Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Theoretical Computer Science 360 (2006) 42–76

Theoretical Computer Science

www.elsevier.com/locate/tcs

On structuring proof search for first order linear logic

Paola Bruscoli*, Alessio Guglielmi[1]

Technische Universität Dresden, Hans-Grundig-Str. 25, 01062 Dresden, Germany

Received 24 August 2004; received in revised form 25 November 2005; accepted 28 November 2005

Communicated by G. Levi

Abstract

Full first-order linear logic can be presented as an abstract logic programming language in Miller's system *Forum*, which yields a sensible operational interpretation in the 'proof search as computation' paradigm. However, *Forum* still has to deal with syntactic details that would normally be ignored by a reasonable operational semantics. In this respect, *Forum* improves on Gentzen systems for linear logic by restricting the language and the form of inference rules. We further improve on *Forum* by restricting the class of formulae allowed, in a system we call *G-Forum*, which is still equivalent to full first-order linear logic. The only formulae allowed in *G-Forum* have the same shape as *Forum* sequents: the restriction does not diminish expressiveness and makes *G-Forum* amenable to proof theoretic analysis. *G-Forum* consists of two (big) inference rules, for which we show a cut elimination procedure. This does not need to appeal to finer detail in formulae and sequents than is provided by *G-Forum*, thus successfully testing the internal symmetries of our system.

Keywords: Proof theory; Linear logic; Cut elimination; Proof search; Forum

1. Introduction

Forum [8,9] is a presentation of linear logic, conceived by Miller and based on previous work by Andreoli [1], which only produces uniform proofs [10]. This guarantees that a sensible computational interpretation of proof search is possible. Surprisingly, Forum is complete for linear logic; this contrasts with the situation in classical logic, where a complete presentation that only produces uniform proofs is not possible [11]. Given linear logic's flexibility in interpreting a broad range of computational situations, Forum represents a major step towards practical applications.

This paper is motivated by the search for adequate operational models of Forum, especially behavioural models like labelled event structures, which describe causal relations between events. We will not deal with labelled event structures in this paper, but those familiar with them should be able to read them out of the derivations that we show. This paper is a purely proof theoretic investigation where we design a deductive system equivalent to Forum and we study its cut elimination procedure. This brings proof search in Forum closer to the operational behaviour of formulae as suggested in [8].

* Corresponding author. Tel.: +49 351 463 38280; fax: +49 351 463 38341.
E-mail addresses: Paola.Bruscoli@Inf.TU-Dresden.DE (P. Bruscoli), A.Guglielmi@Bath.Ac.UK (A. Guglielmi).

[1] Present address: University of Bath, United Kingdom.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.11.047

A proof in Forum mainly consists of small, mostly deterministic steps, corresponding to applying each of several inference rules. This determinism is not a surprise, of course, since Forum has been designed precisely for the purpose of reducing and isolating non-determinism. However, most of these steps do not correspond to 'interesting' observations in a computation. For example, two applications of a rule for $\parr$ are necessary for decomposing $A \parr (B \parr C)$ into its constituents A, B, C; moreover, the order of application of rules would be different for decomposing $(A \parr B) \parr C$, but the result would be the same. Since in such a case one is interested only in the result A, B, C, the detail about the order of applications of the rule for $\parr$ is not necessary. This is a trivial case of a more general phenomenon in which a result is essentially deterministic, but its computation depends on irrelevant factors like a casual decision about associations of formulae connected by $\parr$ connectives. In other words, one might be interested in identifying derivations that differ for such details as the ones shown above.

To get Forum, Miller imposed certain restrictions on the sequents, inference rules, and possible connectives of linear logic, but he left formula building free. In this paper, and limiting ourselves to the first order case, we restrict the class of formulae allowed, along lines already imagined by Miller in [8], and we design correspondingly an equivalent system called G-Forum. By doing this, we have that formulae drive the construction of proofs in a very structured way, which allows us to individuate big chunks of derivations that essentially behave in a deterministic way: these will be the building blocks of our desired behavioural semantics. The restriction on formulae makes them isomorphic to the sequents in Forum. Thus, we obtain two results: we get derivations which are closer to the operational properties we want to observe, and we also get a clean correspondence between the object level (the language of formulae) and the meta level (sequents).

