

Lemma 1 *(Level assignment). Given any definition $\mathcal{D}$, there is a level assignment to every predicate p occurring in $\mathcal{D}$ such that if $\forall\vec{x}.\ p\,\vec{x} \stackrel{\triangle}{=} B\,p\,\vec{x}$ is in $\mathcal{D}$, then $|p\,\vec{x}| > |B\,X^p\,\vec{x}|$ for every parameter $X^p \in \mathcal{P}_p$.*

Proof. Let $\prec$ be a binary relation on predicate symbols defined as follows: $q \prec p$ iff q occurs in the body of the definition clause for p. Let $\prec^*$ be the reflexive-transitive closure of $\prec$. Since we restrict to non-mutually recursive definitions and there are only finitely many definition clauses (Definition 2), it follows that $\prec^*$ is a well-founded partial order. We now compute a level assignment to predicate symbols by induction on $\prec^*$. This is simply done by letting $\mathrm{lvl}(p) = 1$, if p is undefined, and $\mathrm{lvl}(p) = |B\,X^p\,\vec{x}| + 1$, for some parameter X^p, if $\forall\vec{x}.\ p\,\vec{x} \stackrel{\triangle}{=} B\,p\,\vec{x}$. Note that in the latter case, by induction hypothesis, every predicate symbol q, other than p, in B has already been assigned a level, so $|B\,X^p\,\vec{x}|$ is already defined at this stage. $\square$

We shall assume from now on that predicates are assigned levels satisfying the condition of Lemma 1, so whenever we have a definition clause of the form $\forall\vec{x}.p\,\vec{x} \stackrel{\triangle}{=} B\,p\,\vec{x}$, we implicitly assume that $|p\,\vec{x}| > |B\,X^p\,\vec{x}|$ for every $X^p \in \mathcal{P}_p$.

Remark 2 *(Non-monotonicity).* $FO\lambda^{\Delta\mathbb{N}}$ uses a notion of stratification to rule out non-monotone, or in Halnäs' terminology [18] *partial*, definitions such as $p \stackrel{\triangle}{=} p \supset \bot$, for which cut-elimination is problematic.[2] In fact, from the above definition both p and $p \supset \bot$ are provable, but there is no direct proof of $\bot$. This can be traced back to the fact that unfolding of definitions in Linc and $FO\lambda^{\Delta\mathbb{N}}$ is allowed on both the left and the right-hand side of sequent. In Linc$^-$, inconsistency does not arise even allowing a non-monotone definition as the above, due to the fact that arbitrary unfolding of fixed points is not permitted. On the other hand, in Linc$^-$ one cannot reason about some well-founded inductive definitions which are not stratified. For example, consider the non-stratified definition:

$$\forall x.\ ev\ x \stackrel{\mu}{=} (x = z) \vee \big(\exists y.x = (s\ y) \wedge (ev\ y \supset \bot)\big).$$

If this definition were interpreted as a logic program (with negation-as-failure), then its least fixed point is the set of even natural numbers. However, the above encoding in Linc$^-$ is incomplete with respect to this interpretation, since not all even natural numbers can be derived using the above definition. For example, $ev\ (s\ (s\ z))$ is not derivable, as this would require a derivation of $X^{ev}\,(s\ z) \longrightarrow \bot$, for some inductive parameter X^{ev}, which is impossible because no unfolding of inductive parameter is allowed on the left of a sequent. The same idea prevents the derivability of $\longrightarrow p$ given the definition $p \stackrel{\triangle}{=} p \supset \bot$. So while inconsistency in the presence of non-monotone definitions is avoided in Linc$^-$, its reasoning power does not extend that of Linc significantly.

3. Linc$_i^-$: A system with implicit signatures

To simplify the cut elimination proof of Linc$^-$, we shall define an intermediate proof system, called Linc$_i^-$, in which the signature is *implicit*. Thus sequents in Linc$_i^-$ take the form $\Gamma \longrightarrow C$. We prove cut elimination for Linc$_i^-$ in Section 5 and then show that cut admissibility for Linc$_i^-$ implies cut admissibility of Linc$^-$ in Section 6.

The rules of Linc$_i^-$ are those given in Fig. 1, but with all the signatures removed. Additionally, there are two distinctive features:

1. In $\exists\mathcal{R}$ and $\forall\mathcal{L}$ in Linc$_i^-$, the premise sequents may contain new eigenvariables not appearing in the conclusion sequents.
2. We add the following rule to Linc$_i^-$:

$$\frac{\{\Gamma[\theta] \longrightarrow C[\theta]\}_{\theta \in \mathbb{S}}}{\Gamma \longrightarrow C}\ subst.$$

Traditionally, many-sorted sequent calculi are presented with an implicit typing context. In this case, the typing context of a sequent in a derivation consists of the eigenvariables introduced below that sequent along the path to the root sequent.

[2] This phenomenon already appears in logics with definitional reflection [38], even before (co-)induction is considered. Other ways beyond stratification of recovering cut-elimination in those weaker logics are disallowing contraction or restricting to an *init* rule for undefined atoms.

Consider Π_1 and Π_2 below. If the type τ is empty, then the instance of $\exists\mathcal{R}$ in Π_1 is not allowed, whereas the instance of $\exists\mathcal{L}$ in Π_2 is

$$\Pi_1 = \dfrac{\dfrac{\overline{q\,x \longrightarrow q\,x}\; init}{\longrightarrow q\,x \supset q\,x}\supset\mathcal{R}}{\longrightarrow \exists_\tau y.(q\,y \supset q\,y)}\exists\mathcal{R}, \qquad \Pi_2 = \dfrac{\dfrac{\begin{matrix}\Pi_1\\ \longrightarrow \exists_\tau y.(q\,y \supset q\,y)\end{matrix}}{p\,x \longrightarrow \exists_\tau y.(q\,y \supset q\,y)}\mathrm{w}\mathcal{L}}{\exists_\tau x.\,p\,x \longrightarrow \exists_\tau y.(q\,y \supset q\,y)}\exists\mathcal{L}.$$

Such a non-local dependency of applicability of rules complicates the definition of reducibility and reducibility candidates (Section 5.3). We use, in fact, an inductive construction whereby reducibility of a derivation depends on reducibility of its subderivations. With respect to the example above, to define the reducibility of Π_2, we need first to define the reducibility of Π_1, which is not possible because it is not a valid derivation. So to be able to handle empty types, the notions of reducibility and reducibility candidates need to be parameterized by signatures, which would complicate further the already involved definitions.

To avoid doing that, we consider these slightly more flexible inference rules for $\exists\mathcal{R}$ and $\forall\mathcal{R}$ in Linc_i^-. Thus, both Π_1 and Π_2 are valid derivations in Linc_i^-. As a consequence, we can prove, for example, $\exists_\tau x.\top$ in Linc_i^-, for any type τ. In other words, we assume in Linc_i^- that all types are inhabited. Cut elimination for Linc^- is then proved indirectly via a detour through Linc_i^-, by showing that the validity of a Linc_i^- derivation is preserved by cut reductions; see Theorem 26.

In *subst*, every instance of the conclusion sequent, including the conclusion sequent itself, is a premise of the rule. This rule is just a 'macro' for the following derivation:

$$\dfrac{\dfrac{}{\longrightarrow t = t}\,\mathrm{eq}\mathcal{R} \qquad \dfrac{\{\Gamma[\theta] \longrightarrow C[\theta]\}_{\theta \in \mathbb{U}(t,t)}}{t = t, \Gamma \longrightarrow C}\,\mathrm{eq}\mathcal{L}}{\Gamma \longrightarrow C}\,mc$$

where t is some arbitrary term. The motivation behind the rule *subst* is purely technical; it allows us to prove that a derivation transformation commutes with cut reduction (see Lemma 7 and Remark 4).

Since the rule *subst* hides a simple form of cut, to prove cut-elimination of Linc_i^-, we have to show that *subst*, in addition to *mc*, is admissible.

Lemma 2 *(subst-elimination). If the sequent $\Gamma \longrightarrow C$ is (cut-free) derivable in* Linc_i^- *with subst then it is (cut-free) derivable in* Linc_i^- *without subst.*

Proof. Simply replace any subderivation of the form

$$\dfrac{\left\{\begin{matrix}\Pi^\theta\\ \Delta[\theta] \longrightarrow B[\theta]\end{matrix}\right\}_{\theta \in \mathbb{S}}}{\Delta \longrightarrow B}\,subst$$

with its premise Π^ϵ. □

4. Eigenvariables and parameters instantiations

We now discuss some properties of derivations in Linc_i^- which involve instantiations of eigenvariables and parameters. These properties will be used in the cut-elimination proof in subsequent sections.

Following [22], we define a *measure* which corresponds to the height of a derivation:

Definition 4. Given a derivation Π with premise derivations $\{\Pi_i\}_{i \in I}$, for some index set I, the measure $\mathrm{ht}(\Pi)$ is the least upper bound $\mathrm{lub}(\{\mathrm{ht}(\Pi_i)\}_{i \in I}) + 1$.

Given the possible infinite branching of the $\mathrm{eq}\mathcal{L}$ rule, this measure can in general be (countable) ordinals. Therefore proofs and definitions on this measure require transfinite induction and recursion. However, in most of the proofs to follow, we do case analysis on the last rule of a derivation. In such a situation, the inductive cases for both successor and limit ordinals are basically covered by the case analysis on the inference figures involved, and we shall not make explicit use of transfinite principles.

With respect to the use of eigenvariables and parameters in a derivation, there may be occurrences of the formers that are not free in the end sequent. We refer to these variables and parameters as the *internal variables and parameters*, respectively. We view the choices of those variables and parameters as arbitrary and therefore identify derivations which differ on the choice of internal variables and parameters. In other words, we quotient derivations modulo injective renaming of internal eigenvariables and parameters.

Notation. To ease presentation, we shall use the following notations in the following to abbreviate derivations: The derivation

$$\frac{\begin{array}{c}\Pi_1\\ \Delta_1 \longrightarrow B_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi_n\\ \Delta_n \longrightarrow B_n\end{array} \quad \begin{array}{c}\Pi\\ B_1,\ldots,B_n,\Gamma \longrightarrow C\end{array}}{\Delta_1,\ldots,\Delta_n,\Gamma \longrightarrow C}\ mc$$

is abbreviated as $mc(\Pi_1,\ldots,\Pi_n,\Pi)$. Whenever we write $mc(\Pi_1,\ldots,\Pi_n,\Pi)$ we assume implicitly that the derivation is well-formed, i.e., Π is a derivation ending with some sequent $\Gamma \longrightarrow C$ and the right-hand side of the end sequent of each Π_i is a formula $B_i \in \Gamma$. We use the notation Id_B to denote the derivation ending with the *init* rule on the sequent $B \longrightarrow B$. The notation $subst(\{\Pi^\theta\}_{\theta\in\mathbb{S}})$ denotes a derivation ending with the rule *subst* with premise derivations $\{\Pi^\theta\}_{\theta\in\mathbb{S}}$. Similarly, we use $\mathrm{eq}\mathcal{L}(\{\Pi^\theta\}_{\theta\in\mathbb{U}(s,t)})$ to denote a derivation ending with $\mathrm{eq}\mathcal{L}$, where the equality being introduced is $s=t$, and with premise derivations $\{\Pi^\theta\}_{\theta\in\mathbb{U}(s,t)}$.

4.1. Instantiating eigenvariables

The following definition extends eigenvariable substitutions to apply to derivations. Since we identify derivations that differ only in the choice of internal eigenvariables, we will assume that such variables are chosen to be distinct from the variables in the domain of the substitution and from the free variables of the range of the substitution. Thus applying a substitution to a derivation will only affect all the variables free in the end-sequent.

Definition 5. If Π is a derivation of $\Gamma \longrightarrow C$ and θ is a substitution, then we define the derivation $\Pi[\theta]$ of $\Gamma[\theta] \longrightarrow C[\theta]$ as follows:

1. Suppose Π ends with the $\mathrm{eq}\mathcal{L}$ rule as shown below left. Observe that if $s[\theta][\rho]=t[\theta][\rho]$ then $s[\theta\circ\rho]=t[\theta\circ\rho]$. So we have $\mathbb{U}(s[\theta],t[\theta]) \subseteq \mathbb{U}(s,t)$. Thus $\Pi[\theta]$ is as shown below right:
$$\frac{\left\{\begin{array}{c}\Pi^\rho\\ \Gamma'[\rho]\longrightarrow C[\rho]\end{array}\right\}_{\rho\in\mathbb{U}(s,t)}}{s=t,\Gamma'\longrightarrow C}\ \mathrm{eq}\mathcal{L}, \qquad \frac{\left\{\begin{array}{c}\Pi^{\theta\circ\rho}\\ \Gamma'[\theta][\rho]\longrightarrow C[\theta][\rho]\end{array}\right\}_{\rho\in\mathbb{U}(s[\theta],t[\theta])}}{s[\theta]=t[\theta],\Gamma'[\theta]\longrightarrow C[\theta]}\ \mathrm{eq}\mathcal{L}.$$
2. If Π ends with *subst* with premise derivations $\{\Pi^\rho\}_{\rho\in\mathbb{S}}$ then $\Pi[\theta]$ also ends with the same rule and has premise derivations $\{\Pi^{\theta\circ\rho}\}_{\rho\in\mathbb{S}}$.
3. If Π ends with any other rule and has premise derivations $\Pi_1,\ldots,\Pi_n$, then $\Pi[\theta]$ ends with the same rule and has premise derivations $\Pi_1[\theta],\ldots,\Pi_n[\theta]$.

Notice that in the case where Π ends with $\mathrm{eq}\mathcal{L}$ or *subst*, the substitution θ is *not* recursively applied to the premise derivations of Π; the set of premise derivations of $\Pi[\theta]$ is a subset of the set of premise derivations of Π.

Among the premise sequents of the inference rules of Linc_i^- (with the exception of $\mathrm{CI}\mathcal{R}$), certain premises share the same right-hand side formula with the sequent in the conclusion. We refer to such premises as major premises. This notion of major premise will be useful in proving cut-elimination, as certain proof transformations involve only major premises.

Definition 6. Given an inference rule R with one or more premises, we define its *major premise*(s) as follows.

1. If R is either $\supset\mathcal{L}$, mc or $\mathrm{I}\mathcal{L}$, then its rightmost premise is the major premise.
2. If R is $\mathrm{CI}\mathcal{R}$ then its left premise is the major premise.
3. Otherwise, all the premises of R are major premises.

A *minor premise* of a rule R is a premise of R which is not a major premise. The definition extends to derivations by replacing premise sequents with premise derivations.

Lemma 3. *For any derivation Π and substitution θ,* $\mathrm{ht}(\Pi) \geqslant \mathrm{ht}(\Pi[\theta])$.

Proof. By induction on $\mathrm{ht}(\Pi)$. Note that $\mathrm{ht}(\Pi[\theta])$ can be smaller than $\mathrm{ht}(\Pi)$ because substitutions may reduce the number of premises in $\mathrm{eq}\mathcal{L}$, i.e., if Π ends with an $\mathrm{eq}\mathcal{L}$ acting on, say $x=y$ (which are unifiable), and θ maps x and y to distinct constants then $\Pi[\theta]$ ends with $\mathrm{eq}\mathcal{L}$ with an empty premise. □

Lemma 4. *For any derivation Π and substitutions θ and ρ, the derivations $\Pi[\theta][\rho]$ and $\Pi[\theta\circ\rho]$ are the same derivation.*

4.2. Instantiating parameters

Definition 7. A *parameter substitution* Θ is a partial map from parameters to pairs of proofs and *closed* terms such that whenever

$$\Theta(X^p) = (\Pi_S, S)$$

then S has the same type as p and either one of the following holds:

- $p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$, for some B and $\vec{x}$, and Π_S is a derivation of $B\,S\,\vec{x} \longrightarrow S\,\vec{x}$, or
- $p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$, for some B and $\vec{x}$, and Π_S is a derivation of $S\,\vec{x} \longrightarrow B\,S\,\vec{x}$.

The *support* of Θ is the set

$$supp(\Theta) = \{X^p \mid \Theta(X^p) \text{ is defined}\}.$$

We consider only parameter substitutions with finite support.

We say that X^p is *fresh for* Θ, written $X^p \# \Theta$, if for each $Y^q \in supp(\Theta)$, $X^p \neq Y^q$ and X^p does not occur in S whenever $\Theta(Y^q) = (\Pi_S, S)$.

We shall often enumerate a parameter substitution using a similar notation to (eigenvariables) substitution, e.g., $[(\Pi_1, S_1)/X^{p_1}, \ldots, (\Pi_n, S_n)/X^{p_n}]$ denotes a parameter substitution Θ with support $\{X^{p_1}, \ldots, X^{p_n}\}$ such that $\Theta(X^{p_i}) = (\Pi_i, S_i)$. Given the same Θ and a formula C, we write $C[\Theta]$ to denote the formula $C[S_1/X^{p_1}, \ldots, S_n/X^{p_n}]$.

Definition 8. Let Π be a derivation of $\Gamma \longrightarrow C$ and let Θ be a parameter substitution. Define the derivation $\Pi[\Theta]$ of $\Gamma[\Theta] \longrightarrow C[\Theta]$ by induction on the height of Π as follows:

- Suppose $C = X^p\,\vec{t}$ for some X^p such that $\Theta(X^p) = (\Pi_S, S)$ and Π ends with $\mathrm{I}\mathcal{R}_p$, where $p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$, as shown below left. Then $\Pi[\Theta]$ is as shown below right:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow B\,X^p\vec{t}\end{array}}{\Gamma \longrightarrow X^p\vec{t}}\ \mathrm{I}\mathcal{R}_p, \qquad \dfrac{\begin{array}{c}\Pi'[\Theta] \\ \Gamma[\Theta] \longrightarrow B\,S\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S[\vec{t}/\vec{x}] \\ B\,S\,\vec{t} \longrightarrow S\,\vec{t}\end{array}}{\Gamma[\Theta] \longrightarrow S\,\vec{t}}\ mc.$$

- Similarly, suppose Π ends with $\mathrm{CI}\mathcal{L}_p$ on $X^p\,\vec{t}$:

$$\dfrac{\begin{array}{c}\Pi' \\ B\,X^p\vec{t}, \Gamma' \longrightarrow C\end{array}}{X^p\vec{t}, \Gamma' \longrightarrow C}\ \mathrm{CI}\mathcal{L}_p$$

  where $p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$ and $\Theta(X^p) = (\Pi_S, S)$. Then $\Pi[\Theta]$ is

$$\dfrac{\dfrac{\dfrac{}{S\,\vec{t} \longrightarrow S\,\vec{t}}\ init \quad \begin{array}{c}\Pi_S[\vec{t}/\vec{x}] \\ S\,\vec{t} \longrightarrow B\,S\,\vec{t}\end{array}}{S\,\vec{t} \longrightarrow B\,S\,\vec{t}}\ mc \qquad \begin{array}{c}\Pi'[\Theta] \\ B\,S\,\vec{t}, \Gamma'[\Theta] \longrightarrow C[\Theta]\end{array}}{S\,\vec{t}, \Gamma'[\Theta] \longrightarrow C[\Theta]}\ mc.$$

- In all other cases, suppose Π ends with a rule R with premise derivations $\{\Pi_i\}_{i\in\mathbb{I}}$ for some index set $\mathbb{I}$. Since we identify derivations up to renaming of internal parameters, we assume without loss of generality that the internal eigenvariables in the premises of R (if any) do not appear in Θ. Then $\Pi[\Theta]$ ends with the same rule, with premise derivations $\{\Pi_i[\Theta]\}_{i\in\mathbb{I}}$.

Remark 3. Definition 8 is asymmetric in the treatment of inductive and co-inductive parameters, i.e., in the cases where Π ends with $\mathrm{I}\mathcal{R}_p$ and $\mathrm{CI}\mathcal{L}_p$. In the latter case, the substituted derivation uses a seemingly unnecessary cut, i.e., $mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}])$. The reason behind this is rather technical; in our main cut elimination proof, we need to establish that $\Pi_S[\vec{t}/\vec{x}]$ is “reducible” (i.e., all the cuts in it can be eventually eliminated), given that $mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}])$ is reducible. In a typical cut elimination procedure one would have expected that $mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}])$ reduces to $\Pi_S[\vec{t}/\vec{x}]$, hence reducibility of $\Pi_S[\vec{t}/\vec{x}]$ would follow from reducibility of $mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}])$. However, according to our cut reduction rules (see Section 5.1), $mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}])$ does not necessarily reduce to $\Pi_S[\vec{t}/\vec{x}]$. Still, if the instance of *init* appears instead on the right premise of the cut, e.g., as in

$$\dfrac{\begin{array}{c}\Pi_S[\vec{t}/\vec{x}] \\ B\,S\,\vec{t} \longrightarrow S\,\vec{t}\end{array} \quad \dfrac{}{S\,\vec{t} \longrightarrow S\,\vec{t}}\ init}{B\,S\,\vec{t} \longrightarrow S\,\vec{t}}\ mc$$

the cut elimination procedure does reduce this to $\Pi_S[\vec{t}/\vec{x}]$, so it is not necessary to introduce explicitly this cut instance in the case involving inductive parameters. It is possible to define a symmetric notion of parameter substitution, but that would require different cut reduction rules than the ones we proposed in this paper. Another possibility would be to push the asymmetry to the definition of *reducibility* (see Section 5). We have explored these alternative options, but for the purpose of proving cut elimination, we found that the current definition yields a simpler proof.[3]

Note that since parameter substitutions replace parameters with closed terms, they commute with (eigenvariable) substitutions. When writing a sequence of applications of eigenvariable substitutions and parameter substitutions, we again omit parentheses to simplify presentation. So for example, we shall write $\Pi[\Theta][\delta]$ to mean $(\Pi[\Theta])[\delta]$.

Lemma 5. *For every derivation Π, substitution δ, parameter substitution Θ, the derivation $\Pi[\Theta][\delta]$ is the same as the derivation $\Pi[\delta][\Theta]$.*

In the following, we denote with $[\Theta, (\Pi_S, S)/X^p]$, where $X^p\#\Theta$, a parameter substitution obtained by extending Θ with the map $X^p \mapsto (\Pi_S, S)$.

Lemma 6. *Let Π be a derivation of $\Gamma \longrightarrow C$, Θ a parameter substitution and X^p a parameter such that $X^p\#supp(\Theta)$. Then for every Π_S and every S:*

1. *if X^p does not occur in $\Gamma \longrightarrow C$, then $\Pi[\Theta, (\Pi_S, S)/X^p] = \Pi[\Theta]$; and*
2. *$\Pi[\Theta][(\Pi_S, S)/X^p] = \Pi[\Theta, (\Pi_S, S)/X^p]$.*

Proof. Both are proved by induction on Π. Statement 1 is trivial. We show here a non-trivial case of the proof of statement 2.

Let $\Theta_1 = [(\Pi_S, S)/X^p]$ and $\Theta_2 = [\Theta, (\Pi_S, S)/X^p]$. We first note that for any formula A, we have $A[\Theta][\Theta_1] = A[\Theta_2]$ by the freshness assumption $X^p\#\Theta$. Further, $\Gamma[\Theta][\Theta_1] = \Gamma[\Theta_2]$.

- Suppose $C = X^p\,\vec{t}$ and Π ends with $\mathrm{I}\mathcal{R}_p$, where $p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$, as shown below left. Then $\Pi[\Theta]$ is as shown below right:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow B\,X^p\vec{t}\end{array}}{\Gamma \longrightarrow X^p\vec{t}}\ \mathrm{I}\mathcal{R}_p, \qquad \dfrac{\begin{array}{c}\Pi'[\Theta] \\ \Gamma[\Theta] \longrightarrow B\,X^p\vec{t}\end{array}}{\Gamma[\Theta] \longrightarrow X^p\vec{t}}\ \mathrm{I}\mathcal{R}_p.$$

  The derivation $\Pi[\Theta][\Theta_1]$ is $mc(\Pi'[\Theta][\Theta_1], \Pi_S[\vec{t}/\vec{x}])$. On the other hand, $\Pi[\Theta_2] = mc(\Pi'[\Theta_2], \Pi_S[\vec{t}/\vec{x}])$. By the induction hypothesis, $\Pi'[\Theta][\Theta_1] = \Pi'[\Theta_2]$, and therefore $\Pi[\Theta_2] = \Pi[\Theta][\Theta_2]$.
- Suppose $C = Y^q\,\vec{t}$, where $\Theta(Y^q) = (\Pi_I, I)$, and Π ends with $\mathrm{I}\mathcal{R}_p$ as shown below left. Then $\Pi[\Theta]$ is shown below right:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow B\,Y^q\vec{t}\end{array}}{\Gamma \longrightarrow Y^q\vec{t}}\ \mathrm{I}\mathcal{R}_p, \qquad \dfrac{\begin{array}{c}\Pi'[\Theta] \\ \Gamma[\Theta] \longrightarrow B\,I\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_I[\vec{t}/\vec{x}] \\ B\,I\,\vec{t} \longrightarrow I\,\vec{t}\end{array}}{\Gamma[\Theta] \longrightarrow I\,\vec{t}}\ mc.$$

  Note that since $X^p\#\Theta$, by Definition 7 X^p does not occur in I. Since B cannot contain any parameters, it follows that X^p does not occur in the sequent $B\,I\,\vec{t} \longrightarrow I\,\vec{t}$. Therefore by statement 1, we have $\Pi_I[\vec{t}/\vec{x}][\Theta_1] = \Pi_I[\vec{t}/\vec{x}]$. Therefore, $\Pi[\Theta][\Theta_1] = mc(\Pi'[\Theta][\Theta_1], \Pi_I[\vec{t}/\vec{x}])$. By the induction hypothesis, $\Pi'[\Theta][\Theta_1] = \Pi'[\Theta_2]$. Therefore $\Pi[\Theta_2] = \Pi[\Theta][\Theta_1]$.

The cases where Π ends with $\mathrm{CI}\mathcal{L}_p$ can be proved analogously. □

5. Cut elimination for $\mathbf{Linc}_i^-$

The central result of our work is cut-elimination, from which consistency of the logic follows. Gentzen's classic proof of cut-elimination for first-order logic uses an induction on the size of the cut formula. The cut-elimination procedure consists of a set of reduction rules that reduces a cut of a compound formula to cuts on its sub-formulae of smaller size. In the case of Linc_i^-, the use of induction/co-induction complicates the reduction of cuts. Consider for example a cut involving the induction rules:

$$\dfrac{\dfrac{\begin{array}{c}\Pi_1 \\ \Delta \longrightarrow B\,X^p\,\vec{t}\end{array}}{\Delta \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R} \qquad \dfrac{\begin{array}{c}\Pi_B \\ B\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{array} \quad \begin{array}{c}\Pi \\ S\,\vec{t}, \Gamma \longrightarrow C\end{array}}{p\,\vec{t}, \Gamma \longrightarrow C}\ \mathrm{I}\mathcal{L}}{\Delta, \Gamma \longrightarrow C}\ mc.$$

[3] But we conjecture that in the classical case a fully symmetric definition of parameter substitution and cut reduction would be needed.

There are at least two problems in reducing this cut. First, any permutation upwards of the cut will necessarily involve a cut with S that can be of larger size than p, and hence a simple induction on the size of the cut formula will not work. Second, the invariant S does not appear in the conclusion of the left premise of mc. This means that we need to transform the left premise so that its end sequent will agree with the right premise. Any such transformation will most likely be *global*, and hence simple induction on the height of derivations will not work either.

We shall use the *reducibility* technique to prove cut elimination. More specifically, we shall build on the notion of reducibility introduced by Martin-Löf to prove normalization of an intuitionistic logic with iterative inductive definition [21]. Martin-Löf's proof has been adapted to sequent calculus by McDowell and Miller [22], but in a restricted setting where only natural number induction is allowed. Since our logic involves arbitrary inductive definitions, which also includes iterative inductive definitions, we shall need different, and more general, cut reductions. But the real difficulty in our case is in establishing cut elimination in the presence of co-inductive definitions.

The main part of the reducibility technique is a definition of the family of reducible sets of derivations. In Martin-Löf's theory of iterative inductive definition, this family of sets is defined inductively by the "type" of the derivations they contain, i.e., the formula in the right-hand side of the end sequent in a derivation. Extending this definition of reducibility to Linc_i^- is not obvious. In particular, in establishing the reducibility of a derivation of type $p\,\vec{t}$ ending with a $\mathrm{CI}\mathcal{R}$ rule one must first establish the reducibility of its premise derivations, which may have larger types, since $S\,\vec{t}$ could be any formula. Therefore a simple inductive definition based on types of derivations would not be well-founded.

The key to properly stratifying the definition of reducibility is to consider reducibility under parameter substitutions. This notion of reducibility, called *parametric reducibility*, was originally developed by Girard to prove strong normalization of System F, i.e., in the interpretation of universal types. As with strong normalization of System F, (co-)inductive parameters are substituted with some "reducibility candidates", which in our case are certain sets of derivations satisfying closure conditions similar to those for System F, but which additionally satisfy certain closure conditions related to (co-)inductive definitions.

The remainder of this section is structured as follows. In Section 5.1 we define a set of cut reduction rules that are used to eliminate the applications of the cut rule. For the cases involving logical operators, our cut-reduction rules are the same as those in [22]. The crucial differences are, of course, in the rules involving induction and co-induction rules, where we use the transformation described in Definition 7. We then proceed to define two notions essential to our cut elimination proof: *normalizability* (Section 5.2) and *parametric reducibility* (Section 5.3). These can be seen as counterparts for Martin-Löf's notions of normalizability and *computability* [21], respectively. Normalizability of a derivation implies that all the cuts in it can be eventually eliminated (via the cut reduction rules defined earlier). Reducibility is a stronger notion, in that it implies normalizability. The main part of the cut elimination proof is presented in Section 5.4, where we show that every derivation is reducible, hence it can be turned into a cut-free derivation.

5.1. Cut reduction

We now define a reduction relation on derivations ending with mc, following McDowell and Miller [22].

Definition 9 *(Reduction).* We define a *reduction* relation between derivations. The redex is always a derivation Ξ ending with the multicut rule

$$\dfrac{\begin{matrix}\Pi_1 \\ \Delta_1 \longrightarrow B_1\end{matrix} \quad \cdots \quad \begin{matrix}\Pi_n \\ \Delta_n \longrightarrow B_n\end{matrix} \quad \begin{matrix}\Pi \\ B_1,\ldots,B_n,\Gamma \longrightarrow C\end{matrix}}{\Delta_1,\ldots,\Delta_n,\Gamma \longrightarrow C}\ mc.$$

We refer to the formulas $B_1,\ldots,B_n$ produced by the mc as *cut formulas*.

If $n = 0$, Ξ reduces to the premise derivation Π. For $n > 0$ we specify the reduction relation based on the last rule of the premise derivations. If the rightmost premise derivation Π ends with a left rule acting on a cut formula B_i, then the last rule of Π_i and the last rule of Π together determine the reduction rules that apply. We classify these rules according to the following criteria: we call the rule an *essential* case when Π_i ends with a right rule; if it ends with a left rule or *subst*, it is a *left-commutative* case; if Π_i ends with the *init* rule, then we have an *axiom* case; a *multicut* case arises when it ends with the mc rule. When Π does not end with a left rule acting on a cut formula, then its last rule is alone sufficient to determine the reduction rules that apply. If Π ends with *subst* or a rule acting on a formula other than a cut formula, then we call this a *right-commutative* case. A *structural* case results when Π ends with a contraction or weakening on a cut formula. If Π ends with the *init* rule, this is also an axiom case; similarly a multicut case arises if Π ends in the mc rule. For simplicity of presentation, we always show $i = 1$.

This reduction relation is an extension of the similar cut reduction relation used in the cut elimination proof for $FO\lambda^{\Delta\mathbb{N}}$ [22]. The main differences are in the rules involving induction and co-induction. There is also slight difference in one reduction rule involving equality, which in our case utilizes the derived rule *subst*. Therefore in the following definition, we shall highlight only those reductions that involve (co-)induction and equality rules. Note that the left-commutative case where Π_1 ends with $\mathrm{CI}\mathcal{L}$ is subsumed by the reduction rule $\bullet\mathcal{L}/\circ\mathcal{L}$. The complete list of reduction rules can be found in Appendix A.

Essential cases:

$\boxed{\mathrm{eq}\,\mathcal{L}/\,\mathrm{eq}\,\mathcal{R}}$ Suppose Π_1 and Π are

$$\frac{}{\Delta_1 \longrightarrow s = t}\ \mathrm{eq}\,\mathcal{R}, \qquad \frac{\left\{\begin{matrix}\Pi^{\rho} \\ B_2[\rho], \ldots, B_n[\rho], \Gamma[\rho] \longrightarrow C[\rho]\end{matrix}\right\}_{\rho \in \mathbb{U}(s,t)}}{s = t, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{eq}\,\mathcal{L}.$$

Note that because $s =_{\beta\eta} t$, we have $\mathbb{U}(s,t) = \mathbb{S}$, i.e., any substitution is a unifier of s and t. Let $\Xi_1 = mc(\Pi_2, \ldots, \Pi_n, subst(\{\Pi^{\rho}\}_{\rho \in \mathbb{S}})$. In this case Ξ reduces to

$$\dfrac{\begin{matrix}\Xi_1 \\ \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C\end{matrix}}{\Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}$$

where we use the double horizontal lines to indicate that the relevant inference rule (in this case, $\mathrm{w}\mathcal{L}$) may need to be applied zero or more times.

$\boxed{\mathrm{I}\mathcal{R}/\mathrm{I}\mathcal{L}}$ Suppose Π_1 and Π are, respectively,

$$\frac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow D\, X^p\, \vec{t}\end{matrix}}{\Delta_1 \longrightarrow p\, \vec{t}}\ \mathrm{I}\mathcal{R}, \qquad \frac{\begin{matrix}\Pi_S \\ D\, S\, \vec{y} \longrightarrow S\, \vec{y}\end{matrix} \quad \begin{matrix}\Pi' \\ S\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{matrix}}{p\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{I}\mathcal{L}$$

where $p\, \vec{x} \stackrel{\mu}{=} D\, p\, \vec{x}$ and X^p is a new parameter. Then Ξ reduces to

$$mc\big(mc\big(\Pi_1'\big[(\Pi_S, S)/X^p\big], \Pi_S[\vec{t}/\vec{y}]\big), \Pi_2, \ldots, \Pi_n, \Pi'\big).$$

$\boxed{\mathrm{CI}\,\mathcal{R}/\,\mathrm{CI}\,\mathcal{L}}$ Suppose Π_1 and Π are

$$\frac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow S\, \vec{t}\end{matrix} \quad \begin{matrix}\Pi_S \\ S\, \vec{y} \longrightarrow D\, S\, \vec{y}\end{matrix}}{\Delta_1 \longrightarrow p\, \vec{t}}\ \mathrm{CI}\,\mathcal{R}, \qquad \frac{\begin{matrix}\Pi' \\ D\, X^p\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{matrix}}{p\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{CI}\,\mathcal{L}$$

where $p\, \vec{x} \stackrel{\nu}{=} D\, p\, \vec{x}$ and X^p is a new parameter. Then Ξ reduces to

$$mc\big(mc\big(\Pi_1', \Pi_S[\vec{t}/\vec{y}]\big), \Pi_2, \ldots, \Pi_n, \Pi'\big[(\Pi_S, S)/X^p\big]\big).$$

Left-commutative cases: In the following, we suppose that Π ends with a left rule, other than $\{\mathrm{c}\mathcal{L}, \mathrm{w}\mathcal{L}\}$, acting on B_1.

$\boxed{\mathrm{I}\mathcal{L}/\circ\mathcal{L}}$ Suppose Π_1 is

$$\frac{\begin{matrix}\Pi_S \\ D\, S\, \vec{y} \longrightarrow S\, \vec{y}\end{matrix} \quad \begin{matrix}\Pi_1' \\ S\, \vec{t}, \Delta_1' \longrightarrow B_1\end{matrix}}{p\, \vec{t}, \Delta_1' \longrightarrow B_1}\ \mathrm{I}\mathcal{L}$$

where $p\, \vec{x} \stackrel{\mu}{=} D\, p\, \vec{x}$. Let $\Xi_1 = mc(\Pi_1', \Pi_2, \ldots, \Pi_n, \Pi)$. Then Ξ reduces to

$$\frac{\begin{matrix}\Pi_S \\ D\, S\, \vec{y} \longrightarrow S\, \vec{y}\end{matrix} \quad \begin{matrix}\Xi_1 \\ S\, \vec{t}, \Delta_1', \ldots, \Delta_n, \Gamma \longrightarrow C\end{matrix}}{p\, \vec{t}, \Delta_1', \ldots, \Delta_n \longrightarrow C}\ \mathrm{I}\mathcal{L}.$$

Right-commutative cases:

$\boxed{-/\mathrm{I}\mathcal{L}}$ Suppose Π is

$$\frac{\begin{matrix}\Pi_S \\ D\, S\, \vec{y} \longrightarrow S\, \vec{y}\end{matrix} \quad \begin{matrix}\Pi' \\ B_1, \ldots, B_n, S\, \vec{t}, \Gamma' \longrightarrow C\end{matrix}}{B_1, \ldots, B_n, p\, \vec{t}, \Gamma' \longrightarrow C}\ \mathrm{I}\mathcal{L}$$

where $p\,\vec{x} \stackrel{\mu}{=} D\,p\,\vec{x}$. Let $\Xi_1 = mc(\Pi_1, \ldots, \Pi_n, \Pi')$. Then Ξ reduces to

$$\dfrac{\begin{array}{c}\Pi_S\\ D\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{array} \quad \begin{array}{c}\Xi_1\\ \Delta_1, \ldots, \Delta_n, S\,\vec{t}, \Gamma' \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, p\,\vec{t}, \Gamma' \longrightarrow C}\ \mathrm{I}\mathcal{L}.$$

$\boxed{-/\mathrm{CI}\mathcal{R}}$ Suppose Π is

$$\dfrac{\begin{array}{c}\Pi'\\ B_1, \ldots, B_n, \Gamma \longrightarrow S\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S\\ S\,\vec{y} \longrightarrow D\,S\,\vec{y}\end{array}}{B_1, \ldots, B_n, \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}$$

where $p\,\vec{x} \stackrel{\nu}{=} D\,p\,\vec{x}$. Let $\Xi_1 = mc(\Pi_1, \ldots, \Pi_n, \Pi')$. Then Ξ reduces to

$$\dfrac{\begin{array}{c}\Xi_1\\ \Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow S\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S\\ S\,\vec{y} \longrightarrow D\,S\,\vec{y}\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}.$$

It is clear from an inspection of the inference rules in Fig. 1 and the definition of cut reduction (see Appendix A) that every derivation ending with a multicut has a reduct. Note that since the left-hand side of a sequent is a multiset, the same formula may occur more than once in the multiset. In the cut reduction rules, we should view these occurrences as distinct so that no ambiguity arises as to which occurrence of a formula is subject to the *mc* rule.

The following lemma shows that the reduction relation is preserved by eigenvariable substitution. The proof is given in Appendix B.

Lemma 7. *Let Π be a derivation ending with a mc and let θ be a substitution. If $\Pi[\theta]$ reduces to Ξ then there exists a derivation Π' such that $\Xi = \Pi'[\theta]$ and Π reduces to Π'.*

Remark 4. An apparently simpler reduction rule $\mathrm{eq}\mathcal{R}/\mathrm{eq}\mathcal{L}$ in Definition 9, which does not use the *subst*-rule, would be to let $\Xi_1 = mc(\Pi_2, \ldots, \Pi_n, \Pi^\epsilon)$. Alas, if this reduction rule is used, then Lemma 7 would fail precisely in this case.[4] To see the problem, suppose Π is

$$\dfrac{\dfrac{}{\longrightarrow t = t}\ \mathrm{eq}\mathcal{R} \qquad \dfrac{\left\{\begin{array}{c}\Pi^\rho\\ \Gamma[\rho] \longrightarrow C\rho\end{array}\right\}_{\rho \in \mathbb{S}}}{t = t, \Gamma \longrightarrow C}\ \mathrm{eq}\mathcal{L}}{\Gamma \longrightarrow C}\ mc.$$

Then Π reduces to $mc(\Pi^\epsilon)$, but $\Pi[\theta]$ reduces to $mc(\Pi^{\theta\circ\epsilon})$, not $mc(\Pi^\epsilon[\theta])$. In general, $\Pi^{\theta\circ\epsilon}$ may not be the same as $\Pi^\epsilon[\theta]$, so clearly Lemma 7 would be invalidated if this reduction were adopted. One fix to this problem is to use stronger definitions of normalizability and reducibility that consider all reductions of *all instances* of a redex, so that Lemma 7 would not be needed, in the first place e.g., as in the cut-elimination proof in [22]. This, however, has the drawback of a more complicated case analysis in the main cut elimination proof (Lemma 19). Another approach, adopted by Baelde in his cut elimination proof for μMALL [4], is to restrict the $\mathrm{eq}\mathcal{L}$-rule to use a certain set of CSU, and to modify the definition of substitutions of eigenvariables in proofs (in Baelde's case, the substitution is recursively applied to the premises of $\mathrm{eq}\mathcal{L}$, unlike our case). Instead, we use a *subst* rule to postpone eliminating this $\mathrm{eq}\mathcal{R}/\mathrm{eq}\mathcal{L}$-cut until all other cuts have been eliminated. This has the advantage that one does not need to place any restrictions on the $\mathrm{eq}\mathcal{L}$ rule and it does not rely on specific properties of CSU. This may help in extending our methods to proof systems with different notions of substitution, such as the logic $\mathcal{G}$ [15].

5.2. Normalizability

Definition 10. We define the set of *normalizable* derivations to be the smallest set that satisfies the following conditions:

1. If a derivation Π ends with *mc*, then it is normalizable if every reduct of Π is normalizable.
2. If a derivation ends with any rule other than *mc*, then it is normalizable if the premise derivations are normalizable.

The set of all normalizable derivations is denoted by **NM**.

[4] We are grateful to David Baelde who pointed out this problem.

Each clause in the definition of normalizability asserts that a derivation is normalizable if certain (possibly infinitely many) other derivations are normalizable. We call the latter the *predecessors* of the former. Thus a derivation is normalizable if the tree of its successive predecessors is well-founded. We refer to this well-founded tree as its *normalization*. Since a normalization is well-founded, it has an associated induction principle: for any property P of derivations, if for every derivation Π in the normalization, P holds for every predecessor of Π implies that P holds for Π, then P holds for every derivation in the normalization. We shall define explicitly a measure on a normalizable derivation based on its normalization tree.

Definition 11 *(Normalization degree).* Let Π be a normalizable derivation. The *normalization degree of* Π, denoted by $nd(\Pi)$, is defined by induction on the normalization of Π as follows:

$$nd(\Pi) = 1 + \left(\left\{nd\left(\Pi'\right) \mid \Pi' \text{ is a predecessor of } \Pi\right\}\right).$$

The normalization degree of Π is basically the height of its normalization tree. Note that $nd(\Pi)$ can be an ordinal in general, due to the possibly infinite-branching rule eq$\mathcal{L}$.

Lemma 8. *If there is a normalizable derivation of a sequent, then there is a cut-free derivation of the sequent.*

Proof. Similarly to [22]. □

In the proof of the main lemma for cut elimination (Lemma 19) we shall use induction on the normalization degree, instead of using directly the normalization ordering. The reason is that in some inductive cases in the proof, we need to compare a (normalizable) derivation with its instances, but the normalization ordering does not necessarily relate the two, e.g., Π and $\Pi[\theta]$ may not be related by the normalization ordering, although their normalization degrees are (see Lemma 10). Later, we shall define a stronger ordering called *reducibility*, which implies normalizability. In the cut elimination proof for $FO\lambda^{\Delta\mathbb{N}}$ [22], in one of the cases, an implicit reducibility ordering is assumed to hold between derivation Π and its instance $\Pi[\theta]$. As the reducibility ordering in their setting is a subset of the normalizability ordering, this assumption may not hold in all cases, and as a consequence there is a gap in the proof in [22].[5]

The next lemma states that normalization is closed under substitutions.

Lemma 9. *If Π is a normalizable derivation, then for any substitution θ $\Pi[\theta]$ is normalizable.*

Proof. By induction on $nd(\Pi)$.

1. If Π ends with *mc*, then $\Pi[\theta]$ also ends with *mc*. By Lemma 7 every reduct of $\Pi[\theta]$ corresponds to a reduct of Π, therefore by induction hypothesis every reduct of $\Pi[\theta]$ is normalizable, and hence $\Pi[\theta]$ is normalizable.
2. Suppose Π ends with a rule other than *mc* and has premise derivations $\{\Pi_i\}_{i\in\mathbb{I}}$ for some index set $\mathbb{I}$. By Definition 5 each premise derivation of $\Pi[\theta]$ is either Π_i (i.e., in the case where Π ends with eq$\mathcal{L}$ or *subst*, in which case the premise derivations of $\Pi[\theta]$ are already in $\{\Pi_i\}_{i\in\mathbb{I}}$), or $\Pi_i[\theta]$. Since Π is normalizable, Π_i is normalizable, and so by the induction hypothesis $\Pi_i[\theta]$ is also normalizable. Thus $\Pi[\theta]$ is normalizable. □

The normalization degree is non-increasing under eigenvariable substitution.

Lemma 10. *Let Π be a normalizable derivation. Then $nd(\Pi) \geqslant nd(\Pi[\theta])$ for every substitution θ.*

Proof. By induction on $nd(\Pi)$ using Definition 5 and Lemma 7. □

5.3. Parametric reducibility

In the following, we shall use the term “type” in two different settings: in categorizing terms and in categorizing derivations. To avoid confusion, we shall refer to the types of terms as *syntactic types*, and the term “type” is reserved for types of derivations.

Our notion of a type of a set of derivations may abstract from particular terms in a formula. This is because our definition of reducibility (candidates) will have to be closed under eigenvariable substitutions, which is in turn imposed by the fact that our proof rules allow instantiation of eigenvariables in the derivations (i.e., the eq$\mathcal{L}$ and the *subst* rules).

[5] This gap was fixed in [41] by strengthening the main lemma for cut elimination. Andrew Gacek and Gopalan Nadathur proposed another fix by assigning an explicit ordinal to each reducible derivation, and using the ordering on ordinals to replace the reducibility ordering in the lemma. A discussion of these fixes can be found in the errata page of that paper [22]: http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tcs00.errata.html. We essentially follow Gacek and Nadathur's approach here, although we assign ordinals to normalizable derivations rather than to reducible derivations.

Definition 12 *(Types of derivations).* We say that *a derivation Π has type C* if the end sequent of Π is of the form $\Gamma \longrightarrow C$ for some Γ. Let S be a term with syntactic type $\tau_1 \to \cdots \to \tau_n \to o$. A set of derivations $\mathcal{S}$ is said to be *of type S* if every derivation in $\mathcal{S}$ has type $S\,u_1 \ldots u_n$ for some terms $u_1, \ldots, u_n$. Given a list of terms $\vec{u} = u_1 : \tau_1, \ldots, u_n : \tau_n$ and a set of derivations $\mathcal{S}$ of type $S : \tau_1 \to \cdots \to \tau_n \to o$, we denote with $\mathcal{S}\,\vec{u}$ the set

$$\mathcal{S}\,\vec{u} = \{\Pi \in \mathcal{S} \mid \Pi \text{ has type } S\,\vec{u}\}.$$

Definition 13 *(Reducibility candidate).* Let S be a *closed term* having the syntactic type $\tau_1 \to \cdots \to \tau_n \to o$. A set of derivations $\mathcal{R}$ of type S is said to be a *reducibility candidate of type S* if the following hold:

CR0 If $\Pi \in \mathcal{R}$ then $\Pi[\theta] \in \mathcal{R}$, for every θ.
CR1 If $\Pi \in \mathcal{R}$ then Π is normalizable.
CR2 If $\Pi \in \mathcal{R}$ and Π reduces to Π' then $\Pi' \in \mathcal{R}$.
CR3 If Π ends with *mc* and all its reducts are in $\mathcal{R}$, then $\Pi \in \mathcal{R}$.
CR4 If Π ends with *init*, then $\Pi \in \mathcal{R}$.
CR5 If Π ends with a left-rule or *subst*, and all its minor premise derivations are normalizable, and all its major premise derivations are in $\mathcal{R}$, then $\Pi \in \mathcal{R}$.