An important question is whether G-Forum has good proof-theoretic standing, and this is the subject of this paper. We test the internal harmony of G-Forum the classic way: we show a cut elimination procedure for our system. Of course, since G-Forum is equivalent to first order Forum, and so it is complete for first order linear logic, there is no need to show that the cut rule is admissible: this is a consequence of completeness. On the other hand, this fact does not tell us anything about constructively eliminating cuts *while staying inside* G-Forum. This is an important result if we want to use G-Forum as a specification language for a semi-automatic verifier, for example, where a cut rule and procedural cut elimination would find uses. We show that it is not necessary to resort to Forum (or, worse, to a sequent calculus presentation of linear logic) for this to work: G-Forum rules are enough.

In other words, the coarser granularity of G-Forum with respect to Forum is sufficient for cut elimination and the proof transformations associated to it. Arguably, it also guarantees better semantic properties. This result is very delicate and it depends crucially on finely tuning the correspondence between the object level and the meta level. Actually, the exact syntax of formulae has been obtained from the cut elimination proof by the (usual, in proof theory) trial and mistake method.

In Section 2, we give a quick account of first order Forum, then we develop G-Forum in Section 3. The cut elimination proof is in Section 4. There are two versions of Forum: [8] and [9]; in this paper we refer to the LICS '94 version in [8].

2. First order Forum

This section is a quick account of Miller's [8], restricted to the first order case and with some minor technical and notational changes. We will tend to use the word 'Forum' to indicate the first order version of Forum.

We deal with first order formal systems, and the following conventions apply.

2.1. Notation. The letters h, k and l denote *natural numbers*, and i and j are used as *indices* on natural numbers. *Multisets* are denoted by braces as in $\{\ldots\}_+$; *multiset union* is $\uplus$ and the *empty multiset* is $\emptyset_+$. If a belongs to multiset M we write $a \Subset M$.

2.2. Definition. *First order variables* are denoted by x, y and z; *terms* are denoted by t, *atoms* by a, b, c, …, $a(t_1, \ldots, t_h)$, $b(\ldots)$, $c(\ldots)$, …. *Sequences* are denoted in vector notation, as in $\forall \vec{x}.a(\vec{t})$. *Formulae* are denoted by F and other letters which will be introduced later on. Formulae are considered equal under α-conversion.

This work is founded on linear logic; we are mainly interested in its first order sequent calculus presentation. We refer to the literature for details, especially to Girard's [5].

2.3. Definition. The formal system of full *first order linear logic*, in its Gentzen's sequents presentation, and its language, are both denoted by FOLL. Formulae in FOLL are freely built from first order atoms and *constants* 1, $\bot$, $\top$, 0 by using *binary connectives* $\otimes$, $\parr$, $\mathbin{\&}$, $\oplus$, $\multimap$, *modalities* $!$, $?$, *negation* ${}^{\bot}$ and the *quantifiers* $\forall$ and $\exists$. Constants 1, $\bot$ and connectives $\otimes$, $\parr$ and $\multimap$ are called the *multiplicatives*; $\top$, 0, $\mathbin{\&}$ and $\oplus$ are called the *additives*. *Equivalence* is written $\equiv$. In linear logic $F \equiv F'$ iff $(F \multimap F') \mathbin{\&} (F' \multimap F)$ is provable.

Intuitionistic implication $\Rightarrow$ admits the well-known decomposition $F \Rightarrow F' \equiv\, !F \multimap F'$; we can consider $\Rightarrow$ part of our language.

2.4. Definition. The *binary connective* $\Rightarrow$ is introduced such that $F \Rightarrow F'$ is equivalent to $!F \multimap F'$.

2.5. Definition. Multiplicative connectives, except for $\multimap$, take precedence over additive ones; implications are the weakest connectives; modalities and quantifiers are stronger than binary connectives; negation takes precedence over everything. Implications associate to the right. Whenever possible, we omit parentheses.

For example, $!\forall x.a^{\bot} \multimap b \mathbin{\&} c \parr d \Rightarrow e$ stands for $\big(!(\forall x.(a^{\bot}))\big) \multimap \big((b \mathbin{\&} (c \parr d)) \Rightarrow e\big)$.