We shall write $\mathcal{R} : S$ to denote a reducibility candidate $\mathcal{R}$ of type S.

The conditions **CR1** and **CR2** are similar to the eponymous conditions in Girard's definition of reducibility candidates in his strong normalization proof for System F (see [17], Chapter 14). Girard's **CR3** is expanded in our definition to **CR3**, **CR4** and **CR5**. These conditions deal with what Girard refers to as "neutral" proof terms (i.e., proof terms ending with elimination rules or the axiom rule in natural deduction). In our setting, neutrality corresponds to derivations ending in *mc*, *init*, *subst*, or a left rule.

The condition **CR0** is needed because our cut reduction rules involve substitutions of eigenvariables in some cases (i.e., those that involve permutation of $\mathrm{eq}\mathcal{L}$ and *subst* in the left/right commutative cases), and consequently, the notion of reducibility (candidate) needs to be preserved under eigenvariable substitution.

Let $\mathcal{S}$ be a set of derivations of type B and let $\mathcal{T}$ be a set of derivations of type C. Then $\mathcal{S} \Rightarrow \mathcal{T}$ denotes the set of derivations such that $\Pi \in \mathcal{S} \Rightarrow \mathcal{T}$ if and only if Π ends with a sequent $\Gamma \longrightarrow C$ whenever $B \in \Gamma$ and for every $\Xi \in \mathcal{S}$, we have $mc(\Xi, \Pi) \in \mathcal{T}$.

Let S be a closed term. Define $\mathbf{NM}_S$ to be the set

$$\mathbf{NM}_S = \{\Pi \mid \Pi \in \mathbf{NM} \text{ and is of type } S\,\vec{u} \text{ for some } \vec{u}\}.$$

Note that S here is an abstraction, and intuitively, $\mathbf{NM}_S$ consists of the set of all normalizable derivations with types which are instances of S.

Lemma 11. *Let S be a term of syntactic type $\tau_1 \to \cdots \to \tau_n \to o$. Then the set $\mathbf{NM}_S$ is a reducibility candidate of type S.*

Proof. **CR0** follows from Lemma 9, **CR1** follows from the definition of $\mathbf{NM}_S$, and the rest follows from Definition 10. $\square$

Definition 14 *(Candidate substitution).* A *candidate substitution* Ω is a partial map from parameters to triples of reducibility candidates, derivations and closed terms such that whenever $\Omega(X^p) = (\mathcal{R}, \Pi, S)$ we have

- S has the same syntactic type as p,
- $\mathcal{R}$ is a reducibility candidate of type S, and
- either one of the following holds:
  - $p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$ and Π is a normalizable derivation of $B\,S\,\vec{y} \longrightarrow S\,\vec{y}$, or
  - $p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$ and Π is a normalizable derivation of $S\,\vec{y} \longrightarrow B\,S\,\vec{y}$.

We denote with $supp(\Omega)$ the *support* of Ω, i.e., the set of parameters on which Ω is defined. Each candidate substitution Ω determines a unique parameter substitution Θ, given by:

$$\Theta\big(X^p\big) = (\Pi, S) \quad \text{iff} \quad \Omega\big(X^p\big) = (\mathcal{R}, \Pi, S) \text{ for some } \mathcal{R}.$$

We denote with $Sub(\Omega)$ the parameter substitution Θ obtained this way. We say that a parameter X^p is *fresh for Ω*, written $X^p \# \Omega$, if $X^p \# Sub(\Omega)$.

Notation. Since every candidate substitution has a corresponding parameter substitution, we shall often treat a candidate substitution as a parameter substitution. In particular, we shall write $C[\Omega]$ to denote $C[Sub(\Omega)]$ and $\Pi[\Omega]$ to denote $\Pi[Sub(\Omega)]$.

We are now ready to define the notion of *parametric reducibility*. We follow a similar approach for $FO\lambda^{\Delta\mathbb{N}}$ [22], where families of reducibility sets are defined by the *level* of derivations, i.e. the size of the types of derivations. In defining a family (or families) of sets of derivations at level k, we assume that reducibility sets at level $j < k$ are already defined. The main difference with the notion of reducibility for $FO\lambda^{\Delta\mathbb{N}}$, aside from the use of parameters in the clause for (co-)induction rules (which do not exist in $FO\lambda^{\Delta\mathbb{N}}$), is in the treatment of the induction rules.

Definition 15 *(Parametric reducibility).* Let $\mathcal{F}_k$ be the set of all formula of size k (Definition 3), i.e. $\{B \mid |B| = k\}$. The family of *parametric reducibility sets* $\mathbf{RED}_C[\Omega]$, where C is a formula and Ω is a candidate substitution, is defined by induction on the size of C as follows. For each k, the family of *parametric reducibility sets of level k*

$$\{\mathbf{RED}_C[\Omega]\}_{C\in\mathcal{F}_k}$$

is the smallest family of sets satisfying, for each $C \in \mathcal{F}_k$:

P1 Suppose $C = X^p\,\vec{u}$ for some $\vec{u}$ and some parameter X^p. If $X^p \in supp(\Omega)$ then $\mathbf{RED}_C[\Omega] = \mathcal{R}\,\vec{u}$, where $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$. Otherwise,

$$\mathbf{RED}_C[\Omega] = \mathbf{NM}_{X^p}\,\vec{u}.$$

Otherwise, $C \neq X^p\,\vec{u}$, for any $\vec{u}$ and X^p. Then a derivation Π of type $C[\Omega]$ is in $\mathbf{RED}_C[\Omega]$ if it is normalizable and one of the following holds:

P2 Π ends with mc, and all its reducts are in $\mathbf{RED}_C[\Omega]$.

P3 Π ends with $\supset\mathcal{R}$, i.e., $C = B \supset D$ and Π is of the form:

$$\dfrac{\begin{array}{c}\Pi'\\ \Gamma, B[\Omega] \longrightarrow D[\Omega]\end{array}}{\Gamma \longrightarrow B[\Omega] \supset D[\Omega]}\supset\mathcal{R}$$

and for every substitution ρ, $\Pi'[\rho] \in (\mathbf{RED}_{B[\rho]}[\Omega] \Rightarrow \mathbf{RED}_{D[\rho]}[\Omega])$.

P4 Π ends with $\mathrm{I}\mathcal{R}$, i.e.,

$$\dfrac{\begin{array}{c}\Pi'\\ \Gamma \longrightarrow B\,X^p\,\vec{t}\end{array}}{\Gamma \longrightarrow p\,\vec{t}}\,\mathrm{I}\mathcal{R},\quad \text{where } p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}.$$

Without loss of generality, assume that $X^p\#\Omega$: for every reducibility candidate $(\mathcal{S} : I)$, where I is a closed term of the same syntactic type as p, for every normalizable derivation Π_I of $B\,I\,\vec{y} \longrightarrow I\,\vec{y}$, if for every $\vec{u}$ the following holds:

$$\Pi_I[\vec{u}/\vec{y}] \in \big(\mathbf{RED}_{(B\,X^p\,\vec{u})}\big[\Omega, (\mathcal{S}, \Pi_I, I)/X^p\big] \Rightarrow \mathcal{S}\,\vec{u}\big),$$

then $mc(\Pi'[(\Pi_I, I)/X^p], \Pi_I[\vec{t}/\vec{y}]) \in \mathcal{S}\,\vec{t}$.

P5 Π ends with $\mathrm{CI}\mathcal{R}$, i.e.,

$$\dfrac{\begin{array}{c}\Pi'\\ \Gamma \longrightarrow I\,\vec{t}\end{array}\quad\begin{array}{c}\Pi_I\\ I\,\vec{y} \longrightarrow B\,I\,\vec{y}\end{array}}{\Gamma \longrightarrow p\,\vec{t}}\,\mathrm{CI}\mathcal{R},\quad \text{where } p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$$

and there exist a parameter X^p such that $X^p\#\Omega$ and a reducibility candidate $(\mathcal{S} : I)$ such that $\Pi' \in \mathcal{S}$ and

$$\Pi_I[\vec{u}/\vec{y}] \in \big(\mathcal{S}\,\vec{u} \Rightarrow \mathbf{RED}_{B\,X^p\,\vec{u}}\big[\Omega, (\mathcal{S}, \Pi_I, I)/X^p\big]\big)\quad \text{for every } \vec{u}.$$

P6 Π ends with any other rule and its major premise derivations are in the parametric reducibility sets of the appropriate types.

We write $\mathbf{RED}_C$ for $\mathbf{RED}_C[\Omega]$, when $supp(\Omega) = \emptyset$. A derivation Π of type C is *reducible* if $\Pi \in \mathbf{RED}_C$.

Some comments and comparison with Girard's definition of parametric reducibility for System F [17] are in order, although our technical setting is somewhat different:

- Condition **P3** quantifies over ρ. This is needed to show that reducibility is closed under substitution (see Lemma 13). A similar quantification is used in the definition of reducibility for $FO\lambda^{\Delta\mathbb{N}}$ [22] for the same purpose. In the same clause, we also quantify over derivations in $\mathbf{RED}_{B[\rho]}[\Omega]$, but since $|B[\rho]| < |B \supset D|$, this quantification is legitimate and the definition is well-founded. Note also the similar quantification in **P4** and **P5**, where the parametric reducibility set $\mathbf{RED}_{p\,\vec{t}}[\Omega]$ is defined in terms of $\mathbf{RED}_{(B\,X^p\,\vec{t})}[\Omega]$. By Lemma 1 $|p\,\vec{t}| > |B\,X^p\,\vec{t}|$ so in both cases the set $\mathbf{RED}_{(B\,X^p\,\vec{t})}[\Omega]$ is already defined by induction. It is clear by inspection of the clauses that the definition of parametric reducibility is well-founded.

- **P2** and **P6** are needed to show that the notion of parametric reducibility is closed under left-rules, *id* and *mc*, i.e., conditions **CR3**–**CR5**. This is also a point where our definition of parametric reducibility diverges from a typical definition of reducibility in natural deduction (e.g., [17]), where closure under reduction for "neutral" terms is a derived property.
- **P4** (and dually **P5**) can be intuitively explained in terms of the second-order encoding of inductive definitions. To simplify presentation, we restrict to the propositional case, so, **P4** can be simplified as follows:

  Suppose Π ends with $\mathrm{I}\mathcal{R}$, i.e.,

$$\frac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow B\,X^p\end{array}}{\Gamma \longrightarrow p}\ \mathrm{I}\mathcal{R}, \quad \text{where } p \stackrel{\mu}{=} B\,p.$$

  Without loss of generality, assume that $X^p \# \Omega$: for every reducibility candidate $(\mathcal{S} : I)$, where I is a closed term of the same syntactic type as p, for every normalizable derivation Π_I of $B\,I \longrightarrow I$, if $\Pi_I \in (\mathbf{RED}_{B\,X^p}[\Omega, (\mathcal{S}, \Pi_I, I)/X^p] \Rightarrow \mathcal{S})$, then $mc(\Pi'[(\Pi_I, I)/X^p], \Pi_I) \in \mathcal{S}$.

  Note that in propositional Linc_i^-, the set

$$\mathbf{RED}_{B\,X^p}\big[\Omega, (\mathcal{S}, \Pi_I, I)/X^p\big] \quad \Rightarrow \quad \mathcal{S}$$

  is equivalent to

$$\mathbf{RED}_{B\,X^p \supset X^p}\big[\Omega, (\mathcal{S}, \Pi_I, I)/X^p\big],$$

  i.e., a set of reducible derivations of type $B\,I \supset I$. So, intuitively, Π' can be seen as a higher-order function that takes any function of type $B\,I \supset I$ (i.e., the derivation Π_I), and turns it into a derivation of type I (i.e., the derivation $mc(\Pi'[(\Pi_I, I)/X^p], \Pi_I)$), for all candidate $(\mathcal{S} : I)$. This intuitive reading matches the second-order interpretation of p, i.e., $\forall I.(B\,I \supset I) \supset I$, where the universal quantification is interpreted as the universal type constructor and $\supset$ is interpreted as the function type constructor in System F.

We shall now establish a list of properties of parametric reducibility sets. The main property that we are after is one that shows that a certain set of derivations formed using a family of parametric reducibility sets actually forms a reducibility candidate. This will be important later in constructing a reducibility candidate which acts as a co-inductive "witness" in the main cut elimination proof. The proofs of the following lemmas are mostly routine and rather tedious – they can be found in Appendix B.

Lemma 12. *If $\Pi \in \mathbf{RED}_C[\Omega]$ then Π is normalizable.*

Since every $\Pi \in \mathbf{RED}_C[\Omega]$ is normalizable, $nd(\Pi)$ is defined. This fact will be used implicitly in subsequent proofs, i.e., we shall do induction on $nd(\Pi)$ to prove properties of $\mathbf{RED}_C[\Omega]$.

Lemma 13. *If $\Pi \in \mathbf{RED}_C[\Omega]$ then for every substitution ρ, $\Pi[\rho] \in \mathbf{RED}_{C[\rho]}[\Omega]$.*

Lemma 14. *Let $\Omega = [\Omega', (\mathcal{R}, \Pi_S, S)/X^p]$. Let C be a formula such that $X^p \# C$. Then for every Π, $\Pi \in \mathbf{RED}_C[\Omega]$ if and only if $\Pi \in \mathbf{RED}_C[\Omega']$.*

Lemma 15. *Let Ω be a candidate substitution and S a closed term of syntactic type $\tau_1 \to \cdots \to \tau_n \to o$. Then the set*

$$\mathcal{R} = \big\{\Pi \;\big|\; \Pi \in \mathbf{RED}_{S\,\vec{u}}[\Omega] \text{ for some } \vec{u}\big\}$$

is a reducibility candidate of type $S[\Omega]$.

Lemma 16. *Let Ω be a candidate substitution and let X^p be a parameter such that $X^p \# \Omega$. Let S be a closed term of the same syntactic type as p and let*

$$\mathcal{R} = \big\{\Pi \;\big|\; \Pi \in \mathbf{RED}_{S\,\vec{u}}[\Omega] \text{ for some } \vec{u}\big\}.$$

Suppose $[\Omega, (\mathcal{R}, \Psi, S[\Omega])/X^p]$ is a candidate substitution, for some Ψ. Then

$$\mathbf{RED}_{C[S/X^p]}[\Omega] = \mathbf{RED}_C\big[\Omega, \big(\mathcal{R}, \Psi, S[\Omega]\big)/X^p\big].$$

5.4. Cut elimination

We shall now show that every derivation is reducible, hence every derivation can be normalized to a cut-free derivation. To prove this, we need a slightly more general lemma, which states that every derivation is in $\mathbf{RED}_C[\Omega]$ for a certain kind of candidate substitution Ω. The precise definition is given below.

Definition 16 *(Definitional closure).* A candidate substitution Ω is *definitionally closed* if for every $X^p \in supp(\Omega)$, if $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$ then either one of the following holds:

- $p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$, for some B and for every $\vec{u}$ of the appropriate syntactic types:
$$\Pi_S[\vec{u}/\vec{x}] \in \mathbf{RED}_{B\,X^p\,\vec{u}}[\Omega] \quad \Rightarrow \quad \mathcal{R}\,\vec{u}.$$
- $p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$, for some B and for every $\vec{u}$ of the appropriate syntactic types:
$$\Pi_S[\vec{u}/\vec{x}] \in \mathcal{R}\,\vec{u} \quad \Rightarrow \quad \mathbf{RED}_{B\,X^p\,\vec{u}}[\Omega].$$

The next two lemmas show that definitionally closed substitutions can be extended in a way that preserves definitional closure.

Lemma 17. *Let $\Omega = [\Omega', (\mathcal{R}, \Pi_S, S)/X^p]$ be a candidate substitution such that $p\,\vec{x} \stackrel{\mu}{=} B\,p\,\vec{x}$, Ω' is definitionally closed, and for every $\vec{u}$ of the same types as $\vec{x}$,*
$$\Pi_S[\vec{u}/\vec{x}] \in \mathbf{RED}_{B\,X^p\,\vec{u}}[\Omega] \quad \Rightarrow \quad \mathcal{R}\,\vec{u}.$$
Then Ω is definitionally closed.

Proof. Let $Y^q \in supp(\Omega)$. Suppose $\Omega(Y^q) = (\mathcal{S}, \Pi_I, I)$. We need to show that
$$\Pi_I[\vec{t}/\vec{x}] \in \mathbf{RED}_{B\,Y^q\,\vec{t}}[\Omega] \quad \Rightarrow \quad \mathcal{S}\,\vec{t} \tag{1}$$
for every $\vec{t}$ of the same types as $\vec{x}$. If $Y^q = X^p$, then this follows from the assumption of the lemma. Otherwise, $Y^q \in supp(\Omega')$, and by the definitional closure assumption on Ω', we have
$$\Pi_I[\vec{t}/\vec{x}] \in \mathbf{RED}_{B\,Y^q\,\vec{t}}\big[\Omega'\big] \quad \Rightarrow \quad \mathcal{S}\,\vec{t} \tag{2}$$
for every $\vec{t}$. Since $X^p \# (B\,Y^q\,\vec{t})$ (recall that definitions cannot contain occurrences of parameters), by Lemma 14 we have $\mathbf{RED}_{B\,Y^q\,\vec{t}}[\Omega'] = \mathbf{RED}_{B\,Y^q\,\vec{t}}[\Omega]$. The latter, together with (2), implies (1). □

Lemma 18. *Let $\Omega = [\Omega', (\mathcal{R}, \Pi_S, S)/X^p]$ be a candidate substitution such that $p\,\vec{x} \stackrel{\nu}{=} B\,p\,\vec{x}$, Ω' is definitionally closed, and for every $\vec{u}$ of the same types as $\vec{x}$,*
$$\Pi_S[\vec{u}/\vec{x}] \in \mathcal{R}\,\vec{u} \quad \Rightarrow \quad \mathbf{RED}_{B\,X^p\,\vec{u}}[\Omega].$$
Then Ω is definitionally closed.

Proof. Analogous to the proof of Lemma 17. □

We are now ready to state the main lemma for cut elimination.

Lemma 19. *Let Ω be a definitionally closed candidate substitution. Let Π be a derivation of $B_1, \dots, B_n, \Gamma \longrightarrow C$, and let*
$$\begin{array}{c} \Pi_1 \\ \Delta_1 \longrightarrow B_1[\Omega] \end{array} \quad \dots \quad \begin{array}{c} \Pi_n \\ \Delta_n \longrightarrow B_n[\Omega] \end{array}$$
where $n \geqslant 0$, be derivations in, respectively, $\mathbf{RED}_{B_1}[\Omega], \dots, \mathbf{RED}_{B_n}[\Omega]$. Then the derivation Ξ
$$\frac{\begin{array}{c} \Pi_1 \\ \Delta_1 \longrightarrow B_1[\Omega] \end{array} \quad \cdots \quad \begin{array}{c} \Pi_n \\ \Delta_n \longrightarrow B_n[\Omega] \end{array} \quad \begin{array}{c} \Pi[\Omega] \\ B_1[\Omega], \dots, B_n[\Omega], \Gamma[\Omega] \longrightarrow C[\Omega] \end{array}}{\Delta_1, \dots, \Delta_n, \Gamma[\Omega] \longrightarrow C[\Omega]}\ mc$$
is in $\mathbf{RED}_C[\Omega]$.

Proof. The proof is by induction on

$$\mathcal{M}(\Xi) = \left\langle \mathrm{ht}(\Pi), \sum_{i=1}^{n} |B_i|, \mathrm{ND}(\Xi) \right\rangle$$

where $\mathrm{ND}(\Xi)$ is the multiset $\{nd(\Pi_1), \dots, nd(\Pi_n)\}$ of normalization degrees of Π_1 to Π_n. The measure $\mathcal{M}$ can be well-ordered lexicographically. We shall refer to this ordering simply as $<$. Note that in the third component of $\mathcal{M}$, we use the measure $nd(\Pi_i)$ rather than simply the height of Π_i. This is imposed by the cut reduction rule $mc/\circ\mathcal{L}$ (see case **II.3** below).

The measure $\mathcal{M}$ is insensitive to the order in which Π_i is given, thus when we need to distinguish one of the Π_i, we shall refer to it as Π_1 without loss of generality. The derivation Ξ is in $\mathbf{RED}_C[\Omega]$ if all its reducts are in $\mathbf{RED}_C[\Omega]$.

CASE I: $n = 0$. In this case, Ξ reduces to $\Pi[\Omega]$, thus it is enough to show that $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$. This is proved by case analysis on C and on the last rule of Π.

I.1. Suppose $C = X^p\,\vec{t}$ for some parameter $X^p \in supp(\Omega)$ and some terms $\vec{t}$. Suppose $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$. Then there are several cases to consider, based on the last rule of Π. In all cases, we need to show that $\Pi[\Omega] \in \mathcal{R}\,\vec{t}$. Note that since $\Pi[\Omega]$ is of type $S\,\vec{t}$, $\Pi[\Omega] \in \mathcal{R}$ implies that $\Pi[\Omega] \in \mathcal{R}\,\vec{t}$. So in some cases we only need to show $\Pi[\Omega] \in \mathcal{R}$.

- Π ends with *init*: then $\Pi[\Omega]$ also ends with *init* and by **CR4**, $\Pi[\Omega] \in \mathcal{R}$.
- Π ends with *mc*: This follows from the induction hypothesis and Lemma 12.
- Π ends with $\mathrm{CI}\,\mathcal{L}_p$: Suppose Π ends with $\mathrm{CI}\,\mathcal{L}_p$ acting on a formula $Y^q\,\vec{u}$. If $Y^q \notin supp(\Omega)$, then this follows immediately from the induction hypothesis and **CR5**. If $Y^q \in supp(\Omega)$, then we use the same arguments as shown in **I.2** below to show that $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$.
- Π ends with *subst*or a left-rule other than $\mathrm{CI}\mathcal{L}_p$: Suppose the premise derivations of the rule are $\{\Psi_i\}_{i\in\mathbb{I}}$ for some index set $\mathbb{I}$, where each Ψ_i is of type C_i. Then $\Pi[\Omega]$ ends with the same left rule and has premise derivations $\{\Psi_i[\Omega]\}_{i\in\mathbb{I}}$. By the induction hypothesis $\Psi_i[\Omega] \in \mathbf{RED}_{C_i}[\Omega]$ for every $i \in \mathbb{I}$, and by Lemma 12 each $\Psi_i[\Omega]$ is also normalizable. The latter implies that $\Pi[\Omega]$ is normalizable. Note that if Ψ_i is a major premise derivation, it must be of type $X^p\,\vec{u}$ for some $\vec{u}$, and we have $\Psi_i[\Omega] \in \mathcal{R}$. Therefore, by **CR5**, we have that $\Pi[\Omega] \in \mathcal{R}$.
- Suppose Π ends with $\mathrm{I}\,\mathcal{R}_p$:

$$\cfrac{\begin{matrix}\Pi' \\ \Gamma \longrightarrow D\,X^p\,\vec{t}\end{matrix}}{\Gamma \longrightarrow X^p\vec{t}}\ \mathrm{I}\,\mathcal{R}_p$$

where $p\,\vec{x} \stackrel{\mu}{=} D\,p\,\vec{x}$. Then $\Pi[\Omega] = mc(\Pi'[\Omega], \Pi_S[\vec{t}/\vec{x}])$. From the induction hypothesis we have that $\Pi'[\Omega] \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega]$. This, together with the definitional closure of Ω, implies that $\Pi[\Omega]$ is indeed in $\mathcal{R}\,\vec{t}$.