We briefly introduce the Forum formal system, in its first order version. The presentation corresponds to the one in [8], restricted to the first order case and with some minor modifications. An alternative and more detailed exposition can be found in [9]; our results do not trivially extend to that version, although we do expect them to work with a relatively minor effort.

2.6. Definition. The language of *first order* Forum is the subset of FOLL freely built over atoms and the constants $\bot$ and $\top$ by use of the binary connectives $\parr$, $\mathbin{\&}$, $\multimap$ and $\Rightarrow$ and of the quantifier $\forall$. We will say 'Forum' instead of 'first order Forum.' Generic Forum formulae are denoted by A and B.

So, Forum presents fewer connectives than FOLL, by getting rid of some of the redundant ones. It is not difficult to prove the following ellquivalences in FOLL:

$$\begin{array}{rclrcl}
1 & \equiv & \bot^{\bot}, & 0 & \equiv & \top^{\bot}, \\
F \otimes F' & \equiv & (F^{\bot} \parr F'^{\bot})^{\bot}, & F \oplus F' & \equiv & (F^{\bot} \mathbin{\&} F'^{\bot})^{\bot}, \\
!F & \equiv & (F \Rightarrow \bot)^{\bot}, & ?F & \equiv & F^{\bot} \Rightarrow \bot,
\end{array}$$
$$\begin{aligned}
\exists x.F &\equiv (\forall x.F^{\bot})^{\bot}, \\
F^{\bot} &\equiv F \multimap \bot.
\end{aligned}$$

Then, one can equivalently write any FOLL formula into the Forum language.

2.7. Definition. *Sequents* are expressions of the form

$$\begin{bmatrix} \Psi \\ \Gamma \end{bmatrix} A \vdash \begin{bmatrix} \\ \Lambda \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \Psi \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} \Xi \\ \Lambda \end{bmatrix},$$

where all formulae are Forum formulae and

- Ψ is a finite multiset of formulae (the *left classical context* or *classical program*);
- Γ is a finite multiset of formulae (the *left linear context* or *linear program*);
- A is a formula (the *left focused formula*);
- Ξ is a finite sequence of formulae (the *right linear context*);
- Λ is a finite multiset of atoms (the *atomic context*).

Γ, Ξ and Λ are collectively referred to as the *linear context*. Ψ and Γ together are called the *program*. In the following, Ψ, Γ, Ξ and Λ respectively stand for multisets, multisets and sequences of formulae and multisets of atoms. We write 'Γ, A', or 'A, Γ', instead of $\Gamma \uplus \{A\}_{+}$ and 'Γ, Γ'' instead of $\Gamma \uplus \Gamma'$. Sequents are denoted by Σ. Sequents where no focused formula is present and Ξ is empty are called *state sequents* and are written

$$\begin{bmatrix} \Psi \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} \Lambda \end{bmatrix}.$$

Structural Rules

$$\mathsf{i}\,\frac{}{\begin{bmatrix}\Psi\\ \ \end{bmatrix} a \vdash \begin{bmatrix}\ \\ a\end{bmatrix}} \qquad \mathsf{d}_{\mathsf{L}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma, A\end{bmatrix} \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}} \qquad \mathsf{d}_{\mathsf{C}}\,\frac{\begin{bmatrix}\Psi, A\\ \Gamma\end{bmatrix} A \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi, A\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}} \qquad \mathsf{a}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ a, \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}a, \Xi\\ \Lambda\end{bmatrix}}$$

Left Rules

Right Rules

$$\bot_{\mathsf{L}}\,\frac{}{\begin{bmatrix}\Psi\\ \ \end{bmatrix} \bot \vdash \begin{bmatrix}\ \end{bmatrix}} \qquad \bot_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\bot, \Xi\\ \Lambda\end{bmatrix}}$$

$$\top_{\mathsf{R}}\,\frac{}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\top, \Xi\\ \Lambda\end{bmatrix}}$$