I.2. Suppose $C \neq X^p\,\vec{t}$ for any $X^p \in supp(\Omega)$ and terms $\vec{t}$, and Π ends with a left rule, *init*, or *mc*. This follows mostly straightforwardly from the induction hypothesis and Lemma 12. The only interesting case is when Π ends with $\mathrm{CI}\,\mathcal{L}_p$ on some $Y^q\,\vec{u}$ such that $Y^q \in supp(\Omega)$, i.e., Π takes the form

$$\cfrac{\begin{matrix}\Pi' \\ D\,Y^q\,\vec{u}, \Gamma \longrightarrow C\end{matrix}}{Y^q\,\vec{u}, \Gamma \longrightarrow C}\ \mathrm{CI}\,\mathcal{L}_p.$$

Suppose $\Omega(Y^q) = (\mathcal{R}, \Pi_S, S)$. Then $\Pi[\Omega] = mc(mc(\mathrm{Id}_{S\,\vec{u}}, \Pi_S[\vec{u}/\vec{x}]), \Pi'[\Omega])$. By **CR4** we have that $\mathrm{Id}_{S\,\vec{u}} \in \mathcal{R}$, so by the definitional closure of Ω and **CR3**, we have $mc(\mathrm{Id}_{S\,\vec{u}}, \Pi_S[\vec{u}/\vec{x}]) \in \mathbf{RED}_{D\,S\,\vec{u}}[\Omega]$. Since $\mathrm{ht}(\Pi') < \mathrm{ht}(\Pi)$, and $\Pi[\Omega] = mc(mc(\mathrm{Id}_{S\,\vec{u}}, \Pi_S[\vec{u}/\vec{x}]), \Pi'[\Omega])$, by the induction hypothesis we have $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$.

I.3. Suppose $C \neq X^p\,\vec{t}$ for any parameter $X^p \in supp(\Omega)$ and any terms $\vec{t}$, and Π ends with a right-rule. We show here the non-trivial subcases:

I.3.a. Suppose Π ends with $\supset\mathcal{R}$, as shown below left. Then $\Pi[\Omega]$ is as shown below right:

$$\cfrac{\begin{matrix}\Pi' \\ \Gamma, C_1 \longrightarrow C_2\end{matrix}}{\Gamma \longrightarrow C_1 \supset C_2}\ \supset\mathcal{R}, \qquad \cfrac{\begin{matrix}\Pi'[\Omega] \\ \Gamma[\Omega], C_1[\Omega] \longrightarrow C_2[\Omega]\end{matrix}}{\Gamma[\Omega] \longrightarrow C_1[\Omega] \supset C_2[\Omega]}\ \supset\mathcal{R}.$$

To show $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$, we need to show that $\Pi[\Omega]$ is normalizable and that

$$\Pi'[\Omega][\theta] \in \mathbf{RED}_{C_1[\theta]}[\Omega] \quad\Rightarrow\quad \mathbf{RED}_{C_2[\theta]}[\Omega] \tag{3}$$

for every θ. Since $\mathrm{ht}(\Pi') < \mathrm{ht}(\Pi)$, by the induction hypothesis $\Pi'[\Omega] \in \mathbf{RED}_{C_2}[\Omega]$. Normalizability of $\Pi[\Omega]$ then follows immediately from Lemma 12. It remains to show that Statement (3) holds. Let Ψ be a derivation in $\mathbf{RED}_{C_1[\theta]}[\Omega]$. Let $\Xi_1 = mc(\Psi, \Pi'[\Omega][\theta])$. Note that since parameter substitution commutes with eigenvariable substitution $\Pi'[\Omega][\theta] = \Pi'[\theta][\Omega]$.

Since $\mathrm{ht}(\Pi'[\theta]) \leqslant \mathrm{ht}(\Pi') < \mathrm{ht}(\Pi)$ (Lemma 3), by induction hypothesis we have $\Xi_1 \in \mathbf{RED}_{C_2[\theta]}[\Omega]$. Since (3) holds for an arbitrary θ, by Definition 15 $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$.

I.3.b. Suppose Π ends with $\mathrm{I}\mathcal{R}$, as shown below left, where $p\,\vec{x} \stackrel{\mu}{=} D\,p\,\vec{x}$. We can assume w.l.o.g. that $X^p \# \Omega$. Then $\Pi[\Omega]$ is as shown below right:

$$\dfrac{\begin{matrix}\Pi' \\ \Gamma \longrightarrow D\,X^p\,\vec{t}\end{matrix}}{\Gamma \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}, \qquad \dfrac{\begin{matrix}\Pi'[\Omega] \\ \Gamma[\Omega] \longrightarrow D\,X^p\,\vec{t}\end{matrix}}{\Gamma[\Omega] \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}.$$

To show that $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$, we need to show that $\Pi[\Omega]$ is normalizable (this follows easily from the induction hypothesis and Lemma 12) and that

$$mc\big(\Pi'[\Omega]\big[(\Pi_S, S)/X^p\big], \Pi_S[\vec{t}/\vec{x}]\big) \in \mathcal{R}\,\vec{t} \tag{4}$$

for every candidate $(\mathcal{R} : S)$ and every Π_S that satisfies:

$$\Pi_S[\vec{u}/\vec{x}] \in \mathbf{RED}_{D\,X^p\,\vec{u}}\big[\Omega, (\mathcal{R}, \Pi_S, S)/X^p\big] \quad \Rightarrow \quad \mathcal{R}\,\vec{u} \quad \text{for every } \vec{u}. \tag{5}$$

Let $\Omega' = [\Omega, (\mathcal{R}, \Pi_S, S)/X^p]$. Note that since $X^p \# \Omega$, we have, by Lemma 6(2), $\Pi'[\Omega][(\Pi_S, S)/X^p] = \Pi'[\Omega']$. So statement (4) can be rewritten to

$$mc\big(\Pi'[\Omega'], \Pi_S[\vec{t}/\vec{x}]\big) \in \mathcal{R}\,\vec{t}. \tag{6}$$

By Lemma 17, we have that Ω' is definitionally closed. Therefore we can apply the induction hypothesis to Π' and Ω', obtaining $\Pi'[\Omega'] \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega']$. This, together with the definitional closure of Ω' and assumption (5), immediately implies Statement (6) above; hence $\Pi[\Omega]$ is indeed in $\mathbf{RED}_C[\Omega]$.

I.3.c. Suppose Π ends with $\mathrm{CI}\mathcal{R}$, as shown below left, where $p\,\vec{y} \stackrel{\nu}{=} D\,p\,\vec{y}$. Let $S' = S[\Omega]$. Then $\Pi[\Omega]$ is as shown below right:

$$\dfrac{\begin{matrix}\Pi' \\ \Gamma \longrightarrow S\,\vec{t}\end{matrix} \quad \begin{matrix}\Pi_S \\ S\,\vec{x} \longrightarrow D\,S\,\vec{x}\end{matrix}}{\Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}, \qquad \dfrac{\begin{matrix}\Pi'[\Omega] \\ \Gamma[\Omega] \longrightarrow S'\,\vec{t}\end{matrix} \quad \begin{matrix}\Pi_S[\Omega] \\ S'\vec{x} \longrightarrow D\,S'\,\vec{x}\end{matrix}}{\Gamma[\Omega] \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}.$$

Note that $\Pi[\Omega]$ is normalizable by the induction hypothesis and Lemma 12. To show that $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$ it remains to show that there exists a reducibility candidate $(\mathcal{R} : S')$ such that

(a) $\Pi'[\Omega] \in \mathcal{R}$ and
(b) $\Pi_S[\Omega][\vec{u}/\vec{x}] \in \mathcal{R}\,\vec{u} \Rightarrow \mathbf{RED}_{D\,X^p\,\vec{u}}[\Omega, (\mathcal{R}, \Pi_S[\Omega], S')/X^p]$ for a new $X^p \# \Omega$, and for every $\vec{u}$.

Let $\mathcal{R} = \{\Psi \mid \Psi \in \mathbf{RED}_{S\,\vec{u}}[\Omega]\}$. By Lemma 15 $\mathcal{R}$ is a reducibility candidate of type S'. By the induction hypothesis we have $\Pi'[\Omega] \in \mathcal{R}$, so $\mathcal{R}$ satisfies (a). Since substitution does not increase the height of derivations, we have that $\mathrm{ht}(\Pi_S[\vec{u}/\vec{x}]) \leqslant \mathrm{ht}(\Pi_S)$, and therefore, by applying the induction hypothesis to $\Pi_S[\vec{x}/\vec{u}]$, we have $mc(\Psi, \Pi_S[\Omega][\vec{u}/\vec{x}]) \in \mathbf{RED}_{D\,S\,\vec{u}}[\Omega]$ for every $\Psi \in \mathbf{RED}_{S\,\vec{u}}[\Omega]$. In other words,

$$\Pi_S[\Omega][\vec{u}/\vec{x}] \in \mathbf{RED}_{S\,\vec{u}}[\Omega] \quad \Rightarrow \quad \mathbf{RED}_{D\,S\,\vec{u}}[\Omega].$$

Since $\mathbf{RED}_{S\,\vec{u}}[\Omega] = \mathcal{R}\,\vec{u}$, the above statement can be rewritten to

$$\Pi_S[\Omega][\vec{u}/\vec{x}] \in \mathcal{R}\,\vec{u} \quad \Rightarrow \quad \mathbf{RED}_{D\,S\,\vec{u}}[\Omega].$$

By Lemma 16 $\mathbf{RED}_{D\,S\,\vec{u}}[\Omega] = \mathbf{RED}_{D\,X^p\,\vec{u}}[\Omega, (\mathcal{R}, \Pi_S[\Omega], S')/X^p]$, which means that $\mathcal{R}$ indeed satisfies condition (b) above, and therefore $\Pi[\Omega] \in \mathbf{RED}_C[\Omega]$.

CASE II: $n > 0$. To show that $\Xi \in \mathbf{RED}_C[\Omega]$, we need to show that all its reducts are in $\mathbf{RED}_C[\Omega]$ and that Ξ is normalizable. The latter follows from the former by Lemma 12 and Definition 10, so in the following we need only to show the former.

Note that in this case, we do not need to distinguish cases based on whether C is headed by a parameter or not. To see why, suppose $C = X^p\,\vec{t}$ for some parameter X^p. If $X^p \notin supp(\Omega)$, then to show $\Xi \in \mathbf{RED}_C[\Omega]$ we need to show that it is normalizable, which means that we need to show that all its reducts are normalizable. But since all reducts of Ξ have the same type $X^p\,\vec{t}$, showing their normalizability amounts to the same thing as showing that they are in $\mathbf{RED}_C[\Omega]$. If $X^p \in supp(\Omega)$, then we need to show that $\Xi \in \mathcal{R}$. Then by **CR3**, it is enough to show that all reducts of Ξ are in $\mathcal{R}$, which is the same as showing that all reducts of Ξ are in $\mathbf{RED}_C[\Omega]$.

Since the applicable reduction rules to Ξ are driven by the shape of $\Pi[\Omega]$, and since $\Pi[\Omega]$ is determined by Π, we shall perform case analysis on Π in order to determine the possible reduction rules that apply to Ξ, and show in each case that the reduct of Ξ is in the same parametric reducibility set. There are several main cases depending on whether Π ends with a rule acting on a cut formula B_i or not. Again, when we refer to B_i, without loss of generality, we assume $i = 1$.

In the following, we say that an instance of $\mathrm{CI}\mathcal{L}_p$ is *trivial* if it applies to a formula $Y^q\,\vec{u}$ for some $\vec{u}$, but $Y^q \notin supp(\Omega)$. Otherwise, it is *non-trivial*.

II.1. Suppose Π ends with a left rule other than $\mathrm{c}\mathcal{L}$, $\mathrm{w}\mathcal{L}$, and a non-trivial $\mathrm{CI}\mathcal{L}_p$ acting on B_1, and suppose Π_1 ends with a right-introduction rule. There are several subcases depending on the logical rules that are applied to B_1. We show here the non-trivial cases:

$\boxed{\supset\mathcal{R}/\supset\mathcal{L}}$ Suppose Π_1 and Π are

$$\dfrac{\begin{matrix}\Pi_1' \\ \Delta_1, B_1'[\Omega] \longrightarrow B_1''[\Omega]\end{matrix}}{\Delta_1 \longrightarrow B_1'[\Omega] \supset B_1''[\Omega]}\supset\mathcal{R}, \qquad \dfrac{\begin{matrix}\Pi' \\ B_2,\dots,B_n\Gamma \longrightarrow B_1'\end{matrix} \quad \begin{matrix}\Pi'' \\ B_1'', B_2,\dots B_n,\Gamma \longrightarrow C\end{matrix}}{B_1' \supset B_1'', B_2,\dots,B_n,\Gamma \longrightarrow C}\supset\mathcal{L}.$$

Let $\Xi_1 = mc(\Pi_2,\dots,\Pi_n,\Pi'[\Omega])$. Then $\Xi_1 \in \mathbf{RED}_{B_1'}[\Omega]$ by induction hypothesis since $\mathrm{ht}(\Pi') < \mathrm{ht}(\Pi)$ and therefore $\mathcal{M}(\Xi_1) < \mathcal{M}(\Xi)$. Since $\Pi_1 \in \mathbf{RED}_{B_1}[\Omega]$, by Definition 15 we have

$$\Pi_1' \in \mathbf{RED}_{B_1'}[\Omega] \quad\Rightarrow\quad \mathbf{RED}_{B_1''}[\Omega]$$

and therefore the derivation $\Xi_2 = mc(\Xi_1,\Pi_1')$ of $\Delta_1,\dots,\Delta_n,\Gamma[\Omega] \longrightarrow B_1''[\Omega]$ is in $\mathbf{RED}_{B_1''}[\Omega]$. Let $\Xi_3 = mc(\Xi_2,\Pi_2,\dots,\Pi_n,\Pi''[\Omega])$. The reduct of Ξ in this case is the derivation Ξ':

$$\dfrac{\begin{matrix}\Xi_3 \\ \Delta_1,\dots,\Delta_n,\Gamma[\Omega],\Delta_2,\dots,\Delta_n,\Gamma[\Omega] \longrightarrow C[\Omega]\end{matrix}}{\Delta_1,\dots,\Delta_n,\Gamma[\Omega] \longrightarrow C[\Omega]}\,\mathrm{c}\mathcal{L}.$$

By the induction hypothesis we have $\Xi_3 \in \mathbf{RED}_C[\Omega]$, and therefore it is normalizable by Lemma 12. By Definition 10 this means that Ξ' is normalizable and by Definition 15 $\Xi' \in \mathbf{RED}_C[\Omega]$.

$\boxed{\forall\mathcal{L}/\forall\mathcal{R}}$ Suppose Π_1 and Π are

$$\dfrac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow B_1'[\Omega][y/x]\end{matrix}}{\Delta_1 \longrightarrow \forall x.B_1'[\Omega]}\forall\mathcal{R}, \qquad \dfrac{\begin{matrix}\Pi' \\ B_1'[t/x], B_2,\dots,B_n,\Gamma \longrightarrow C\end{matrix}}{\forall x.B_1', B_2,\dots,B_n,\Gamma \longrightarrow C}\forall\mathcal{L}.$$

The reduct of Ξ in this case is

$$\Xi' = mc\big(\Pi_1'[t/y],\Pi_2,\dots,\Pi_n,\Pi'[\Omega]\big).$$

Since $\Pi_1' \in \mathbf{RED}_{B_1'[y/x]}[\Omega]$, by Lemma 13 we have $\Pi_1'[t/y] \in \mathbf{RED}_{B_1'[t/x]}[\Omega]$. Note that $\mathrm{ht}(\Pi') < \mathrm{ht}(\Pi)$, so we can apply the induction hypothesis to obtain $\Xi' \in \mathbf{RED}_C[\Omega]$.

$\boxed{\mathrm{eq}\mathcal{R}/\mathrm{eq}\mathcal{L}}$ Suppose Π_1 and Π are

$$\dfrac{}{\Delta_1 \longrightarrow s = t}\,\mathrm{eq}\mathcal{R}, \qquad \dfrac{\left\{\begin{matrix}\Pi^\rho \\ B_2[\rho],\dots,B_n[\rho],\Gamma[\rho] \longrightarrow C[\rho]\end{matrix}\right\}_{\rho\in\mathbb{U}(s,t)}}{s = t, B_2,\dots,B_n,\Gamma \longrightarrow C}\,\mathrm{eq}\mathcal{L}.$$

Note that in this case s must be the same term as t, and therefore $\mathbb{U}(s,t) = \mathbb{S}$. Let Π' be the derivation $subst(\{\Pi^\rho\}_{\rho\in\mathbb{S}})$ of the sequent $B_2,\dots,B_n,\Gamma \longrightarrow C$ and let $\Xi_1 = mc(\Pi_2,\dots,\Pi_n,\Pi'[\Omega])$. Since $\mathrm{ht}(\Pi') = \mathrm{ht}(\Pi)$ and since the total size of the cut formulas in Ξ_1 is smaller than in Ξ, by the induction hypothesis we have $\Xi_1 \in \mathbf{RED}_C[\Omega]$. Then the reduct of Ξ in this case is the derivation Ξ':

$$\dfrac{\begin{matrix}\Xi_1 \\ \Delta_2,\dots,\Delta_n,\Gamma \longrightarrow C\end{matrix}}{\Delta_1,\Delta_2,\dots,\Delta_n,\Gamma \longrightarrow C}\,\mathrm{w}\mathcal{L}$$

which is also in $\mathbf{RED}_C[\Omega]$ by the definition of parametric reducibility.

$\boxed{\mathrm{I}\mathcal{R}/\mathrm{I}\mathcal{L}}$ Suppose Π_1 and Π are the derivations

$$\dfrac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow D\,X^p\,\vec{t}\end{matrix}}{\Delta_1 \longrightarrow p\,\vec{t}}\,\mathrm{I}\mathcal{R}, \qquad \dfrac{\begin{matrix}\Pi_S \\ D\,S\,\vec{x} \longrightarrow S\,\vec{x}\end{matrix} \quad \begin{matrix}\Pi' \\ S\,\vec{t}, B_2,\dots,B_n,\Gamma \longrightarrow C\end{matrix}}{p\,\vec{t}, B_2,\dots,B_n,\Gamma \longrightarrow C}\,\mathrm{I}\mathcal{L}$$

where $p\,\vec{y} \stackrel{\mu}{=} D\,p\,\vec{y}$ and X^p is a new parameter not occurring in the end sequent of Π_1 (we can assume w.l.o.g. that $X^p \# \Omega$ and that it does not occur also in the end sequent of Π). Then $\Pi[\Omega]$ is the derivation, where $S' = S[\Omega]$:

$$\dfrac{\begin{matrix}\Pi_S[\Omega] \\ D\,S'\,\vec{x} \longrightarrow S'\,\vec{x}\end{matrix} \quad \begin{matrix}\Pi'[\Omega] \\ S'\,\vec{t}, B_2[\Omega], \ldots, B_n[\Omega], \Gamma[\Omega] \longrightarrow C[\Omega]\end{matrix}}{p\,\vec{t}, B_2[\Omega], \ldots, B_n[\Omega], \Gamma[\Omega] \longrightarrow C[\Omega]}\ \mathrm{I}\mathcal{L}.$$

Let $\Xi_1 = mc(\Pi_1'[(\Pi_S[\Omega], S')/X^p], \Pi_S[\Omega][\vec{t}/\vec{x}])$. Then the reduct of Ξ in this case is the derivation

$$\Xi' = mc\big(\Xi_1, \Pi_2, \ldots, \Pi_n, \Pi'[\Omega]\big).$$

Note that $\Pi_S[\vec{u}/\vec{x}][\Omega] = \Pi_S[\Omega][\vec{u}/\vec{x}]$ by Lemma 5. Since

$$\big(\Pi_S[\vec{u}/\vec{x}]\big) \leqslant \mathrm{ht}(\Pi_S) < \mathrm{ht}(\Pi),$$

we have that for every derivation $\Psi \in \mathbf{RED}_{D\,S\,\vec{u}}[\Omega]$,

$$\mathcal{M}\big(mc\big(\Psi, \Pi_S[\vec{u}/\vec{x}][\Omega]\big)\big) = \big\langle\big(\Pi_S[\vec{u}/\vec{x}]\big), |D\,S\,\vec{u}|, \big\{nd(\Psi)\big\}\big\rangle < \mathcal{M}(\Xi).$$

So by the induction hypothesis we have that, for every $\Psi \in \mathbf{RED}_{D\,S\,\vec{u}}[\Omega]$,

$$mc\big(\Psi, \Pi_S[\Omega][\vec{u}/\vec{x}]\big) = mc\big(\Psi, \Pi_S[\vec{u}/\vec{x}][\Omega]\big) \in \mathbf{RED}_{S\,\vec{u}}[\Omega].$$

In other words, we have:

$$\Pi_S[\Omega][\vec{u}/\vec{x}] \in \mathbf{RED}_{D\,S\,\vec{u}}\,[\Omega] \quad \Rightarrow \quad \mathbf{RED}_{S\,\vec{u}}\,[\Omega]. \tag{7}$$

Let $\mathcal{R} = \{\Psi \mid \Psi \in \mathbf{RED}_{S\,\vec{u}}\,[\Omega]$ for some $\vec{u}\}$. Then by Lemma 15 $\mathcal{R}$ is a reducibility candidate of type S'. Moreover, by Lemma 16 we have

$$\mathbf{RED}_{D\,S\,\vec{u}}[\Omega] = \mathbf{RED}_{D\,X^p\,\vec{u}}\big[\Omega, \big(\mathcal{R}, \Pi_S[\Omega], S'\big)/X^p\big].$$

This, together with Statement (7) above, implies that for every $\vec{u}$:

$$\Pi_S[\Omega][\vec{u}/\vec{x}] \in \mathbf{RED}_{D\,X^p\,\vec{u}}\big[\Omega, \big(\mathcal{R}, \Pi_S[\Omega], S'\big)/X^p\big] \quad \Rightarrow \quad \mathcal{R}\,\vec{u}. \tag{8}$$

Since $\Pi_1 \in \mathbf{RED}_{p\,\vec{t}}\,[\Omega]$, it follows from Definition 15 that for every reducibility candidate $(\mathcal{S} : I)$ and Π_I such that

$$\Pi_I[\vec{u}/\vec{x}] \in \mathbf{RED}_{D\,X^p\,\vec{u}}\big[\Omega, (\mathcal{S}, \Pi_I, I)/X^p\big] \quad \Rightarrow \quad \mathcal{S}\,\vec{u} \quad \text{for every } \vec{u},$$

we have $mc(\Pi_1'[(\Pi_I, I)/X^p], \Pi_I[\vec{t}/\vec{x}]) \in \mathcal{S}\,\vec{t}$. Substituting $\mathcal{R}$ for $\mathcal{S}$, $\Pi_S[\Omega]$ for Π_I and S' for I, and using Statement (8) above we obtain:

$$\Xi_1 = mc\big(\Pi_1'\big[\big(\Pi_S[\Omega], S'\big)/X^p\big], \Pi_S[\Omega][\vec{t}/\vec{x}]\big) \in \mathcal{R}\,\vec{t} = \mathbf{RED}_{S\,\vec{t}}\,[\Omega].$$

Since $\mathrm{ht}(\Pi') < \mathrm{ht}(\Pi)$, we can then apply the induction hypothesis to conclude that $\Xi' \in \mathbf{RED}_C[\Omega]$.

$\boxed{\mathrm{CI}\mathcal{R}/\mathrm{CI}\mathcal{L}}$ Suppose Π_1 and Π are

$$\dfrac{\begin{matrix}\Pi_1' \\ \Delta_1 \longrightarrow S\,\vec{t}\end{matrix} \quad \begin{matrix}\Pi_S \\ S\,\vec{x} \longrightarrow D\,S\,\vec{x}\end{matrix}}{\Delta_1 \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}, \qquad \dfrac{\begin{matrix}\Pi' \\ D\,X^p\,\vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{matrix}}{p\,\vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{CI}\mathcal{L}$$

where $p\,\vec{y} \stackrel{\nu}{=} D\,p\,\vec{y}$ and X^p is a parameter not already occurring in the end sequent of Π (and w.l.o.g. assume also $X^p \# \Omega$ and X^p not occurring in Δ_i or B_i). Then $\Pi[\Omega]$ is

$$\dfrac{\begin{matrix}\Pi'[\Omega] \\ D\,X^p\,\vec{t}, B_2[\Omega], \ldots, B_n, \Gamma[\Omega] \longrightarrow C[\Omega]\end{matrix}}{p\,\vec{t}, B_2[\Omega], \ldots, B_n, \Gamma[\Omega] \longrightarrow C[\Omega]}\ \mathrm{CI}\mathcal{L}.$$

Since $\Pi_1 \in \mathbf{RED}_{p\,\vec{t}}\,[\Omega]$, by Definition 15 there exists a reducibility candidate $(\mathcal{R} : S)$ such that $\Pi_1' \in \mathcal{R}$ and such that for every $\vec{u}$,

$$\Pi_S[\vec{u}/\vec{x}] \in \mathcal{R}\,\vec{u} \quad \Rightarrow \quad \mathbf{RED}_{D\,X^p\,\vec{u}}\big[\Omega, (\mathcal{R}, \Pi_S, S)/X^p\big].$$

Let $\Omega' = [\Omega, (\mathcal{R}, \Pi_S, S)/X^p]$. Then by Lemma 18 Ω' is definitionally closed. Let $\Xi_1 = mc(\Pi_1', \Pi_S[\vec{t}/\vec{x}])$. By the definitional closure of Ω', we have that $\Xi_1 \in \mathbf{RED}_{D\,X^p\,\vec{t}}\,[\Omega']$. The reduct of Ξ in this case is the derivation

$$\Xi' = mc\big(\Xi_1, \Pi_2, \ldots, \Pi_n, \Pi'\big[\Omega'\big]\big).$$

Note that, since X^p does not occur in Δ_i or B_i, by Lemma 14 we have that

$$\Pi_i \in \mathbf{RED}_{B_i}[\Omega] = \mathbf{RED}_{B_i}\left[\Omega'\right]$$

for every $i \in \{2, \ldots, n\}$. Therefore, by induction hypothesis, we have that $\Xi' \in \mathbf{RED}_C[\Omega']$. But since X^p is also new for C, we have $\mathbf{RED}_C[\Omega'] = \mathbf{RED}_C[\Omega]$, and therefore $\Xi' \in \mathbf{RED}_C[\Omega]$.

II.2. Π ends with a left rule other than c$\mathcal{L}$, w$\mathcal{L}$, and a non-trivial instance of CI$\mathcal{L}_p$, acting on B_1, and Π_1 ends with a left-rule or *subst*.

Note that in these cases the reducts always end with a left-rule. The proof for the following cases abides to the same pattern: we first establish that the premise derivations of the reduct are either normalizable or in certain reducibility sets. We then proceed to show that the reduct itself is reducible by applying the closure conditions of reducibility under applications of left-rules. For the latter, we distinguish three cases depending on C: If $C = X^p\vec{t}$ for some $X^p \in supp(\Omega)$, then closure under left-rules is guaranteed by **C5**; if $X^p \notin supp(\Omega)$, then we need to show that the reduct is normalizable, while the closure condition under left-rules is guaranteed by the definition of normalizability. Otherwise, C is not headed by any parameter; in this case, the closure condition follows from **P6**. We shall explicitly do these case analysis in one of the subcases below, but will otherwise leave them implicit. We show the non-trivial subcases only; other cases can be proved by straightforward applications of the induction hypothesis.

$\boxed{\supset\mathcal{L}/\circ\mathcal{L}}$ Suppose Π_1 is

$$\frac{\begin{array}{c}\Pi_1' \\ \Delta_1' \longrightarrow D_1\end{array} \quad \begin{array}{c}\Pi_1'' \\ D_2, \Delta_1' \longrightarrow B_1[\Omega]\end{array}}{D_1 \supset D_2, \Delta_1' \longrightarrow B_1[\Omega]} \supset\mathcal{L}.$$

Since $\Pi_1 \in \mathbf{RED}_{B_1}[\Omega]$, it follows from Definitions 15 and 10 that Π_1' is normalizable and $\Pi_1'' \in \mathbf{RED}_{B_1}[\Omega]$. Let $\Xi_1 = mc(\Pi_1'', \Pi_2, \ldots, \Pi_n, \Pi[\Omega])$. Since $nd(\Pi_1'') < nd(\Pi_1)$, and therefore $\mathcal{M}(\Xi_1) < \mathcal{M}(\Xi)$, by the induction hypothesis $\Xi_1 \in \mathbf{RED}_C[\Omega]$. The reduct of Ξ in this case is the derivation Ξ':

$$\frac{\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1' \longrightarrow D_1\end{array}}{\Delta_1', \ldots, \Gamma[\Omega] \longrightarrow D_1}\,\mathrm{w}\mathcal{L} \quad \begin{array}{c}\Xi_1 \\ D_2, \Delta_1', \Delta_2, \ldots, \Gamma[\Omega] \longrightarrow C[\Omega]\end{array}}{D_1 \supset D_2, \Delta_1', \Delta_2, \ldots, \Gamma[\Omega] \longrightarrow C[\Omega]} \supset\mathcal{L}.$$

Since Π_1' is normalizable, by Definition 10 the left premise derivation of Ξ' is normalizable and since reducibility implies normalizability (Lemma 12), the right premise is also normalizable, hence Ξ' is normalizable. Now to show $\Xi' \in \mathbf{RED}_C[\Omega]$, we distinguish three cases based on C:

- Suppose $C = X^p\vec{t}$ for some $X^p \in supp(\Omega)$ and $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$. Then we need to show that $\Xi' \in \mathcal{R}\vec{t}$. This follows from Definition 13, more specifically, from **CR5** and the fact that $\Xi_1 \in \mathbf{RED}_C[\Omega] = \mathcal{R}\vec{t}$.
- Suppose $C = X^p\vec{t}$ but $X^p \notin supp(\Omega)$. Then we need to show that Ξ' is normalizable. But this follows immediately from the normalizability of both of its premise derivations.
- Suppose $C \neq X^p\vec{t}$ for any parameter X^p and any terms $\vec{t}$. Since $\Xi_1 \in \mathbf{RED}_C[\Omega]$, by Definition 15 we have $\Xi' \in \mathbf{RED}_C[\Omega]$.

$\boxed{\mathrm{eq}\mathcal{L}/\circ\mathcal{L}}$ Suppose $\Pi_1 = \mathrm{eq}\mathcal{L}(\{\Pi_1^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where eq$\mathcal{L}$ introduces an equation $s = t$ in Δ_1. Let $\Xi^\rho = mc(\Pi_1^\rho, \Pi_2[\rho], \ldots, \Pi_n[\rho], \Pi[\rho][\Omega])$. Then the reduct of Ξ is the derivation $\Xi' = \mathrm{eq}\mathcal{L}(\{\Xi^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where eq$\mathcal{L}$ is applied to the same equation $s = t$ in Δ_1. Since $nd(\Pi_1^\rho) < nd(\Pi_1)$ (in the case where $\mathbb{U}(s,t)$ is infinite, the measure $nd(\Pi_1)$ is a limit ordinal of all $nd(\Pi^\rho)$), and the other measures are non-increasing, we have $\Xi^\rho \in \mathbf{RED}_{C[\rho]}[\Omega]$ by the induction hypothesis. Hence, $\Xi' \in \mathbf{RED}_C[\Omega]$ by the definition of parametric reducibility.

$\boxed{\mathrm{I}\mathcal{L}/\circ\mathcal{L}}$ Suppose Π_1 is

$$\frac{\begin{array}{c}\Pi_S \\ D\,S\,\vec{x} \longrightarrow S\,\vec{x}\end{array} \quad \begin{array}{c}\Pi_1' \\ S\,\vec{t}, \Delta_1' \longrightarrow B_1[\Omega]\end{array}}{p\,\vec{t}, \Delta_1' \longrightarrow B_1[\Omega]}\,\mathrm{I}\mathcal{L}.$$

Since $\Pi_1 \in \mathbf{RED}_{B_1}[\Omega]$, we have that Π_S is normalizable and $\Pi_1' \in \mathbf{RED}_{B_1}[\Omega]$. Let $\Xi_1 = mc(\Pi_1', \Pi_2, \ldots, \Pi_n, \Pi[\Omega])$. Then $\Xi_1 \in \mathbf{RED}_{B_1}[\Omega]$ by the induction hypothesis, since $nd(\Pi_1') < nd(\Pi_1)$. Therefore the reduct of Ξ

$$\frac{\begin{array}{cc}\Pi_S & \Xi_1 \\ D\,S\,\vec{x} \longrightarrow S\,\vec{x} & S\,\vec{u}, \Delta'_1, \ldots, \Delta_n, \Gamma[\Omega] \longrightarrow C[\Omega]\end{array}}{p\,\vec{u}, \Delta'_1, \ldots, \Delta_n, \Gamma[\Omega] \longrightarrow C[\Omega]}\ \mathrm{I}\mathcal{L}$$

is also in $\mathbf{RED}_C[\Omega]$.

II.3. Π ends with a left rule other than $\mathrm{c}\mathcal{L}$, $\mathrm{w}\mathcal{L}$ and a non-trivial instance of $\mathrm{CI}\mathcal{L}_p$, acting on B_1, and Π_1 ends with *mc* or *init*. Since the *init* case is trivial, we show here the case where Π_1 ends with *mc*; here, Ξ reduces to $\Xi' = mc(\Pi'_1, \Pi_2, \ldots, \Pi_n, \Pi[\Omega])$ where Π'_1 is a reduct of Π_1, hence $nd(\Pi'_1) < nd(\Pi_1)$ and therefore $\mathcal{M}(\Xi') < \mathcal{M}(\Xi)$. By the induction hypothesis $\Xi' \in \mathbf{RED}_C[\Omega]$. Note that cut reductions may increase the height of derivations, so in general $|\Pi'_1| < |\Pi_1|$ may not hold. This is why we use the degree of normalizations, rather than the height of derivations, in the third component of $\mathcal{M}$.

II.4. Suppose Π ends with a non-trivial application of $\mathrm{CI}\,\mathcal{L}_p$ on B_1. That is, $B_1 = X^p\,\vec{t}$, for some $X^p \in supp(\Omega)$ and some $\vec{t}$, and Π is

$$\frac{\begin{array}{c}\Pi' \\ D\,X^p\,\vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{X^p\,\vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{CI}\,\mathcal{L}_p$$

where $p\,\vec{x} \stackrel{\nu}{=} D\,p\,\vec{x}$. Suppose $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$. Then $\Pi[\Omega]$ is

$$mc\big(mc\big(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}]\big), \Pi'[\Omega]\big).$$

Let $\Xi_1 = mc(\Pi_1, mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}]))$. Ξ_1 has exactly one reduct, that is,

$$\Xi_2 = mc\big(mc(\Pi_1, \mathrm{Id}_{S\,\vec{t}}), \Pi_S[\vec{t}/\vec{x}]\big).$$

Note that $mc(\Pi_1, \mathrm{Id}_{S\,\vec{t}})$ also has exactly one reduct, namely, Π_1. Since $\Pi_1 \in \mathbf{RED}_{X^p\,\vec{t}}[\Omega] = \mathcal{R}\,\vec{t}$, **CR3** entails that $mc(\Pi_1, \mathrm{Id}_{S\,\vec{t}})$ is in $\mathcal{R}\,\vec{t}$. Since Ω is definitionally closed, we have that $\Xi_2 \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega]$. And since Ξ_2 is the only reduct of Ξ_1, this also means that $\Xi_1 \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega]$ by Definition 15. The reduct of Ξ, i.e. the derivation $mc(\Xi_1, \Pi_2, \ldots, \Pi_n, \Pi'[\Omega])$ is therefore in $\mathbf{RED}_C[\Omega]$ by the induction hypothesis.

II.5. Suppose Π ends with $\mathrm{w}\mathcal{L}$ or $\mathrm{c}\mathcal{L}$ acting on B_1, or *init*. Then $\Pi[\Omega]$ also ends with the same rule. The cut reduction rule that applies in this case is either $-/\mathrm{w}\mathcal{L}$, $-/\mathrm{c}\mathcal{L}$ or $-/init$. In these cases, parametric reducibility of the reducts follows straightforwardly from the assumption (in case of *init*), the induction hypothesis and Definition 15.

II.6. Suppose Π ends with *mc*. Then $\Pi[\Omega]$ also ends with *mc*. The reduction rule that applies in this case is the reduction $-/mc$. Parametric reducibility of the reduct in this case follows straightforwardly from the induction hypothesis and Definition 15.

II.7. Suppose Π ends with *subst* or a rule acting on a formula other than a cut formula. Most cases follow straightforwardly from the induction hypothesis, Lemmas 12 and 13, which is needed in the reduction case $-/\,\mathrm{eq}\,\mathcal{L}$ and $-/subst$. We show the interesting subcases here:

$\boxed{-/\mathrm{I}\mathcal{R}_p}$ Suppose Π ends with a non-trivial $\mathrm{I}\,\mathcal{R}_p$, i.e., Π is

$$\frac{\begin{array}{c}\Pi' \\ B_1, \ldots, B_n, \Gamma \longrightarrow D\,X^p\,\vec{t}\end{array}}{B_1, \ldots, B_n, \Gamma \longrightarrow X^p\,\vec{t}}\ \mathrm{I}\,\mathcal{R}_p$$

where $p\,\vec{x} \stackrel{\mu}{=} D\,p\,\vec{x}$ and $X^p \in supp(\Omega)$. Suppose $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$. Then $\Pi[\Omega]$ is the derivation $mc(\Pi'[\Omega], \Pi_S[\vec{t}/\vec{x}])$. The reduct of Ξ in this case is the derivation

$$\Xi' = mc\big(mc\big(\Pi_1, \ldots, \Pi_n, \Pi'[\Omega]\big), \Pi_S[\vec{t}/\vec{x}]\big).$$

By the induction hypothesis we have $mc(\Pi_1, \ldots, \Pi_n, \Pi'[\Omega]) \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega]$. This and the definitional closure of Ω imply that $\Xi' \in \mathcal{R}\,\vec{t} = \mathbf{RED}_{X^p\,\vec{t}}[\Omega]$.

$\boxed{-/\mathrm{I}\mathcal{R}}$ Suppose Π is

$$\frac{\begin{array}{c}\Pi' \\ B_1, \ldots, B_n, \Gamma \longrightarrow D\,X^p\,\vec{t}\end{array}}{B_1, \ldots, B_n, \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{I}\,\mathcal{R}$$

where $p\,\vec{y} \stackrel{\mu}{=} D\,p\,\vec{y}$. Without loss of generality we can assume that X^p is chosen to be sufficiently fresh (e.g., not occurring in Ω, Δ_1, B_1, etc.). Let $\Xi_1 = mc(\Pi_1, \ldots, \Pi_n, \Pi'[\Omega])$. Then the reduct of Ξ is the derivation Ξ'

$$\dfrac{\begin{array}{c}\Xi_1\\ \Delta_1, \ldots, \Delta_n, \Gamma[\Omega] \longrightarrow D\,X^p\,\vec{t}\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma[\Omega] \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}.$$

To show that $\Xi' \in \mathbf{RED}_C[\Omega]$, we first need to show that it is normalizable. This follows straightforwardly from the induction hypothesis (which shows that $\Xi_1 \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega]$) and Lemma 12. It then remains to show that

$$\Xi_2 = mc\big(\Xi_1\big[(\Pi_S, S)/X^p\big], \Pi_S[\vec{t}/\vec{x}]\big) \in \mathcal{R}\,\vec{t}$$

for every reducibility candidate $(\mathcal{R} : S)$ and every Π_S such that

$$\Pi_S[\vec{u}/\vec{x}] \in \mathbf{RED}_{D\,X^p\,\vec{u}}\big[\Omega, (\mathcal{R}, \Pi_S, S)/X^p\big] \Rightarrow \mathcal{R}\,\vec{u}, \quad \text{for every } \vec{u}. \tag{9}$$

So suppose $(\mathcal{R}, \Pi_S, S)$ satisfies (9) above. Let $\Omega' = [\Omega, (\mathcal{R}, \Pi_S, S)/X^p]$. By Lemma 17 Ω' is definitionally closed. Since we assume that X^p is fresh for B_i, we have $\mathbf{RED}_{B_i}[\Omega] = \mathbf{RED}_{B_i}[\Omega']$ by Lemma 14, and $\Pi_i[(\Pi_S, S)/X^p] = \Pi_i \in \mathbf{RED}_{B_i}[\Omega']$ by Lemma 6(1) for every $i \in \{1, \ldots, n\}$. Therefore, by the induction hypothesis we have:

$$\Xi_1\big[(\Pi_S, S)/X^p\big] = mc\big(\Pi_1, \ldots, \Pi_n, \Pi'\big[\Omega'\big]\big) \in \mathbf{RED}_{D\,X^p\,\vec{t}}\big[\Omega'\big].$$

This, together with the definitional closure of Ω', implies that $\Xi_2 \in \mathcal{R}\,\vec{t}$.

$\boxed{-/\,\mathrm{CI}\,\mathcal{L}_p}$ Suppose Π ends with a non-trivial CI $\mathcal{L}_p$, i.e., Π is

$$\dfrac{\begin{array}{c}\Pi'\\ B_1, \ldots, B_n, D\,X^p\,\vec{t}, \Gamma' \longrightarrow C\end{array}}{B_1, \ldots, B_n, X^p\,\vec{t}, \Gamma' \longrightarrow C}\ \mathrm{CI}\,\mathcal{L}_p$$

where $p\,\vec{x} \stackrel{\nu}{=} D\,p\,\vec{x}$ and $X^p \in supp(\Omega)$. Suppose $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$. Then

$$\Pi[\Omega] = mc\big(mc\big(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}]\big), \Pi'[\Omega]\big).$$

Let $\Xi_1 = mc(\mathrm{Id}_{S\,\vec{t}}, \Pi_S[\vec{t}/\vec{x}])$. By **CR4** $\mathrm{Id}_{S\,\vec{t}} \in \mathcal{R}\,\vec{t}$, and therefore, by the definitional closure of Ω, we have $\Xi_1 \in \mathbf{RED}_{D\,X^p\,\vec{t}}[\Omega]$. The reduct of Ξ in this case is

$$mc\big(\Xi_1, \Pi_1, \ldots, \Pi_n, \Pi'[\Omega]\big)$$

which is in $\mathbf{RED}_C[\Omega]$ by the induction hypothesis.

$\boxed{-/\,\mathrm{CI}\,\mathcal{R}}$ Suppose Π is

$$\dfrac{\begin{array}{c}\Pi'\\ B_1, \ldots, B_n, \Gamma \longrightarrow S\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S\\ S\,\vec{x} \longrightarrow D\,S\,\vec{x}\end{array}}{B_1, \ldots, B_n, \Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\,\mathcal{R}$$

where $p\,\vec{y} \stackrel{\nu}{=} D\,p\,\vec{y}$. Let $S' = S[\Omega]$. The derivation $\Pi[\Omega]$ in this case is

$$\dfrac{\begin{array}{c}\Pi'[\Omega]\\ B_1[\Omega], \ldots, B_n[\Omega], \Gamma[\Omega] \longrightarrow S'\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S[\Omega]\\ S'\,\vec{x} \longrightarrow D\,S'\,\vec{x}\end{array}}{B_1[\Omega], \ldots, B_n[\Omega], \Gamma[\Omega] \longrightarrow p\,\vec{t}}\ \mathrm{CI}\,\mathcal{R}.$$

Let Ξ_1 be the derivation $mc(\Pi_1, \ldots, \Pi_n, \Pi'[\Omega])$. By the induction hypothesis $\Xi_1 \in \mathbf{RED}_{S\,\vec{t}}[\Omega]$ and $\Pi_S[\Omega] \in \mathbf{RED}_{D\,S\,\vec{x}}[\Omega]$, hence both Ξ_1 and $\Pi_S[\Omega]$ are also normalizable by Lemma 12. The reduct of Ξ is the derivation Ξ'

$$\dfrac{\begin{array}{c}\Xi_1\\ \Delta_1, \ldots, \Delta_n, \Gamma[\Omega] \longrightarrow S'\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S[\Omega]\\ S'\,\vec{x} \longrightarrow D\,S'\,\vec{x}\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma[\Omega] \longrightarrow p\,\vec{t}}\ \mathrm{CI}\,\mathcal{R}.$$

Let X^p be a parameter fresh for Ω, Γ, Δ_i and B_i. To show that $\Xi' \in \mathbf{RED}_C[\Omega]$ we must first show that it is normalizable. This follows immediately from normalizability of Ξ_1 and $\Pi_S[\Omega]$. Then we need to find a reducibility candidate $(\mathcal{R} : S')$ such that

(a) $\Xi_1 \in \mathcal{R}$, and
(b) $\Pi_S[\Omega][\vec{u}/\vec{x}] \in \mathcal{R}\,\vec{u} \Rightarrow \mathbf{RED}_{D\,X^p\,\vec{u}}\,[\Omega, (\mathcal{R}, \Pi_S, S)/X^p]$, for every $\vec{u}$.

Let $\mathcal{R} = \{\Psi \mid \Psi \in \mathbf{RED}_{S\,\vec{u}}[\Omega]\}$. As in case **I.3.c**, we show, using Lemma 15, that $\mathcal{R}$ is a reducibility candidate of type S'. By the induction hypothesis we have $\Xi_1 \in \mathcal{R}$, so $\mathcal{R}$ satisfies (a). Using the same argument as in case **I.3.c** we can show that $\mathcal{R}$ also satisfies (b), i.e. by appealing to the induction hypothesis applied to Π_S. $\square$

Corollary 20. *Every derivation in* Linc_i^- *is reducible.*

Proof. The proof follows from Lemma 19, by setting $n = 0$ and Ω to the empty candidate substitution. $\square$

Since reducibility implies cut-elimination and since every cut-free derivation can be turned into a *subst*-free derivation (Lemma 2), it follows that every proof can be transformed into a cut-free and *subst*-free derivation.

Corollary 21. *Given a fixed definition, a sequent has a derivation in* Linc_i^- *if and only if it has a cut-free and subst-free derivation.*

6. Cut elimination for Linc$^-$

We now show how one can use the cut elimination result for Linc_i^- to prove cut elimination for Linc^-. But first we extend Linc^- with a version of the *subst* rule with signatures:

$$\frac{\{\Sigma[\theta];\, \Gamma[\theta] \longrightarrow C[\theta]\}_{\theta\in\mathbb{S}}}{\Sigma;\, \Gamma \longrightarrow C}\ \mathit{subst}.$$

We refer to this extension as Linc_s^-.

A derivation in Linc_s^- can be turned into a derivation in Linc_i^- by simply removing the signatures from all sequent occurrences in the derivation. The converse, i.e., turning a derivation in Linc_i^- into a derivation in Linc_s^-, is obviously not always possible due to the possibility of introducing new eigenvariables in the premises of $\exists\mathcal{R}$ and $\forall\mathcal{L}$ in Linc_i^-. We define below a class of derivations of Linc_i^- that can be turned into derivations in Linc_s^- by attaching signatures.

Recall that we identify derivations differing only in the choice of names for the internal variables. In the definition of $\mathfrak{D}(\Pi, \Sigma)$ below, we shall assume without loss of generality that the variables in the signature Σ are distinct from the internal variables in Π.