$$\parr_{\mathsf{L}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix} \quad \begin{bmatrix}\Psi\\ \Gamma'\end{bmatrix} B \vdash \begin{bmatrix}\ \\ \Lambda'\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma, \Gamma'\end{bmatrix} A \parr B \vdash \begin{bmatrix}\ \\ \Lambda, \Lambda'\end{bmatrix}} \qquad \parr_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A, B, \Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A \parr B, \Xi\\ \Lambda\end{bmatrix}}$$

$$\&_{\mathsf{LL}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A \mathbin{\&} B \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}} \qquad \&_{\mathsf{LR}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} B \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A \mathbin{\&} B \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}} \qquad \&_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A, \Xi\\ \Lambda\end{bmatrix} \quad \begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}B, \Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A \mathbin{\&} B, \Xi\\ \Lambda\end{bmatrix}}$$

$$\multimap_{\mathsf{L}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A\\ \Lambda\end{bmatrix} \quad \begin{bmatrix}\Psi\\ \Gamma'\end{bmatrix} B \vdash \begin{bmatrix}\ \\ \Lambda'\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma, \Gamma'\end{bmatrix} A \multimap B \vdash \begin{bmatrix}\ \\ \Lambda, \Lambda'\end{bmatrix}} \qquad \multimap_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma, A\end{bmatrix} \vdash \begin{bmatrix}B, \Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A \multimap B, \Xi\\ \Lambda\end{bmatrix}}$$

$$\Rightarrow_{\mathsf{L}}\,\frac{\begin{bmatrix}\Psi\\ \ \end{bmatrix} \vdash \begin{bmatrix}A\\ \ \end{bmatrix} \quad \begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} B \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A \Rightarrow B \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}} \qquad \Rightarrow_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi, A\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}B, \Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A \Rightarrow B, \Xi\\ \Lambda\end{bmatrix}}$$

$$\forall_{\mathsf{L}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} A[t/x] \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \forall x.A \vdash \begin{bmatrix}\ \\ \Lambda\end{bmatrix}} \qquad \forall_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}A[y/x], \Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\forall x.A, \Xi\\ \Lambda\end{bmatrix}}$$

where y is not free in the conclusion

Fig. 1. The first order *Forum* proof system.

2.8. Definition. An *inference rule* is an expression of the form $r\frac{\Sigma_1\ \dots\ \Sigma_h}{\Sigma}$, where $h\geqslant 0$, sequents $\Sigma_1, \dots, \Sigma_h$ are the *premises* of the rule, Σ is its *conclusion* and r is the *name* of the rule. An inference rule with no premises is called an *axiom*.

2.9. Definition. Let Forum be the first order proof system defined by inference rule schemes in Fig. 1. Structural rules are: i (*identity*), d_{L} (*decide linear*), d_{C} (*decide classical*), a (*atom*). Logical rules, divided into left and right ones, are: $\bot_{\mathsf{L}}$, $\bot_{\mathsf{R}}$ (*bottom*); $\top_{\mathsf{R}}$ (*top*, there is no left rule); $\parr_{\mathsf{L}}$, $\parr_{\mathsf{R}}$ (*par*); $\&_{\mathsf{LL}}$, $\&_{\mathsf{LR}}$, $\&_{\mathsf{R}}$ (*with*); $\multimap_{\mathsf{L}}$, $\multimap_{\mathsf{R}}$ (*linear implication*); $\Rightarrow_{\mathsf{L}}$, $\Rightarrow_{\mathsf{R}}$ (*intuitionistic implication*); $\forall_{\mathsf{L}}$, $\forall_{\mathsf{R}}$ (*universal quantification*).

Consider proofs in Forum in a bottom-up reading. In the absence of a left focused formula, the right linear context is acted upon by right rules until it is empty; at that point a formula becomes focused, in a $\mathsf{d_L}$ or $\mathsf{d_C}$ rule. Then left rules only are applicable, until new formulae reach the right linear context, through $\multimap_\mathsf{L}$ and $\Rightarrow_\mathsf{L}$ rules. Proofs in Forum are said to be *uniform* [10,8,9].

Our system's major differences with Forum as presented in [8] are: (1) our classical context is a multiset while in [8] it is a set; (2) our atomic context is a multiset while in [8] it is a sequence. These differences do not affect provability (and uniformity of proofs), as it can be proved trivially.