Definition 17. The *signature decoration* $\mathfrak{D}$ is a function that takes a Linc_i^- derivation Π of $\Gamma \longrightarrow C$ and a signature Σ and outputs either a derivation of $\Sigma; \Gamma \longrightarrow C$ in Linc_s^-, or $\star$ (denoting an ill-formed derivation). $\mathfrak{D}$ is defined by induction on Π as follows: If $\Gamma \longrightarrow C$ contains a free occurrence of a variable not in Σ, then $\mathfrak{D}(\Pi, \Sigma) = \star$. Otherwise:

- Suppose Π ends with $\mathrm{eq}\mathcal{L}$ or *subst*, with premise derivations $\{\Pi^\rho\}_{\rho\in\mathbb{I}}$ for some set of substitutions $\mathbb{I}$. If $\mathfrak{D}(\Pi^\rho, \Sigma[\rho]) \neq \star$ for every $\rho \in \mathbb{I}$, then $\mathfrak{D}(\Pi, \Sigma)$ ends with the same rule with premise derivations $\{\mathfrak{D}(\Pi^\rho, \Sigma[\rho])\}_{\rho\in\mathbb{I}}$. Otherwise, $\mathfrak{D}(\Pi, \Sigma) = \star$.
- Suppose Π ends with $\exists\mathcal{L}$ as shown below left. Then $\mathfrak{D}(\Pi, \Sigma)$ is as shown below right:

$$\dfrac{\begin{array}{c}\Pi' \\ B\,y, \Gamma' \longrightarrow C\end{array}}{\exists x.B\,x, \Gamma' \longrightarrow C}\ \exists\mathcal{L}, \qquad \dfrac{\begin{array}{c}\mathfrak{D}(\Pi', \Sigma \cup \{y\}) \\ \Sigma, y; B\,y, \Gamma' \longrightarrow C\end{array}}{\Sigma; \exists x.B\,x, \Gamma' \longrightarrow C}\ \exists\mathcal{L}$$

if $\mathfrak{D}(\Pi', \Sigma \cup \{y\}) \neq \star$; otherwise $\mathfrak{D}(\Pi, \Sigma) = \star$.
The case where Π ends with $\forall\mathcal{R}$ is defined similarly.
- Suppose Π ends with $\mathrm{I}\mathcal{L}$, as shown below left. If $\mathfrak{D}(\Pi_S, \{\vec{y}\}) \neq \star$ and $\mathfrak{D}(\Pi', \Sigma) \neq \star$, then $\mathfrak{D}(\Pi, \Sigma)$ is as shown below right; otherwise $\mathfrak{D}(\Pi, \Sigma) = \star$. The case where Π ends with $\mathrm{CI}\mathcal{R}$ is defined similarly:

$$\dfrac{\begin{array}{c}\Pi_S \\ B\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{array} \quad \begin{array}{c}\Pi' \\ \Gamma', S\,\vec{t} \longrightarrow C\end{array}}{\Gamma', p\,\vec{t} \longrightarrow C}\ \mathrm{I}\mathcal{L}, \qquad \dfrac{\begin{array}{c}\mathfrak{D}(\Pi_S, \{\vec{y}\}) \\ \vec{y}; B\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{array} \quad \begin{array}{c}\mathfrak{D}(\Pi', \Sigma) \\ \Gamma', S\,\vec{t} \longrightarrow C\end{array}}{\Sigma; \Gamma', p\,\vec{t} \longrightarrow C}\ \mathrm{I}\mathcal{L}.$$

- Suppose Π ends with a rule, other than $\mathrm{eq}\mathcal{L}$, *subst*, $\exists\mathcal{L}$, $\forall\mathcal{R}$, $\mathrm{I}\mathcal{L}$ and $\mathrm{CI}\mathcal{R}$, with premise derivations $\Pi_1, \ldots, \Pi_n$. If $\mathfrak{D}(\Pi_i, \Sigma) \neq \star$ for every $i \in \{1, \ldots, n\}$, then $\mathfrak{D}(\Pi, \Sigma)$ ends with the same rule and with premise derivations $\mathfrak{D}(\Pi_1, \Sigma), \ldots, \mathfrak{D}(\Pi_n, \Sigma)$. Otherwise, $\mathfrak{D}(\Pi, \Sigma) = \star$.

Obviously, if $\mathfrak{D}(\Pi, \Sigma) \neq \star$, it must be a derivation in Linc_s^-.

Lemma 22. *Let Π be a derivation in* Linc_i^-. *Suppose $\mathfrak{D}(\Pi, \Sigma) \neq \star$. Then for any substitution θ, $\mathfrak{D}(\Pi[\theta], \Sigma[\theta]) \neq \star$.*

Lemma 23. *Let Π be a derivation in* Linc_i^-. *Suppose $\mathfrak{D}(\Pi, \Sigma) \neq \star$. Let x_τ be an eigenvariable such that $x_\tau \notin \Sigma$. Then $\mathfrak{D}(\Pi, \Sigma \cup \{x_\tau\}) \neq \star$.*

Decorability of a derivation is in general *not* preserved by parameter substitutions, as the latter can introduce new subderivations that may not be decorable.

Definition 18. A parameter substitution Θ is said to *respect* $\mathfrak{D}$ if for every X^p $\Theta(X^p) = (\Pi_S, S)$ entails $\mathfrak{D}(\Pi_S, \{\vec{x}\}) \neq \star$, where $\{\vec{x}\}$ is the set of free variables occurring in the end sequent of Π_S.

Lemma 24. *Let Π be a derivation in* Linc_i^-. *Suppose $\mathfrak{D}(\Pi, \Sigma) \neq \star$. Let Θ be a parameter substitution that respects $\mathfrak{D}$. Then $\mathfrak{D}(\Pi[\Theta], \Sigma) \neq \star$.*

Proof. By induction on Π. We show here a case involving $\mathrm{I}\mathcal{R}_p$. Suppose Π ends with $\mathrm{I}\mathcal{R}_p$, as shown below left, and suppose $\Theta(X^p) = (\Pi_S, S)$, where $p\,\vec{x} \stackrel{\mu}{=} B\,S\,\vec{x}$ and Π_S is a derivation of $B\,S\,\vec{x} \longrightarrow S\,\vec{x}$. Then $\Pi[\Theta]$ is as shown below right:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow B\,X^p\vec{t}\end{array}}{\Gamma \longrightarrow X^p\vec{t}}\ \mathrm{I}\mathcal{R}_p, \qquad \dfrac{\begin{array}{c}\Pi'[\Theta] \\ \Gamma[\Theta] \longrightarrow B\,S\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_S[\vec{t}/\vec{x}] \\ B\,S\,\vec{t} \longrightarrow S\,\vec{t}\end{array}}{\Gamma[\Theta] \longrightarrow S\,\vec{t}}\ mc.$$

Since $\mathfrak{D}(\Pi, \Sigma) \neq \star$, by Definition 17 we have that $FV(\vec{t}) \subseteq \Sigma$, and also $\mathfrak{D}(\Pi', \Sigma) \neq \star$. By the induction hypothesis, we have

$$\mathfrak{D}\big(\Pi'[\Theta], \Sigma\big) \neq \star. \tag{10}$$

Since Θ respects $\mathfrak{D}$, it follows from Definition 18 that $\mathfrak{D}(\Pi_S, \{\vec{x}\}) \neq \star$. This, together with Lemma 22, implies that $\mathfrak{D}(\Pi_S[\vec{t}/\vec{x}], FV(\vec{t})) \neq \star$. Since $FV(\vec{t}) \subseteq \Sigma$, by Lemma 23 we have $\mathfrak{D}(\Pi_S[\vec{t}/\vec{x}], \Sigma) \neq \star$. This, together with (10), implies that

$$\mathfrak{D}\big(\Pi[\Theta], \Sigma\big) = mc\big(\mathfrak{D}\big(\Pi'[\Theta], \Sigma\big), \mathfrak{D}\big(\Pi_S[\vec{t}/\vec{x}], \Sigma\big)\big) \neq \star.$$

The case where Π ends with $\mathrm{CI}\mathcal{L}$ on a parameter X^p can be proved similarly. All the other cases follow straightforwardly from the induction hypothesis. $\square$

Lemma 25. *Let Ξ be a derivation in* Linc_i^- *ending with mc. If Ξ' is a reduct of Ξ and $\mathfrak{D}(\Xi, \Sigma) \neq \star$, then $\mathfrak{D}(\Xi', \Sigma) \neq \star$.*

Proof. This is easily proved by inspection of the reduction rules (see Appendix A) and by using Lemmas 22, 23 and 24. We show some representative cases here where those lemmas are used. Suppose that $\Xi = mc(\Pi_1, \ldots, \Pi_n, \Pi)$ and let Ξ' be its reduct. We look at the cases where the reduction is determined by the last rule in Π_1 and/or Π.

- $\boxed{\forall\mathcal{L}/\forall\mathcal{R}}$ Suppose Π_1 and Π are:

$$\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1 \longrightarrow B_1'\,y\end{array}}{\Delta_1, \longrightarrow \forall x.B_1'}\ \forall\mathcal{R}, \qquad \dfrac{\begin{array}{c}\Pi' \\ B_1'\,t, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{\forall x.B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \forall\mathcal{L}.$$

  Then Ξ reduces to $mc(\Pi_1'[t/y], \Pi_2, \ldots, \Pi_n, \Pi')$. Since $\mathfrak{D}(\Xi, \Sigma) \neq \star$, it follows that $\mathfrak{D}(\Pi_j, \Sigma) \neq \star$, for $j \in \{1, \ldots, n\}$, $\mathfrak{D}(\Pi', \Sigma) \neq \star$, and (by Lemma 23) $\mathfrak{D}(\Pi_1', \Sigma \cup \{y\}) \neq \star$. By Lemma 22 the latter implies that $\mathfrak{D}(\Pi_1'[t/y], \Sigma) \neq \star$. These together imply that

$$\begin{aligned}\mathfrak{D}(\Xi', \Sigma) &= \mathfrak{D}\big(mc\big(\Pi_1'[t/y], \Pi_2, \ldots, \Pi_n, \Pi'\big), \Sigma\big) \\ &= mc\big(\mathfrak{D}\big(\Pi_1'[t/y], \Sigma\big), \mathfrak{D}(\Pi_2, \Sigma), \ldots, \mathfrak{D}(\Pi_n, \Sigma), \mathfrak{D}(\Pi', \Sigma)\big) \neq \star.\end{aligned}$$

- $\boxed{\mathrm{I}\mathcal{R}/\mathrm{I}\mathcal{L}}$ Suppose Π_1 and Π are, respectively,

$$\dfrac{\begin{array}{c}\Pi_1' \\ \Delta_1 \longrightarrow D\,X^p\,\vec{t}\end{array}}{\Delta_1 \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R}, \qquad \dfrac{\begin{array}{c}\Pi_S \\ D\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{array} \quad \begin{array}{c}\Pi' \\ S\,\vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{p\,\vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{I}\mathcal{L}$$

  where $p\,\vec{x} \stackrel{\mu}{=} D\,p\,\vec{x}$ and X^p is a new parameter. Then Ξ reduces to

$$\Xi' = mc\big(mc\big(\Pi_1'\big[(\Pi_S, S)/X^p\big], \Pi_S[\vec{t}/\vec{y}]\big), \Pi_2, \ldots, \Pi_n, \Pi'\big).$$

  Since $\mathfrak{D}(\Xi, \Sigma) \neq \star$, we have:

1. $\mathfrak{D}(\Pi_i, \Sigma) \neq \star$, for $i \in \{2, \ldots, n\}$, and $\mathfrak{D}(\Pi', \Sigma) \neq \star$.
2. $\mathfrak{D}(\Pi_S, \{\vec{x}\}) \neq \star$, and therefore $[(\Pi_S, S)/X^p]$ respects $\mathfrak{D}$.
3. $FV(\vec{t}) \subseteq \Sigma$. Therefore, by item 2 and Lemma 22 (and possibly Lemma 23 if Σ is strictly larger than $FV(\vec{t})$), $\mathfrak{D}(\Pi_S[\vec{t}/\vec{x}], \Sigma) \neq \star$.
4. $\mathfrak{D}(\Pi_1', \Sigma) \neq \star$. This, together with item 2 above and Lemma 24, implies that $\mathfrak{D}(\Pi_1'[(\Pi_S, S)/X^p], \Sigma) \neq \star$.

From these, it follows that $\mathfrak{D}(\Xi', \Sigma) \neq \star$.

- $\boxed{\mathrm{eq}\mathcal{R}/\mathrm{eq}\mathcal{L}}$ Suppose Π_1 and Π are

$$\frac{}{\Delta_1 \longrightarrow s = t}\ \mathrm{eq}\mathcal{R}, \qquad \frac{\left\{ \begin{array}{c} \Pi^\rho \\ B_2[\rho], \ldots, B_n[\rho], \Gamma[\rho] \longrightarrow C[\rho] \end{array} \right\}_{\rho \in \mathbb{U}(s,t)}}{s = t, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{eq}\mathcal{L}.$$

In this case $\mathbb{U}(s,t) = \mathbb{S}$. Let $\Xi_1 = mc(\Pi_2, \ldots, \Pi_n, subst(\{\Pi^\rho\}_{\rho \in \mathbb{S}}))$. Then Ξ' is

$$\dfrac{\begin{array}{c} \Xi_1 \\ \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C \end{array}}{\Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}.$$

To show that $\mathfrak{D}(\Xi', \Sigma) \neq \star$, it is enough to show that $\mathfrak{D}(\Xi_1, \Sigma) \neq \star$. This follows from the fact that $\mathfrak{D}(\Pi_i, \Sigma) \neq \star$ and that $\mathfrak{D}(\Pi^\rho, \Sigma[\rho]) \neq \star$.

- $\boxed{\bullet\mathcal{L}/\circ\mathcal{L}}$ We show here the instance where $\bullet\mathcal{L}$ is $\exists\mathcal{L}$, i.e., Π_1 is as shown below left. Then Ξ' is the derivation below right:

$$\frac{\begin{array}{c} \Pi_1' \\ D\,y, \Delta_1 \longrightarrow B_1 \end{array}}{\exists x.D\,x, \Delta_1 \longrightarrow B_1}\ \exists\mathcal{L}, \qquad \frac{\begin{array}{c} \Xi_1 \\ D\,y, \Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C \end{array}}{\exists x.D\,x, \Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \exists\mathcal{L}$$

where $\Xi_1 = mc(\Pi_1', \Pi_2, \ldots, \Pi_n, \Pi)$. Let $\Sigma' = \Sigma \cup \{y\}$. By Lemma 23 and the assumption that $\mathfrak{D}(\Xi, \Sigma) \neq \star$, we have that $\mathfrak{D}(\Pi_1', \Sigma') \neq \star$, $\mathfrak{D}(\Pi_i, \Sigma') \neq \star$, for $i \in \{2, \ldots, n\}$, and $\mathfrak{D}(\Pi, \Sigma') \neq \star$. It follows that $\mathfrak{D}(\Xi_1, \Sigma') \neq \star$ and therefore, $\mathfrak{D}(\Xi', \Sigma) \neq \star$. □

Theorem 26. *A sequent has a derivation in* Linc$^-$ *if and only if it has a cut-free derivation.*

Proof. Let Π be a derivation of $\Sigma; \Gamma \longrightarrow C$ in Linc$^-$. Let Π' be a Linc$_i^-$ derivation obtained from Π by removing all signatures. Obviously, we have $\mathfrak{D}(\Pi', \Sigma) = \Pi$. By Corollary 20 Π' can be transformed into a cut-free derivation Ξ. By Lemma 25 we know that cut reduction preserves decorability of derivations, so we have that $\mathfrak{D}(\Xi, \Sigma) \neq \star$. Moreover, as decorations do not introduce extra rules, the derivation $\mathfrak{D}(\Xi, \Sigma)$ is a cut-free derivation in Linc$_s^-$. We then use a transformation (analogous to the one in the proof of Lemma 2) to remove the *subst* instances in $\mathfrak{D}(\Xi, \Sigma)$ and get a cut-free and *subst*-free derivation in Linc$^-$. □

The consistency of Linc$^-$ is an immediate consequence of cut-elimination. By consistency we mean the following: given a fixed definition and an arbitrary formula C, it is not the case that both C and $C \supset \bot$ are provable.

Corollary 27. *The logic* Linc$^-$ *is consistent.*

7. Related work and conclusions

There is a long association between inductive definition and mathematical logics [1], in particular with proof-theory, starting with the Takeuti's conjecture, the earliest relevant entry for our purposes being Martin-Löf's theory of *iterated inductive definitions* [21]. From the representation of algebraic types and the introduction of (co-)inductive types in System F [25], (co-)induction/recursion became mainstream in the theorem proving community and made it into type-theoretic proof assistants such as Coq, eventually in the let-rec style of functional programming languages, as in Giménez's *Calculus of Infinite Constructions* [16].[6] Unlike these type-theoretic settings, we put less emphasis on proof terms and strong normalization; in fact, our cut elimination procedure is actually a form of weak normalization, in the sense that it only guarantees termination

6 In higher order logic (co-)inductive definitions are usually obtained via the Tarski set-theoretic fixed point construction, as realized for example in Isabelle/HOL [32]. As we mentioned before, those approaches are at odd with HOAS even at the level of the syntax. This issue has originated a research field in its own and we refer to [12] for an extensive comparison of approaches and systems.

with respect to a particular strategy, i.e., by reducing the lowest cuts in a derivation tree. Our notion of equality, which internalizes unification in the left rule, departs from the more traditional view. As a consequence of these differences, it is not obvious that strong normalization proofs for term calculi with (co-)inductive types can be adapted straightforwardly to our setting.

Baelde and Miller have recently introduced μMALL, an extension of multiplicative additive linear logic with least and greatest fixed points [5]. There is an ongoing discussion whether definitions or fixed points offer a better proof-theoretic understanding of (co-)induction in a logical framework. Suffice to say here that fixed points allow one to arbitrarily interleave and nest occurrences of μ and ν, although all fixed point bodies are required to be *monotonic*. In the cited work, cut elimination is indirectly argued via the standard encoding of the least and the greatest fixed point operators into second-order linear logic (LL2): once formulae and proofs are translated into LL2, they are then normalized, focused and translated back into cut-free μMALL derivations. This approach is not completely satisfactory, since it relies on some missing components, namely cut-elimination and completeness of focusing for LL2 extended with first-order quantification and Clark's equality theory, the former being the harder – we are not aware of any such proof as far as standard sequent calculi go [31]. Further, indirect proofs tend not to be easily generalizable, as we argued before. In fact, Baelde has more recently given a direct Girard-style cut-elimination proof for μMALL [3,4]. The proof uses a notion of orthogonality in the definition of reducibility, defined via involutive negation; hence it does not look like it can be adapted straightforwardly to an intuitionistic setting like ours.

Baelde has also identified in [4] an intuitionistic logic with least and greatest fixed points by analyzing its encoding into a focused system for μMALL. The obtained fragment is strictly weaker than Linc$^-$, and it basically coincides with the logic implemented in the *Bedwyr* [6] model checker. In his dissertation [3] Baelde also introduces the unrestricted intuitionistic fixed point logic μLJ. He gives cut-reductions rules, but no complete proof. μMALL and μLJ plus ∇ are under implementation in the system *Tac* [7], a proof assistant with an emphasis on automating the proof of (simple) (co-)inductive theorems via focusing.

Circular proofs are also connected with the proof-theory of fixed point logics and process calculi [35,40], as well as in traditional sequent calculi such as in [8]. This approach seemed particularly promising from the viewpoint of proof search, both inductively and co-inductively. The issue is the equivalence between systems with *local* vs. *global* induction, that is, between fixed point rules vs. well-founded and guarded induction. In the traditional sequent calculus, it is unknown whether every global inductive proof can be translated into a local one. Cut-elimination proofs have not been explored, as far as we know.

We have presented a proof-theoretic treatment of both induction and co-induction in a sequent calculus compatible with HOAS encodings. The proof principle underlying the explicit proof rules is basically fixed point (co-)induction. However, the formulation of the rules is inspired by a second-order encoding of least and greatest fixed points. We have developed a new cut elimination proof, radically different from previous proofs [22,41], using a reducibility-candidate technique à la Girard. Consistency of the logic is an easy consequence of cut-elimination. We conjecture that our cut elimination proof can be extended to prove cut elimination for extensions of Linc$^-$ with the ∇-quantifier. It seems that this is easier to do with a weaker form of ∇ as seen in the logic LG^{ω} [42] and the logic $\mathcal{G}$ [15], rather than the version of ∇ in [41], as there is no issue in maintaining another layer of variable contexts like in $FO\lambda^{\Delta\nabla}$ [28]. In particular, the use of "nominal abstraction" in the equality rules in $\mathcal{G}$ allows one to abstract from the underlying notions of substitutions in those rules: hence we conjecture that the cut reduction rules for Linc$^-$ can be carried over to $\mathcal{G}$ with some minor adjustments, such as closing the definitions of reducibility candidates under name permutation.

Acknowledgements

The Linc$^-$ logic was developed in collaboration with Dale Miller. We thank David Baelde for his comments to a draft of this paper and the reviewers for many useful suggestions. The first author is supported by the Australian Research Council Discovery Project DP110103173.

Appendix A. The complete set of cut reduction rules

Essential cases:

$\boxed{\wedge\mathcal{R}/\wedge\mathcal{L}}$ If Π_1 and Π are

$$\frac{\begin{array}{cc}\Pi_1' & \Pi_1'' \\ \Delta_1 \longrightarrow B_1' & \Delta_1 \longrightarrow B_1''\end{array}}{\Delta_1 \longrightarrow B_1' \wedge B_1''}\ \wedge\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi' \\ B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{B_1' \wedge B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \wedge\mathcal{L}$$

then Ξ reduces to $mc(\Pi_1', \Pi_2, \ldots, \Pi_n, \Pi')$. The case for the other $\wedge\mathcal{L}$ rule is symmetric.

$\boxed{\vee\mathcal{R}/\vee\mathcal{L}}$ Suppose Π_1 and Π are

$$\frac{\begin{array}{c}\Pi_1'\\ \Delta_1 \longrightarrow B_1'\end{array}}{\Delta_1 \longrightarrow B_1' \vee B_1''}\ \vee\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi'\\ B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array} \quad \begin{array}{c}\Pi''\\ B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{B_1' \vee B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \vee\mathcal{L}.$$

Then Ξ reduces to $mc(\Pi_1', \Pi_2, \ldots, \Pi')$. The case for the other $\vee\mathcal{R}$ rule is symmetric.

$\boxed{\supset\mathcal{R}/\supset\mathcal{L}}$ Suppose Π_1 and Π are

$$\frac{\begin{array}{c}\Pi_1'\\ B_1', \Delta_1 \longrightarrow B_1''\end{array}}{\Delta_1 \longrightarrow B_1' \supset B_1''}\ \supset\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi'\\ B_2, \ldots, B_n, \Gamma \longrightarrow B_1'\end{array} \quad \begin{array}{c}\Pi''\\ B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{B_1' \supset B_1'', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \supset\mathcal{L}.$$

Let $\Xi_1 = mc(mc(\Pi_2, \ldots, \Pi_n, \Pi'), \Pi_1')$. Then Ξ reduces to $\Delta_1, \ldots, \Delta_n, \Gamma, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C$

$$\cfrac{\cfrac{\begin{array}{c}\Xi_1\\ \ldots \longrightarrow B_1''\end{array} \quad \left\{\begin{array}{c}\Pi_i\\ \Delta_i \longrightarrow B_i\end{array}\right\}_{i\in\{2..n\}} \quad \begin{array}{c}\Pi''\\ B_1'', \{B_i\}_{i\in\{2..n\}}, \Gamma \longrightarrow C\end{array}}{\Delta_1, \ldots, \Delta_n, \Gamma, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ mc.}{\Delta_1, \ldots, \Delta_n, \Gamma \longrightarrow C}\ c\mathcal{L}$$

$\boxed{\forall\mathcal{R}/\forall\mathcal{L}}$ If Π_1 and Π are

$$\frac{\begin{array}{c}\Pi_1'\\ \Delta_1 \longrightarrow B_1'\, y\end{array}}{\Delta_1 \longrightarrow \forall x.B_1'}\ \forall\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi'\\ B_1'\, t, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{\forall x.B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \forall\mathcal{L}$$

then Ξ reduces to $mc(\Pi_1'[t/y], \Pi_2, \ldots, \Pi_n, \Pi')$.

$\boxed{\exists\mathcal{R}/\exists\mathcal{L}}$ If Π_1 and Π are

$$\frac{\begin{array}{c}\Pi_1'\\ \Delta_1 \longrightarrow B_1'\, t\end{array}}{\Delta_1 \longrightarrow \exists x.B_1'}\ \exists\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi'\\ B_1'\, y, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{\exists x.B_1', B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \exists\mathcal{L}$$

then Ξ reduces to $mc(\Pi_1', \Pi_2, \ldots, \Pi_n, \Pi'[t/y])$.

$\boxed{\mathrm{I}\mathcal{R}/\mathrm{I}\mathcal{L}}$ Suppose Π_1 and Π are, respectively,

$$\frac{\begin{array}{c}\Pi_1'\\ \Delta_1 \longrightarrow D\, X^p\, \vec{t}\end{array}}{\Delta_1 \longrightarrow p\, \vec{t}}\ \mathrm{I}\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi_S\\ D\, S\, \vec{y} \longrightarrow S\, \vec{y}\end{array} \quad \begin{array}{c}\Pi'\\ S\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{p\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{I}\mathcal{L}$$

where $p\, \vec{x} \stackrel{\mu}{=} D\, p\, \vec{x}$ and X^p is a new parameter. Then Ξ reduces to

$$mc\big(mc\big(\Pi_1'\big[(\Pi_S, S)/X^p\big], \Pi_S[\vec{t}/\vec{y}]\big), \Pi_2, \ldots, \Pi_n, \Pi'\big).$$

$\boxed{\mathrm{CI}\mathcal{R}/\mathrm{CI}\mathcal{L}}$ Suppose Π_1 and Π are

$$\frac{\begin{array}{c}\Pi_1'\\ \Delta_1 \longrightarrow S\, \vec{t}\end{array} \quad \begin{array}{c}\Pi_S\\ S\, \vec{y} \longrightarrow D\, S\, \vec{y}\end{array}}{\Delta_1 \longrightarrow p\, \vec{t}}\ \mathrm{CI}\mathcal{R}, \qquad \frac{\begin{array}{c}\Pi'\\ D\, X^p\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{array}}{p\, \vec{t}, B_2, \ldots, B_n, \Gamma \longrightarrow C}\ \mathrm{CI}\mathcal{L}$$

where $p\, \vec{y} \stackrel{\nu}{=} D\, p\, \vec{y}$ and X^p is a new parameter. Then Ξ reduces to

$$mc\big(mc\big(\Pi_1', \Pi_S[\vec{t}/\vec{y}]\big), \Pi_2, \ldots, \Pi_n, \Pi'\big[(\Pi_S, S)/X^p\big]\big).$$

$\boxed{\mathrm{eq}\,\mathcal{R}/\mathrm{eq}\,\mathcal{L}}$ Suppose Π_1 is

$$\frac{}{\Delta_1 \longrightarrow s = t}\ \mathrm{eq}\,\mathcal{R}$$

and $\Pi = \mathrm{eq}\,\mathcal{L}(\{\Pi^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where the $\mathrm{eq}\,\mathcal{L}$ rule is applied to the cut formula $s = t$. Note that in this case $\mathbb{U}(s,t) = \mathbb{S}$. Let

$$\Xi_1 = mc\big(\Pi_2, \ldots, \Pi_n, subst\big(\{\Pi^\rho\}_{\rho\in\mathbb{S}}\big)\big).$$

Then Ξ reduces to

$$\frac{\stackrel{\Xi_1}{\Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}}{\Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}.$$

Left-commutative cases: In the following cases, we suppose that Π ends with a left rule, other than $\{\mathrm{c}\mathcal{L}, \mathrm{w}\mathcal{L}\}$, acting on B_1.

$\boxed{\bullet\mathcal{L}/\circ\mathcal{L}}$ Suppose Π_1 is as below left, where $\mathbb{I}$ is an index set and $\bullet\mathcal{L}$ is any left rule except $\supset\mathcal{L}$, $\mathrm{eq}\,\mathcal{L}$, or $\mathrm{I}\mathcal{L}$. Let $\Xi^i = mc(\Pi_1^i, \Pi_2, \ldots, \Pi_n, \Pi)$. Then Ξ reduces to the derivation given below right:

$$\frac{\left\{\stackrel{\Pi_1^i}{\Delta_1^i \longrightarrow B_1}\right\}_{i\in\mathbb{I}}}{\Delta_1 \longrightarrow B_1}\ \bullet\mathcal{L}, \qquad \frac{\left\{\stackrel{\Xi^i}{\Delta_1^i, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\right\}_{i\in\mathbb{I}}}{\Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \bullet\mathcal{L}.$$

$\boxed{\supset\mathcal{L}/\circ\mathcal{L}}$ Suppose Π_1 is

$$\frac{\stackrel{\Pi_1'}{\Delta_1' \longrightarrow D_1'} \quad \stackrel{\Pi_1''}{D_1'', \Delta_1' \longrightarrow B_1}}{D_1' \supset D_1'', \Delta_1' \longrightarrow B_1}\ \supset\mathcal{L}.$$

Let $\Xi_1 = mc(\Pi_1'', \Pi_2, \ldots, \Pi_n, \Pi)$. Then Ξ reduces to

$$\frac{\dfrac{\stackrel{\Pi_1'}{\Delta_1' \longrightarrow D_1'}}{\Delta_1', \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow D_1'}\ \mathrm{w}\mathcal{L} \quad \stackrel{\Xi_1}{D_1'', \Delta_1', \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}}{D_1' \supset D_1'', \Delta_1', \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \supset\mathcal{L}.$$

$\boxed{\mathrm{I}\mathcal{L}/\circ\mathcal{L}}$ Suppose Π_1 is

$$\frac{\stackrel{\Pi_S}{D\,S\,\vec{y} \longrightarrow S\,\vec{y}} \quad \stackrel{\Pi_1'}{S\,\vec{t}, \Delta_1' \longrightarrow B_1}}{p\,\vec{t}, \Delta_1' \longrightarrow B_1}\ \mathrm{I}\mathcal{L}$$

where $p\,\vec{y} \stackrel{\mu}{=} D\,p\,\vec{y}$. Let $\Xi_1 = mc(\Pi_1', \Pi_2, \ldots, \Pi_n, \Pi)$. Then Ξ reduces to

$$\frac{\stackrel{\Pi_S}{D\,S\,\vec{y} \longrightarrow S\,\vec{y}} \quad \stackrel{\Xi_1}{S\,\vec{t}, \Delta_1', \ldots, \Delta_n, \Gamma \longrightarrow C}}{p\,\vec{t}, \Delta_1', \ldots, \Delta_n \longrightarrow C}\ \mathrm{I}\mathcal{L}.$$

$\boxed{\mathrm{eq}\,\mathcal{L}/\circ\mathcal{L}}$ Suppose $\Pi_1 = \mathrm{eq}\,\mathcal{L}(\{\Pi_1^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where the $\mathrm{eq}\,\mathcal{L}$ is applied to an equation $s = t$ in Δ_1. Let $\Xi^\rho = mc(\Pi_1^\rho, \Pi_2[\rho], \ldots, \Pi_n[\rho], \Pi[\rho])$. Then Ξ reduces to $\mathrm{eq}\,\mathcal{L}(\{\Xi^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where the $\mathrm{eq}\,\mathcal{L}$ rule is applied to the same $s = t$ in Δ_1.

$\boxed{subst/\circ\mathcal{L}}$ Suppose Π_1 is $subst(\{\Pi_1^\rho\}_{\rho\in\mathbb{S}})$. Then Ξ reduces to

$$subst\big(\big\{mc\big(\Pi_1^\rho, \Pi_2[\rho], \ldots, \Pi_n[\rho], \Pi[\rho]\big)\big\}_{\rho\in\mathbb{S}}\big).$$

Right-commutative cases:

$\boxed{-/\circ\mathcal{L}}$ Suppose Π is as given below left, where $\mathbb{I}$ is an index set and $\circ\mathcal{L}$ is any left rule other than $\supset\mathcal{L}$, eq$\mathcal{L}$, or I$\mathcal{L}$ acting on a formula other than $B_1,\ldots,B_n$. Let $\Xi^i = mc(\Pi_1,\ldots,\Pi_n,\Pi^i)$. Then Ξ reduces to the derivation given below right:

$$\frac{\left\{\begin{matrix}\Pi^i\\ B_1,\ldots,B_n,\Gamma^i \longrightarrow C\end{matrix}\right\}_{i\in\mathbb{I}}}{B_1,\ldots,B_n,\Gamma \longrightarrow C}\ \circ\mathcal{L}, \qquad \frac{\left\{\begin{matrix}\Xi^i\\ \Delta_1,\ldots,\Delta_n,\Gamma^i \longrightarrow C\end{matrix}\right\}_{i\in\mathbb{I}}}{\Delta_1,\ldots,\Delta_n,\Gamma \longrightarrow C}\ \circ\mathcal{L}.$$

$\boxed{-/\supset\mathcal{L}}$ Suppose Π is

$$\frac{\begin{matrix}\Pi'\\ B_1,\ldots,B_n,\Gamma' \longrightarrow D'\end{matrix} \quad \begin{matrix}\Pi''\\ B_1,\ldots,B_n,D'',\Gamma' \longrightarrow C\end{matrix}}{B_1,\ldots,B_n,D'\supset D'',\Gamma' \longrightarrow C}\ \supset\mathcal{L}.$$

Let $\Xi_1 = mc(\Pi_1,\ldots,\Pi_n,\Pi')$ and let $\Xi_2 = mc(\Pi_1,\ldots,\Pi_n,\Pi'')$. Then Ξ reduces to

$$\frac{\begin{matrix}\Xi_1\\ \Delta_1,\ldots,\Delta_n,\Gamma' \longrightarrow D'\end{matrix} \quad \begin{matrix}\Xi_2\\ \Delta_1,\ldots,\Delta_n,D'',\Gamma' \longrightarrow C\end{matrix}}{\Delta_1,\ldots,\Delta_n,D'\supset D'',\Gamma' \longrightarrow C}\ \supset\mathcal{L}.$$

$\boxed{-/\mathrm{I}\mathcal{L}}$ Suppose Π is

$$\frac{\begin{matrix}\Pi_S\\ D\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{matrix} \quad \begin{matrix}\Pi'\\ B_1,\ldots,B_n,S\,\vec{t},\Gamma' \longrightarrow C\end{matrix}}{B_1,\ldots,B_n,p\,\vec{t},\Gamma' \longrightarrow C}\ \mathrm{I}\mathcal{L}$$

where $p\,\vec{y} \stackrel{\mu}{=} D\,p\,\vec{y}$. Let $\Xi_1 = mc(\Pi_1,\ldots,\Pi_n,\Pi')$. Then Ξ reduces to

$$\frac{\begin{matrix}\Pi_S\\ D\,S\,\vec{y} \longrightarrow S\,\vec{y}\end{matrix} \quad \begin{matrix}\Xi_1\\ \Delta_1,\ldots,\Delta_n,S\,\vec{t},\Gamma' \longrightarrow C\end{matrix}}{\Delta_1,\ldots,\Delta_n,p\,\vec{t},\Gamma' \longrightarrow C}\ \mathrm{I}\mathcal{L}.$$

$\boxed{-/\,\mathrm{eq}\mathcal{L}}$ Suppose $\Pi = \mathrm{eq}\mathcal{L}(\{\Pi^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where the eq$\mathcal{L}$ rule is applied to an equation $s = t$ in Γ. Let $\Xi^\rho = mc(\Pi_1[\rho],\ldots,\Pi_n[\rho],\Pi^\rho)$. Then Ξ reduces to $\mathrm{eq}\mathcal{L}(\{\Xi^\rho\}_{\rho\in\mathbb{U}(s,t)})$ where the eq$\mathcal{L}$ is applied to the same $s = t$ in Γ.

$\boxed{-/subst}$ If $\Pi = subst(\{\Pi^\rho\}_{\rho\in\mathbb{S}})$, then Ξ reduces to

$$subst\big(\big\{mc\big(\Pi_1[\rho],\ldots,\Pi_n[\rho],\Pi^\rho\big)\big\}_{\rho\in\mathbb{S}}\big).$$

$\boxed{-/\circ\mathcal{R}}$ If Π is as below left, where $\mathbb{I}$ is an index set and $\circ\mathcal{R}$ is any right rule except CI$\mathcal{R}$, then Ξ reduces to the derivation below right, where $\Xi^i = mc(\Pi_1,\ldots,\Pi_n,\Pi^i)$:

$$\frac{\left\{\begin{matrix}\Pi^i\\ B_1,\ldots,B_n,\Gamma^i \longrightarrow C^i\end{matrix}\right\}_{i\in\mathbb{I}}}{B_1,\ldots,B_n,\Gamma \longrightarrow C}\ \circ\mathcal{R}, \qquad \frac{\left\{\begin{matrix}\Xi^i\\ \Delta_1,\ldots,\Delta_n,\Gamma^i \longrightarrow C^i\end{matrix}\right\}_{i\in\mathbb{I}}}{\Delta_1,\ldots,\Delta_n,\Gamma \longrightarrow C}\ \circ\mathcal{R}.$$

$\boxed{-/\,\mathrm{CI}\mathcal{R}}$ Suppose Π is

$$\frac{\begin{matrix}\Pi'\\ B_1,\ldots,B_n,\Gamma \longrightarrow S\,\vec{t}\end{matrix} \quad \begin{matrix}\Pi_S\\ S\,\vec{y} \longrightarrow D\,S\,\vec{y}\end{matrix}}{B_1,\ldots,B_n,\Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}$$

where $p\,\vec{y} \stackrel{\nu}{=} D\,p\,\vec{y}$. Let $\Xi_1 = mc(\Pi_1,\ldots,\Pi_n,\Pi')$. Then Ξ reduces to

$$\frac{\begin{matrix}\Xi_1\\ \Delta_1,\ldots,\Delta_n,\Gamma \longrightarrow S\,\vec{t}\end{matrix} \quad \begin{matrix}\Pi_S\\ S\,\vec{y} \longrightarrow D\,S\,\vec{y}\end{matrix}}{\Delta_1,\ldots,\Delta_n,\Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R}.$$

Multicut cases:

$\boxed{mc/\circ\mathcal{L}}$ If Π ends with a left rule other than c$\mathcal{L}$ and w$\mathcal{L}$, acting on B_1 and Π_1 ends with a multicut and reduces to Π_1', then Ξ reduces to $mc(\Pi_1', \Pi_2, \dots, \Pi_n, \Pi)$.

$\boxed{-/mc}$ Suppose Π is

$$\dfrac{\left\{\begin{matrix}\Pi^j\\ \{B_i\}_{i\in\mathbb{I}^j}, \Gamma^j \longrightarrow D^j\end{matrix}\right\}_{j\in\{1..m\}} \quad \begin{matrix}\Pi'\\ \{D^j\}_{j\in\{1..m\}}, \{B_i\}_{i\in\mathbb{I}'}, \Gamma' \longrightarrow C\end{matrix}}{B_1, \dots, B_n, \Gamma^1, \dots, \Gamma^m, \Gamma' \longrightarrow C}\ mc,$$

where $\mathbb{I}^1, \dots, \mathbb{I}^m, \mathbb{I}'$ partition the set $\{1, \dots, n\}$. For $1 \leqslant j \leqslant m$ let Ξ^j be

$$\dfrac{\left\{\begin{matrix}\Pi_i\\ \Delta_i \longrightarrow B_i\end{matrix}\right\}_{i\in\mathbb{I}^j} \quad \begin{matrix}\Pi^j\\ \{B_i\}_{i\in\mathbb{I}^j}, \Gamma^j \longrightarrow D^j\end{matrix}}{\{\Delta_i\}_{i\in\mathbb{I}^j}, \Gamma^j \longrightarrow D^j}\ mc.$$

Then Ξ reduces to

$$\dfrac{\left\{\begin{matrix}\Xi^j\\ \dots \longrightarrow D^j\end{matrix}\right\}_{j\in\{1..m\}} \quad \left\{\begin{matrix}\Pi_i\\ \Delta_i \longrightarrow B_i\end{matrix}\right\}_{i\in\mathbb{I}'} \quad \begin{matrix}\Pi'\\ \dots \longrightarrow C\end{matrix}}{\Delta_1, \dots, \Delta_n, \Gamma^1, \dots, \Gamma^m, \Gamma' \longrightarrow C}\ mc.$$

Structural cases:

$\boxed{-/\mathrm{c}\mathcal{L}}$ If Π is as shown below left, then Ξ reduces to the derivation shown below right, where $\Xi_1 = mc(\Pi_1, \Pi_1, \Pi_2, \dots, \Pi_n, \Pi')$:

$$\dfrac{\begin{matrix}\Pi'\\ B_1, B_1, B_2, \dots, B_n, \Gamma \longrightarrow C\end{matrix}}{B_1, B_2, \dots, B_n, \Gamma \longrightarrow C}\ \mathrm{c}\mathcal{L}, \qquad \dfrac{\begin{matrix}\Xi_1\\ \Delta_1, \Delta_1, \Delta_2, \dots, \Delta_n, \Delta_n, \Gamma \longrightarrow C\end{matrix}}{\Delta_1, \Delta_2, \dots, \Delta_n, \Gamma \longrightarrow C}\ \mathrm{c}\mathcal{L}.$$

$\boxed{-/\mathrm{w}\mathcal{L}}$ If Π is as shown below left, then Ξ reduces to the derivation shown below right, where $\Xi_1 = mc(\Pi_2, \dots, \Pi_n, \Pi')$.

$$\dfrac{\begin{matrix}\Pi'\\ B_2, \dots, B_n, \Gamma \longrightarrow C\end{matrix}}{B_1, B_2, \dots, B_n, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}, \qquad \dfrac{\begin{matrix}\Xi_1\\ \Delta_2, \dots, \Delta_n, \Gamma \longrightarrow C\end{matrix}}{\Delta_1, \Delta_2, \dots, \Delta_n, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}.$$

Axiom cases:

$\boxed{init/\circ\mathcal{L}}$ Suppose Π ends with a left-rule acting on B_1 and Π_1 ends with the *init* rule. Then it must be the case that $\Delta_1 = \{B_1\}$ and Ξ reduces to $mc(\Pi_2, \dots, \Pi_n, \Pi)$.

$\boxed{-/init}$ If Π ends with the *init* rule, then $n = 1$, Γ is the empty multiset, and C must be a cut formula, i.e., $C = B_1$. Therefore Ξ reduces to Π_1.

Appendix B. Proofs for Sections 5.1 and 5.3

Lemma 7. *Let Π be a derivation ending with a mc and let θ be a substitution. If $\Pi[\theta]$ reduces to Ξ, then there exists a derivation Π' such that $\Xi = \Pi'[\theta]$ and Π reduces to Π'.*

Proof. Observe that the redexes of a derivation are not affected by eigenvariable substitution, since the cut reduction rules are determined by the last rule of the premise derivations, which are not changed by substitution. Therefore, any cut reduction rule that is applied to $\Pi[\theta]$ to get Ξ can also be applied to Π. Suppose that Π' is the reduct of Π obtained this way. In all cases, except for the cases where the reduction rule applied is either $\mathrm{I}\mathcal{R}/\mathrm{I}\mathcal{L}$, $\mathrm{CI}\mathcal{L}/\mathrm{CI}\mathcal{R}$, or those involving $\mathrm{eq}\mathcal{L}$, it is a matter of routine to check that $\Pi'[\theta] = \Xi$. For the reduction rules $\mathrm{I}\mathcal{R}/\mathrm{I}\mathcal{L}$ and $\mathrm{CI}\mathcal{L}/\mathrm{CI}\mathcal{R}$, we need Lemma 5, which shows that eigenvariable substitution commutes with parameter substitution. We show here the case involving $\mathrm{eq}\mathcal{L}$. The only interesting case is the reduction $\mathrm{eq}\mathcal{L}/\mathrm{eq}\mathcal{R}$. So suppose Π is the derivation:

$$\dfrac{\dfrac{}{\Delta_1 \longrightarrow t = t}\ \mathrm{eq}\mathcal{R} \quad \left\{\begin{matrix}\Pi_i \\ \Delta_i \longrightarrow B_i\end{matrix}\right\}_{i \in \{2,\ldots,n\}} \quad \begin{matrix}\Pi' \\ t = t, B_2, \ldots, B_n, \Gamma \longrightarrow C\end{matrix}}{\Delta_1, \Delta_2, \Gamma \longrightarrow C}\ mc$$

where $\Pi' = \mathrm{eq}\mathcal{L}(\{\Pi^\rho\}_{\rho \in \mathbb{S}})$ is a derivation ending with $\mathrm{eq}\mathcal{L}$ acting on the cut formula $t = t$. Let Π_1 be the leftmost among the premises of the mc rule above. According to Definition 5 the derivation $\Pi[\theta]$ is

$$mc\big(\Pi_1[\theta], \Pi_2[\theta], \ldots, \Pi_n[\theta], \mathrm{eq}\mathcal{L}\big(\big\{\Pi^{\theta \circ \rho'}\big\}_{\rho' \in \mathbb{S}}\big)\big).$$

Let $\Psi = mc(\Pi_2[\theta], \ldots, \Pi_n[\theta], subst(\{\Pi^{(\theta \circ \rho)}\}_{\rho \in \mathbb{S}}))$. The reduct of $\Pi[\theta]$ in this case (modulo the different order in which the weakening steps are applied) is:

$$\dfrac{\begin{matrix}\Psi \\ \Delta_2[\theta], \ldots, \Delta_n[\theta], \Gamma[\theta] \longrightarrow C[\theta]\end{matrix}}{\Delta_1[\theta], \Delta_2[\theta], \ldots, \Delta_n[\theta], \Gamma[\theta] \longrightarrow C[\theta]}\ \mathrm{w}\mathcal{L}.$$

Let us call this derivation Ξ. Let $\Psi' = mc(\Pi_2, \ldots, \Pi_n, subst(\{\Pi^\rho\}_{\rho \in \mathbb{S}}))$. The above reduct can be matched by the following reduct of Π (using the same order of applications of $\mathrm{w}\mathcal{L}$):

$$\dfrac{\begin{matrix}\Psi' \\ \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C\end{matrix}}{\Delta_1, \Delta_2, \ldots, \Delta_n, \Gamma \longrightarrow C}\ \mathrm{w}\mathcal{L}.$$

Let us call this derivation Π'. By Definition 5 we have $\Psi' = \Psi[\theta]$ and obviously also $\Xi = \Pi'[\theta]$. □

Lemma 12. *If $\Pi \in \mathbf{RED}_C[\Omega]$, then Π is normalizable.*

Proof. If $C = X^p\,\vec{u}$ for some $\vec{u}$ and $X^p \in supp(\Omega)$, then $\Pi \in \mathcal{R}$, where $\Omega(X^p) = (\mathcal{R}, \Pi_S, S)$, hence it is normalizable by Definition 13 (specifically, condition **CR1**). Otherwise, Π is normalizable by Definition 15. □

Lemma 13. *If $\Pi \in \mathbf{RED}_C[\Omega]$, then, for every substitution ρ, $\Pi[\rho] \in \mathbf{RED}_{C[\rho]}[\Omega]$.*

Proof. By induction on $|C|$ with sub-induction on $nd(\Pi)$.