Representing derivations as directed trees whose nodes are sequents is typographically advantageous, especially in the cut elimination proof. The direction of the arrows corresponds to the tree growth during the search for a proof. It should be clear that there is no difference between our non-standard notation and the usual one.

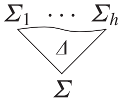
2.10. Definition. To every instance of an inference rule $r\,\frac{\Sigma_1\ \ldots\ \Sigma_h}{\Sigma}$, when $h > 0$, an *elementary derivation*

$$\begin{tikzcd} \Sigma_1 & \cdots & \Sigma_h \\ & \Sigma \arrow[ul, "r"] \arrow[ur, "r"'] & \end{tikzcd}$$

corresponds, i.e., a labelled directed tree whose root is labelled Σ, whose leaves are labelled $\Sigma_1, \ldots, \Sigma_h$ and whose arcs are labelled r; when $h = 0$ the corresponding elementary derivation is

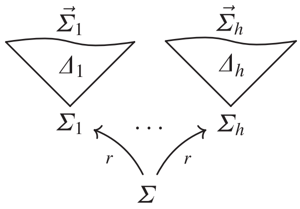
$$\begin{tikzcd} \circ \\ \Sigma \arrow[u, "r"] \end{tikzcd} ,$$

where $\circ$ is a mark distinct from every sequent. *Derivations* are non-empty, finite directed trees whose root is labelled by a sequent and whose other nodes are labelled by sequents or $\circ$ marks and such that every maximal subtree of depth 1 is an elementary derivation. Derivations are denoted by Δ. Given a derivation Δ, its *premises* are the labels of the leaves of Δ other than the $\circ$ ones; its *conclusion* is the sequent labelling the root of Δ. A derivation Δ such that its premises are $\Sigma_1, \ldots, \Sigma_h$ and its conclusion is Σ can be represented as



.

Sometimes the name of the derivation is not shown. If Δ is the derivation



,

where $h \geqslant 0$, we define its *depth* $\mathsf{d}(\Delta)$ as the depth of the corresponding tree, i.e., $\mathsf{d}(\Delta) = \max\{\mathsf{d}(\Delta_1), \ldots, \mathsf{d}(\Delta_h)\} + 1$, where, for every sequent Σ, it holds $\mathsf{d}(\Sigma) = \mathsf{d}(\circ) = 0$. If Δ has no premises we say that Δ is a *proof*. Proofs are denoted by Π. We say that Π *proves* (or *is a proof of*) its conclusion. We say that a formula A is *provable* in Forum, or that Forum *proves* A, if a proof of $\begin{bmatrix} \\ \end{bmatrix} \vdash \begin{bmatrix} A \\ \end{bmatrix}$ exists.

For example, the premises of the derivation in Fig. 2 are $\left\{ \begin{bmatrix} \\ \end{bmatrix} a \vdash \begin{bmatrix} \\ a \end{bmatrix}, \begin{bmatrix} \\ \end{bmatrix} a \vdash \begin{bmatrix} \\ a \end{bmatrix} \right\}_+$ and its conclusion is $\begin{bmatrix} \\ \end{bmatrix} a \parr (b \parr a) \vdash \begin{bmatrix} \\ a, a, b \end{bmatrix}$. This derivation can be completed into a proof by applying two identity rules to its premises.

Please note that arcs are not 'independent' in the growth process of a derivation: all arcs propagating from a node correspond to the application of the same inference rule.

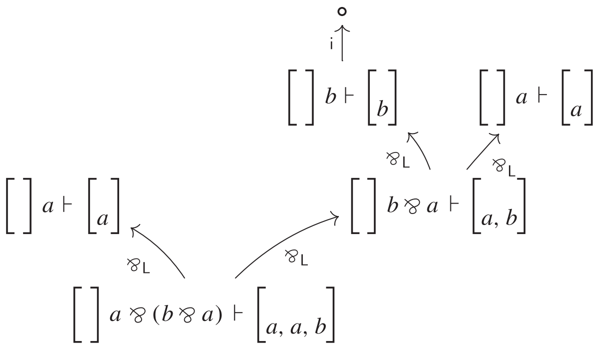


Fig. 2. Example of derivation.

By looking at Fig. 1 it is clear that if we make the classical context a set (as Miller does), derivability is not affected. In fact, the only impact is on the $\Rightarrow_{\mathsf{R}}$ rule, but things do not change, because the classical context is implicitly subject to weakening in all axioms.

2.11 Theorem. *Every* Forum *formula is provable in* Forum *if and only if it is provable in* FOLL (Miller [8,9]).

Since for every formula in FOLL an equivalent formula in Forum can be found, the Forum formal system can be used to prove formulae in FOLL.

3. Derivations at a higher level of abstraction

Consider a formula $\delta = G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k$. In Forum, in a bottom-up construction of a derivation, from $\begin{bmatrix} \ \\ \ \end{bmatrix} \vdash \begin{bmatrix} \delta \\ \ \end{bmatrix}$ we are always led to the state sequents $\begin{bmatrix} G_1, \dots, G_{k''} \\ H_1, \dots, H_{k'} \end{bmatrix} \vdash \begin{bmatrix} \ \\ a_1, \dots, a_k \end{bmatrix}$. Let us call *clauses* formulae like δ, where formulae G_i and H_j, called *goals*, are of the form $\forall \vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)$, and where in the $\&$ conjunction only clauses are allowed.

In this section, we derive a proof system equivalent to FOLL. The new proof system is in fact the old Forum proof system seen at a coarser abstraction level: rules are essentially macro derivations composed of many Forum rules, and the only formulae allowed are goals and clauses.

3.1. Goals and clauses

We define goals and clauses, which are Forum formulae of a constrained shape; then we show that their language is equivalent to Forum and then to FOLL. We borrow from Miller the terminology on goals and clauses, and the reader should be aware that their use is more general than in standard logic programming, where clauses operate on goals in a clear hierarchical relation. In our formalism, goals and clauses are mutually recursive objects that only superficially bear a resemblance to goals and clauses of traditional logic programming.

3.1.1. Definition. Goals and clauses are recursively defined this way:

(1) A *goal* is a formula of the form

$$\forall \vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h),$$

where $\vec{x}$ can be empty, $h \geqslant 0$ and every δ_i is a clause. When $h = 0$ a goal is $\forall \vec{x}.\top$.

(2) A *clause* δ is a formula of the form

$$G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k,$$

where $k, k', k'' \geqslant 0$, formulae G_i and H_i are goals and formulae a_i are atoms. Goals G_i are called the *classical premises* of δ, goals H_i are its *linear premises* and $a_1 \parr \cdots \parr a_k$ is the *head* of the clause. We define $\mathsf{hd}(\delta) = \{a_1, \ldots, a_k\}_+$, $\mathsf{lp}(\delta) = \{H_1, \ldots, H_{k'}\}_+$ and $\mathsf{cp}(\delta) = \{G_1, \ldots, G_{k''}\}_+$. When $k = 0$ the head is $\bot$. When $k' = 0$ and $k'' = 0$ clauses assume the following special forms, respectively:

$$G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow a_1 \parr \cdots \parr a_k \quad \text{and} \quad H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k .$$

The letters G and H always denote goals and the letter δ always denotes clauses.

Clearly, a clause is also a goal.

The shape of goals and clauses is not due to chance, of course. On a technical level, it is motivated by the desire of keeping the cut elimination procedure *inside* the system we are going to define. This means that we would not consider acceptable eliminating cuts by resorting to the more primitive level of abstraction in which generic formulae and Forum inference rules are available. The only way to convince oneself of this is to try and modify the definition and see the impact on the cut elimination procedure. There are possibly many solutions to this problem, and the one we present here is probably only one of many.

However, there is a better explanation, which also offers a unique solution, the one we adopt: goals correspond to the shape of a proof tree, and clauses correspond to state sequents. In fact, a goal stands for the collection of branches of a tree, conveniently quantified universally (note that the branches of a derivation tree may share variables). Every branch of a derivation tree ends in a state sequent, as we will see, and this corresponds to a clause. The mutual recursion between goals and clauses corresponds to the phases in the construction of a proof that we are going to explore in the rest of the paper. There is a certain mysticism in this correspondence, and we are not sure we really understand it enough; for the time being we content ourselves with seeing that it works. But let us now get back to the properties of goals and clauses.