Suppose $C = X^q\,\vec{u}$, for some $\vec{u}$ and some $X^q \in supp(\Omega)$, and suppose $\Omega(X^q) = (\mathcal{R}, \Pi_S, S)$. Then $\Pi \in \mathcal{R}$ by Definition 15. By Definition 13 (**CR0**) we also have $\Pi[\rho] \in \mathcal{R}$. Otherwise, suppose $X^q \notin supp(\Omega)$. Then $\Pi \in \mathbf{NM}_{X^q}$ by Definition 15. By Lemma 9 we have $\Pi[\rho] \in \mathbf{NM}_{X^q}$, therefore $\Pi[\rho] \in \mathbf{RED}_{C[\rho]}[\Omega]$.

Otherwise, $C \neq X^q\,\vec{u}$ for any $\vec{u}$ and any parameter X^q. In this case, to apply the inner induction hypothesis, we need to show that $\Pi[\rho]$ is normalizable, which follows immediately from Lemmas 12 and 9. We distinguish several cases based on the last rule of Π:

- Suppose Π ends with mc, i.e., $\Pi = mc(\Pi_1, \ldots, \Pi_n, \Pi')$ for some $\Pi_1, \ldots, \Pi_n$ and Π'. By Lemma 7 every reduct of $\Pi[\rho]$, say Ξ, is the result of applying ρ to a reduct of Π. By the inner induction hypothesis (on the normalization degree), every reduct of $\Pi[\rho]$ is in $\mathbf{RED}_{C[\rho]}[\Omega]$, and therefore $\Pi[\rho]$ is also in $\mathbf{RED}_{C[\rho]}[\Omega]$ by Definition 15 (**P2**).
- Suppose Π ends with $\supset\mathcal{R}$, with the premise derivation Π'. In this case, $C = B \supset D$ for some B and D. Since $\Pi \in \mathbf{RED}_C[\Omega]$, by**P3**
$$\Pi'[\theta] \in \big(\mathbf{RED}_{B[\theta]}[\Omega] \Rightarrow \mathbf{RED}_{D[\theta]}[\Omega]\big) \tag{11}$$
for every θ. We need to show that
$$\Pi'[\rho][\delta] \in \big(\mathbf{RED}_{B[\rho][\delta]}[\Omega] \Rightarrow \mathbf{RED}_{D[\rho][\delta]}[\Omega]\big)$$
for every δ. Note that by Lemma 4 $\Pi'[\rho][\delta] = \Pi'[\rho \circ \delta]$, so this is just an instance of Statement (11) above.
- Π ends with $\mathrm{I}\mathcal{R}$ or $\mathrm{CI}\mathcal{R}$: This follows from Definition 15 and the fact that reducibility candidates are closed under substitution (condition **CR0** in Definition 13). In the case where Π ends with $\mathrm{I}\mathcal{R}$, we also need the fact that eigenvariable substitution commutes with parameter substitution (Lemma 5). In the case where Π ends with $\mathrm{CI}\mathcal{R}$, to establish $\Pi[\rho] \in \mathbf{RED}_{C[\rho]}[\Omega]$, we can use the same reducibility candidate that is used to establish $\Pi \in \mathbf{RED}_C[\Omega]$.
- Π ends with a rule other than mc, $\supset\mathcal{R}$, $\mathrm{I}\mathcal{R}$ or $\mathrm{CI}\mathcal{R}$: This case follows straightforwardly from the induction hypothesis. □

Lemma 14. *Let $\Omega = [\Omega', (\mathcal{R}, \Pi_S, S)/X^p]$ and C a formula such that $X^p \# C$. Then, for every Π, $\Pi \in \mathbf{RED}_C[\Omega]$ if and only if $\Pi \in \mathbf{RED}_C[\Omega']$.*

Proof. By induction on $|C|$ with sub-induction on $nd(\Pi)$.

($\Rightarrow$) Suppose $C = Y^q\,\vec{u}$ for some $Y^q \in supp(\Omega)$ and $\Omega(Y^q) = (\mathcal{R}', \Pi_I, I)$. Since $X^p \# C$, this means that $Y^q \in supp(\Omega')$ and $\Omega'(Y^q) = \Omega(Y^q)$. Then, obviously, $\Pi \in \mathbf{RED}_C[\Omega]$ iff $\Pi \in \mathbf{RED}_C[\Omega']$. If $Y^q \notin supp(\Omega)$, then $\mathbf{RED}_C[\Omega] = \mathbf{NM}_{Y^q}\,\vec{u} = \mathbf{RED}_C[\Omega']$.

Otherwise, suppose $C \neq Y^q\,\vec{u}$, and $\Pi \in \mathbf{RED}_C[\Omega]$. The latter implies that Π is normalizable. We show by induction on $nd(\Pi)$ that $\Pi \in \mathbf{RED}_C[\Omega']$. In most cases, this follows straightforwardly from the induction hypothesis. We show the interesting cases here:

- Suppose Π ends with $\supset\mathcal{R}$, i.e., $C = B \supset D$ for some B and D and Π is

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma, B[\Omega] \longrightarrow D[\Omega]\end{array}}{\Gamma \longrightarrow B[\Omega] \supset D[\Omega]} \supset\mathcal{R}.$$

  Note that since $X^p \# C$, we have that $B[\Omega] = B[\Omega']$ and $D[\Omega] = D[\Omega']$. Since $\Pi \in \mathbf{RED}_C[\Omega]$, we have

$$\Pi'[\rho] \in \big(\mathbf{RED}_{B[\rho]}[\Omega] \Rightarrow \mathbf{RED}_{D[\rho]}[\Omega]\big)$$

  for every ρ. Since $|B| < |C|$ and $|D| < |C|$, by the (outer) induction hypothesis, we have $\mathbf{RED}_{B[\rho]}[\Omega] = \mathbf{RED}_{B[\rho]}[\Omega']$ and $\mathbf{RED}_{D[\rho]}[\Omega] = \mathbf{RED}_{D[\rho]}[\Omega']$. Therefore, we also have that

$$\Pi'[\rho] \in \big(\mathbf{RED}_{B[\rho]}\big[\Omega'\big] \Rightarrow \mathbf{RED}_{D[\rho]}\big[\Omega'\big]\big)$$

  for every ρ. This means, by Definition 15, that $\Pi \in \mathbf{RED}_C[\Omega']$.
- Suppose Π ends with $\mathrm{I}\mathcal{R}$:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow D\,Y^q\,\vec{t}\end{array}}{\Gamma \longrightarrow q\,\vec{t}}\ \mathrm{I}\mathcal{R}$$

  where $q\,\vec{x} \stackrel{\mu}{=} D\,q\,\vec{x}$ and Y^q is a new parameter. We assume w.l.o.g. that $Y^q \# \Omega$. Note that since the body of a definition cannot contain occurrences of parameters, we also have $X^p \# D\,Y^q\,\vec{t}$. Suppose $\mathcal{S}$ is a reducibility candidate of type I, for some closed term I of the same syntactic type as q, and suppose Π_I is a normalizable derivation of $D\,I\,\vec{y} \longrightarrow I\,\vec{y}$ such that

$$\Pi_I[\vec{u}/\vec{y}] \in \big(\mathbf{RED}_{(D\,Y^q\,\vec{u})}\big[\Omega', (\mathcal{S}, \Pi_I, I)/Y^q\big] \Rightarrow \mathcal{S}\,\vec{u}\big) \tag{12}$$

  for every $\vec{u}$ of the appropriate type. To show that $\Pi \in \mathbf{RED}_C[\Omega']$ we need to show that

$$mc\big(\Pi'\big[(\Pi_I, I)/Y^q\big], \Pi_I[\vec{t}/\vec{y}]\big) \in \mathcal{S}\,\vec{t}.$$

  Since $|(D\,Y^q\,\vec{u})| < |p\,\vec{t}|$ by Lemma 1 we have, by the outer induction hypothesis,

$$\mathbf{RED}_{(D\,Y^q\,\vec{u})}\big[\Omega', (\mathcal{S}, \Pi_I, I)/Y^q\big] = \mathbf{RED}_{(D\,Y^q\,\vec{u})}\big[\Omega, (\mathcal{S}, \Pi_I, I)/Y^q\big].$$

  Hence, by (12), we also have

$$\Pi_I[\vec{u}/\vec{y}] \in \big(\mathbf{RED}_{(D\,Y^q\,\vec{u})}\big[\Omega, (\mathcal{S}, \Pi_I, I)/Y^q\big] \Rightarrow \mathcal{S}\,\vec{u}\big).$$

  Since $\Pi \in \mathbf{RED}_C[\Omega]$ (from the assumption), this means that

$$mc\big(\Pi'\big[(\Pi_I, I)/Y^q\big], \Pi_I[\vec{t}/\vec{y}]\big) \in \mathcal{S}\,\vec{t},$$

  and therefore $\Pi \in \mathbf{RED}_C[\Omega']$.
- Suppose Π ends with $\mathrm{CI}\mathcal{R}$:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma \longrightarrow I\,\vec{t}\end{array} \quad \begin{array}{c}\Pi_I \\ I\,\vec{y} \longrightarrow B\,I\,\vec{y}\end{array}}{\Gamma \longrightarrow q\,\vec{t}}\ \mathrm{CI}\mathcal{R}$$

  where $q\,\vec{x} \stackrel{\nu}{=} B\,q\,\vec{x}$. Since $\Pi \in \mathbf{RED}_C[\Omega]$, by Definition 15 (**P4**) there exist a parameter Y^q such that $Y^q \# \Omega$ and a reducibility candidate $(\mathcal{S} : I)$ such that $\Pi' \in \mathcal{S}$ and

$$\Pi'[\vec{u}/\vec{y}] \in \big(\mathcal{S}\,\vec{u} \Rightarrow \mathbf{RED}_{B\,Y^q\,\vec{u}}\big[\Omega, (\mathcal{S}, \Pi_I, I)/Y^q\big]\big) \tag{13}$$

  for every $\vec{u}$. To show $\Pi \in \mathbf{RED}_C[\Omega']$ we need to find a reducibility candidate satisfying **P4**. We choose $\mathcal{S}$ as that candidate. It remains to show that $\Pi'[\vec{u}/\vec{y}] \in (\mathcal{S}\,\vec{u} \Rightarrow \mathbf{RED}_{B\,Y^q\,\vec{u}}[\Omega', (\mathcal{S}, \Pi_I, I)/Y^q])$. This follows from (13) and the outer induction hypothesis, since

$$\mathbf{RED}_{B\,Y^q\,\vec{u}}\big[\Omega, (\mathcal{S}, \Pi_I, I)/Y^q\big] = \mathbf{RED}_{B\,Y^q\,\vec{u}}\big[\Omega', (\mathcal{S}, \Pi_I, I)/Y^q\big].$$

The converse, i.e., $\Pi \in \mathbf{RED}_C[\Omega']$ implies $\Pi \in \mathbf{RED}_C[\Omega]$, can be proved analogously. In particular, in the case where Π ends with $\mathrm{CI}\mathcal{R}$, we rely on the fact that the choice of the new parameter Y^q is immaterial, as long as it is new, so we can assume without loss of generality that $Y^q \neq X^p$. $\square$

Lemma 15. *Let Ω be a candidate substitution and S a closed term of type $\tau_1 \to \cdots \to \tau_n \to o$. Then the set $\mathcal{R} = \{\Pi \mid \Pi \in \mathbf{RED}_{S\,\vec{u}}[\Omega]$ for some $\vec{u}\}$ is a reducibility candidate of type $S[\Omega]$.*

Proof. Suppose $S = X^p$ for some $X^p \in supp(\Omega)$ and $\Omega(X^p) = (\mathcal{S}, \Pi, S)$. Then we have $\mathcal{R} = \mathcal{S}$, so $\mathcal{R}$ is a reducibility candidate of type S by assumption. If $S = X^p$, but $X^p \notin supp(\Omega)$, then $\mathcal{R} = \mathbf{NM}_{X^p}$, and by Lemma 11 $\mathcal{R}$ is also a reducibility candidate. Otherwise, $S \neq X^p$ for any parameter X^p. We need to show that $\mathcal{R}$ satisfies **CR0–CR5**. **CR0** follows from Lemma 13. **CR1** follows from Lemma 12, and the rest follows from Definition 15. $\square$

Lemma 16. *Let Ω be a candidate substitution and X^p a parameter such that $X^p \# \Omega$. Let S be a closed term of the same syntactic type as p and let*

$$\mathcal{R} = \left\{\Pi \;\middle|\; \Pi \in \mathbf{RED}_{S\,\vec{u}}[\Omega] \text{ for some } \vec{u}\right\}.$$

Suppose $[\Omega, (\mathcal{R}, \Psi, S[\Omega])/X^p]$ is a candidate substitution for some Ψ. Then

$$\mathbf{RED}_{C[S/X^p]}[\Omega] = \mathbf{RED}_C\big[\Omega, \big(\mathcal{R}, \Psi, S[\Omega]\big)/X^p\big].$$

Proof. By induction on $|C|$. If $C = X^p\,\vec{u}$, then

$$\mathbf{RED}_C\big[\Omega, \big(\mathcal{R}, \Psi, S[\Omega]\big)/X^p\big] = \mathcal{R}\,\vec{u} = \mathbf{RED}_{S\,\vec{u}}[\Omega]$$

by assumption. The other cases where C is $Y^q\,\vec{u}$ for some parameter $Y^q \neq X^p$ are straightforward. So suppose $C \neq Y^q\,\vec{u}$ for any $\vec{u}$ and any parameter Y^q. We show that for every Π, $\Pi \in \mathbf{RED}_{C[S/X^p]}[\Omega]$ iff $\Pi \in \mathbf{RED}_C[\Omega, (\mathcal{R}, \Psi, S[\Omega])/X^p]$. If X^p does not occur in C, then $C[S/X^p] = C$ and by Lemma 14 we have

$$\mathbf{RED}_{C[S/X^p]}[\Omega] = \mathbf{RED}_C[\Omega] = \mathbf{RED}_C\big[\Omega, \big(\mathcal{R}, \Psi, S[\Omega]\big)/X^p\big].$$

So assume that X^p is not vacuous in C. Let $\Omega' = [\Omega, (\mathcal{R}, \Psi, S[\Omega])/X^p]$.

- Suppose $\Pi \in \mathbf{RED}_{C[S/X^p]}[\Omega]$. Then Π is normalizable. We show, by induction on $nd(\Pi)$, that $\Pi \in \mathbf{RED}_C[\Omega']$. Most cases follow immediately from the induction hypothesis. The only interesting case is when Π ends with $\supset\mathcal{R}$, where $C = B \supset D$, for some B and D, and Π takes the form:

$$\dfrac{\begin{array}{c}\Pi' \\ \Gamma, B[S/X^p][\Omega] \longrightarrow D[S/X^p][\Omega]\end{array}}{\Gamma \longrightarrow B[S/X^p][\Omega] \supset D[S/X^p][\Omega]} \supset\mathcal{R}.$$

  Since $\Pi \in \mathbf{RED}_{C[S/X^p]}[\Omega]$, we have that

$$\Pi'[\rho] \in \big(\mathbf{RED}_{B[S/X^p][\rho]}[\Omega] \Rightarrow \mathbf{RED}_{D[S/X^p][\rho]}[\Omega]\big)$$

  for every ρ. By the outer induction hypothesis (on the size of C), we have

$$\Pi'[\rho] \in \big(\mathbf{RED}_{B[\rho]}\big[\Omega'\big] \Rightarrow \mathbf{RED}_{D[\rho]}\big[\Omega'\big]\big)$$

  hence $\Pi \in \mathbf{RED}_C[\Omega']$.
- The converse, i.e., $\Pi \in \mathbf{RED}_C[\Omega']$ implies $\Pi \in \mathbf{RED}_{C[S/X^p]}[\Omega]$, can be proved analogously. $\square$

References

[1] P. Aczel, An introduction to inductive definitions, in: J. Barwise (Ed.), Handbook of Mathematical Logic, in: Studies in Logic and the Foundations of Mathematics, vol. 90, North-Holland, Amsterdam, 1977, pp. 739–782, Chapter C.7.

[2] F. Baader, W. Snyder, Unification theory, in: J.A. Robinson, A. Voronkov (Eds.), Handbook of Automated Reasoning, Elsevier and MIT Press, 2001, pp. 445–532.

[3] D. Baelde, A linear approach to the proof-theory of least and greatest fixed points, PhD thesis, École Polytechnique, 2009.

[4] D. Baelde, Least and greatest fixed points in linear logic, ACM Trans. Comput. Log. 13 (1) (2012) 2.

[5] D. Baelde, D. Miller, Least and greatest fixed points in linear logic, in: LPAR, in: LNCS, Springer, 2007, pp. 92–106.

[6] D. Baelde, A. Gacek, D. Miller, G. Nadathur, A. Tiu, The Bedwyr system for model checking over syntactic expressions, in: F. Pfenning (Ed.), CADE, in: LNCS, vol. 4603, Springer, 2007, pp. 391–397.

[7] D. Baelde, D. Miller, Z. Snow, Focused inductive theorem proving, in: J. Giesl, R. Hähnle (Eds.), IJCAR, in: LNCS, vol. 6173, Springer, 2010, pp. 278–292.

[8] J. Brotherston, A. Simpson, Complete sequent calculi for induction and infinite descent, in: LICS, IEEE Computer Society, 2007, pp. 51–62.

[9] K.L. Clark, Negation as failure, in: J. Gallaire, J. Minker (Eds.), Logic and Data Bases, Plenum Press, New York, 1978, pp. 293–322.

[10] J. Despeyroux, A. Hirschowitz, Higher-order abstract syntax with induction in Coq, in: Fifth International Conference on Logic Programming and Automated Reasoning, June 1994, pp. 159–173.

[11] L.-H. Eriksson, A finitary version of the calculus of partial inductive definitions, in: L.-H. Eriksson, L. Hallnäs, P. Schroeder-Heister (Eds.), Proceedings of the Second International Workshop on Extensions to Logic Programming, in: LNAI, vol. 596, Springer-Verlag, 1991, pp. 89–134.
[12] A.P. Felty, A. Momigliano, Hybrid – a definitional two-level approach to reasoning with higher-order abstract syntax, J. Autom. Reasoning 48 (1) (2012) 43–105.
[13] A. Gacek, A framework for specifying, prototyping, and reasoning about computational systems, PhD thesis, University of Minnesota, September 2009.
[14] A. Gacek, D. Miller, G. Nadathur, A two-level logic approach to reasoning about computations, J. Autom. Reasoning 49 (2) (2012) 241–273.
[15] A. Gacek, D. Miller, G. Nadathur, Nominal abstraction, Inf. Comput. 209 (1) (2011) 48–73.
[16] E. Giménez, Un calcul de constructions infinies et son application a la verification des systemes communicants, PhD thesis, PhD 96-11, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, Dec. 1996.
[17] J.-Y. Girard, P. Taylor, Y. Lafont, Proofs and Types, Cambridge University Press, 1989.
[18] L. Hallnäs, Partial inductive definitions, Theor. Comput. Sci. 87 (1) (1991) 115–142.
[19] R. Harper, F. Honsell, G. Plotkin, A framework for defining logics, Journal of the ACM 40 (1) (1993) 143–184.
[20] B. Jacobs, J. Rutten, A tutorial on (co)algebras and (co)induction, Bulletin of the European Association for Theoretical Computer Science (Surveys and Tutorials) 62 (1997) 222–259.
[21] P. Martin-Löf, Hauptsatz for the intuitionistic theory of iterated inductive definitions, in: J.E. Fenstad (Ed.), Proceedings of the Second Scandinavian Logic Symposium, in: Studies in Logic and the Foundations of Mathematics, vol. 63, North-Holland, 1971, pp. 179–216.
[22] R. McDowell, D. Miller, Cut-elimination for a logic with definitions and induction, Theor. Comput. Sci. 232 (2000) 91–119.
[23] R. McDowell, D. Miller, Reasoning with higher-order abstract syntax in a logical framework, ACM Transactions on Computational Logic 3 (1) (2002) 80–136.
[24] R. McDowell, D. Miller, C. Palamidessi, Encoding transition systems in sequent calculus, Theor. Comput. Sci. 294 (3) (2003) 411–437.
[25] N.P. Mendler, Recursive types and type constraints in second-order lambda calculus, in: LICS, IEEE Computer Society, 1987, pp. 30–36.
[26] M. Miculan, K. Yemane, A unifying model of variables and names, in: V. Sassone (Ed.), FoSSaCS, in: LNCS, vol. 3441, Springer, 2005, pp. 170–186.
[27] D. Miller, A logic programming language with lambda-abstraction, function variables, and simple unification, in: P. Schroeder-Heister (Ed.), Extensions of Logic Programming, in: LNAI, vol. 475, Springer-Verlag, 1991, pp. 253–281.
[28] D. Miller, A. Tiu, A proof theory for generic judgments, ACM Trans. Comput. Logic 6 (4) (2005) 749–783.
[29] A. Momigliano, S. Ambler, Multi-level meta-reasoning with higher order abstract syntax, in: A. Gordon (Ed.), FOSSACS'03, in: LNCS, vol. 2620, Springer-Verlag, 2003, pp. 375–392.
[30] A. Momigliano, A. Tiu, Induction and co-induction in sequent calculus, in: S. Berardi, M. Coppo, F. Damiani (Eds.), Types, in: LNCS, vol. 3085, Springer, 2003, pp. 293–308.
[31] M. Pagani, L.T. de Falco, Strong normalization property for second order linear logic, Theor. Comput. Sci. 411 (2) (2010) 410–444.
[32] L.C. Paulson, Mechanizing coinduction and corecursion in higher-order logic, J. Logic Comput. 7 (2) (1997) 175–204.
[33] F. Pfenning, Logical frameworks, in: A. Robinson, A. Voronkov (Eds.), Handbook of Automated Reasoning, Elsevier Science Publisher and MIT Press, 2001, pp. 1063–1147, Chapter 17.
[34] F. Pfenning, C. Paulin-Mohring, Inductively defined types in the calculus of constructions, in: M. Main, A. Melton, M. Mislove, D. Schmidt (Eds.), Proceedings of the Fifth Conference on the Mathematical Foundations of Programming Semantics, in: LNCS, vol. 442, Springer-Verlag, 1989, pp. 209–228.
[35] L. Santocanale, A calculus of circular proofs and its categorical semantics, in: M. Nielsen, U. Engberg (Eds.), FoSSaCS, in: LNCS, vol. 2303, Springer, 2002, pp. 357–371.
[36] U. Schöpp, Modelling generic judgments, Electr. Notes Theor. Comput. Sci. 174 (5) (2007) 19–35.
[37] P. Schroeder-Heister, Definitional reflection and the completion, in: R. Dyckhoff (Ed.), Proceedings of the 4th International Workshop on Extensions of Logic Programming, in: LNAI, vol. 798, Springer-Verlag, 1993, pp. 333–347.
[38] P. Schroeder-Heister, Rules of definitional reflection, in: M. Vardi (Ed.), Eighth Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, IEEE, June 1993, pp. 222–232.
[39] J. Slaney, Solution to a problem of Ono and Komori, J. Philos. Logic 18 (1989) 103–111.
[40] C. Spenger, M. Dams, On the structure of inductive reasoning: Circular and tree-shaped proofs in the μ-calculus, in: A. Gordon (Ed.), FOSSACS'03, in: LNCS, vol. 2620, Springer-Verlag, 2003, pp. 425–440.
[41] A. Tiu, A logical framework for reasoning about logical specifications, PhD thesis, Pennsylvania State University, May 2004.
[42] A. Tiu, A logic for reasoning about generic judgments, Electr. Notes Theor. Comput. Sci. 174 (5) (2007) 3–18.
[43] A. Tiu, D. Miller, Proof search specifications of bisimulation and modal logics for the π-calculus, ACM Trans. Comput. Logic 11 (2) (2010) 1–35.
[44] A. Tiu, A. Momigliano, Induction and co-induction in sequent calculus, arXiv:0812.4727, 2008.