In cut-free sequent systems enjoying the subformula property, like the one we are dealing with, various fragments, which differ in the connectives allowed, can be cut out of bigger ones, while maintaining provability unaffected in the corresponding languages. For example, we could take the fragment of Forum in which $\&$ is not allowed; since, reading proofs bottom-up, no rule can introduce connectives not already present in its conclusion, provability for formulae not containing $\&$ would not be influenced. There is sort of an independence, or modularity, among connectives, which we want to preserve, because it is a valuable property in language design.

We show two ways of getting equivalence between goals and clauses and generic formulae; the first one, with goals, respects independence of connectives by using in an essential way distributivity of $\parr$ over $\&$.

3.1.2. Theorem. *Every formula in* FOLL *is equivalent to a goal in* Forum.

Proof. We already know that for every formula in linear logic there are equivalent formulae in Forum. We show that, taken any formula in Forum, we can exhibit an equivalent goal.
We use the following absorption equivalences:
(1) $F \parr \bot \equiv F$.
(2) $F \parr \top \equiv \top$.
(3) $F \& \top \equiv F$.
We also use the following equivalences:
(4) $F \parr (F' \& F'') \equiv (F \parr F') \& (F \parr F'')$.
(5) $\forall x.F \parr F' \equiv \forall x.(F \parr F')$ whenever x is not free in F'.
(6) $\forall x.F \& F' \equiv \forall x.(F \& F')$ whenever x is not free in F'.
Let A be a formula in Forum: the proof is by induction on its structure.
Basis cases:
- A is an atom.
- $A = \bot$.
- $A = \top$.

In the cases above, A is a goal.

Inductive cases: Given B and B', by the induction hypothesis, we suppose we are also given two goals G and G' such that

$$\begin{aligned} B &\equiv G = \forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h),\\ B' &\equiv G' = \forall\vec{y}.(\delta'_1 \mathbin{\&} \cdots \mathbin{\&} \delta'_{h'}), \end{aligned}$$

where $\vec{x}$ and $\vec{y}$ may be empty and h and h' may be 0. The following cases may occur.

- $A = B \parr B'$. By applications of equivalence (5) and renaming of bounded variables, if necessary, we get

  $$A \equiv \forall\vec{z}.\big((\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h) \parr (\delta'_1 \mathbin{\&} \cdots \mathbin{\&} \delta'_{h'})\big).$$

  If $h = 0$ or $h' = 0$ we can conclude that $A \equiv \forall\vec{z}.\top$, by making use of equivalence (2). Otherwise, we may repeatedly apply equivalence (4) above, and we get

  $$A \equiv \forall\vec{z}.\Big(\big((\delta_1 \parr \delta'_1) \mathbin{\&} \cdots \mathbin{\&} (\delta_h \parr \delta'_1)\big) \mathbin{\&} \cdots \mathbin{\&} \big((\delta_1 \parr \delta'_{h'}) \mathbin{\&} \cdots \mathbin{\&} (\delta_h \parr \delta'_{h'})\big)\Big).$$

  For $1 \leqslant i \leqslant h$ and $1 \leqslant j \leqslant h'$, let

  $$\begin{aligned} \delta_i &= G^i_1 \Rightarrow \cdots \Rightarrow G^i_{h''_i} \Rightarrow H^i_1 \multimap \cdots \multimap H^i_{h'_i} \multimap a^i_1 \parr \cdots \parr a^i_{h_i},\\ \delta'_j &= G'^j_1 \Rightarrow \cdots \Rightarrow G'^j_{k''_j} \Rightarrow H'^j_1 \multimap \cdots \multimap H'^j_{k'_j} \multimap a'^j_1 \parr \cdots \parr a'^j_{k_j}. \end{aligned}$$

  Since $F \Rightarrow F' \equiv ! F \multimap F'$ and $F \multimap F' \equiv F^\perp \parr F'$, commutativity of $\parr$ suffices to show that

  $$\begin{aligned} \delta_i \parr \delta'_j \equiv{}& G^i_1 \Rightarrow \cdots \Rightarrow G^i_{h''_i} \Rightarrow G'^j_1 \Rightarrow \cdots \Rightarrow G'^j_{k''_j} \Rightarrow H^i_1 \multimap \cdots \multimap H^i_{h'_i} \multimap H'^j_1 \multimap \cdots \multimap H'^j_{k'_j}\\ &\multimap a^i_1 \parr \cdots \parr a^i_{h_i} \parr a'^j_1 \parr \cdots \parr a'^j_{k_j}. \end{aligned}$$

  Special cases where $h_i = 0$ or $k_j = 0$ are handled by equivalence (1) above.
- $A = B \mathbin{\&} B'$. By applications of equivalence (6) and renaming of bounded variables, if necessary, we get

  $$A \equiv \forall\vec{z}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h \mathbin{\&} \delta'_1 \mathbin{\&} \cdots \mathbin{\&} \delta'_h).$$

  If $h = 0$ or $h' = 0$ use equivalence (3).
- $A = B \multimap B'$. By using equivalences (5) and (4), and by renaming bounded variables if necessary, we have

  $$\begin{aligned} A &\equiv G^\perp \parr \forall\vec{y}.(\delta'_1 \mathbin{\&} \cdots \mathbin{\&} \delta'_h)\\ &\equiv \forall\vec{z}.(G^\perp \parr (\delta'_1 \mathbin{\&} \cdots \mathbin{\&} \delta'_h))\\ &\equiv \forall\vec{z}.((G^\perp \parr \delta'_1) \mathbin{\&} \cdots \mathbin{\&} (G^\perp \parr \delta'_h)). \end{aligned}$$

  By commutativity of $\parr$ it is easily seen that every $(G^\perp \parr \delta'_i)$ is a clause. If $B' \equiv \top$ then $A \equiv \top$.
- $A = B \Rightarrow B'$. The argument goes as in the previous case.
- $A = \forall x.B$. Trivial.

3.1.3. Corollary. *Every formula in* FOLL *is equivalent to a clause.*

Proof. Let F be a formula and $G \equiv F$, where G is obtained as in Theorem 3.1.2. Then, $(G \multimap \bot) \multimap \bot$ is a clause equivalent to G. □

From the proof of the theorem we can derive an obvious algorithm that transforms a Forum formula into a goal. If A is a Forum formula and G the equivalent goal found by the algorithm, then the set of connectives appearing in G is not greater than that of A, which is of course an important modularity property. On the other hand, the translation could transform a linear logic formula or a generic Forum formula into a bigger and not simply correlated goal in Forum.

We are going to show now an alternative, more direct translation of Forum generic formulae into clauses. The new translation does not respect independence of all connectives, because it introduces $\multimap$ and $\bot$, what could be a mild constraint in some circumstances. We can do so by using a double negation trick, as follows.

$$\begin{array}{c}
\begin{bmatrix}\Psi, G_1, \dots, G_{k''}\\ \Gamma, H_1, \dots, H_{k'}\end{bmatrix} \vdash \begin{bmatrix} \hfill \Xi\\ a_1, \dots, a_k, \Lambda\end{bmatrix}\\
{\scriptstyle(\parr_{\mathsf{R}}\ \text{or}\ \mathsf{a})^\star}\Big\uparrow\\
\begin{bmatrix}\Psi, G_1, \dots, G_{k''}\\ \Gamma, H_1, \dots, H_{k'}\end{bmatrix} \vdash \begin{bmatrix} a_1 \parr \cdots \parr a_k, \Xi\\ \hfill \Lambda\end{bmatrix}\\
{\scriptstyle\multimap^\star_{\mathsf{R}}}\Big\uparrow\\
\begin{bmatrix}\Psi, G_1, \dots, G_{k''}\\ \hfill \Gamma\end{bmatrix} \vdash \begin{bmatrix} H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k, \Xi\\ \hfill \Lambda\end{bmatrix}\\
{\scriptstyle\Rightarrow^\star_{\mathsf{R}}}\Big\uparrow\\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix} G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k, \Xi\\ \hfill \Lambda\end{bmatrix}
\end{array}$$

Fig. 3. Clause reduction right inference rule δ_{R}.

3.1.4. Theorem (*Alternative proof to 3.1.3*). *Every formula in* FOLL *is equivalent to a clause.*

Proof. Structural induction on a Forum formula A equivalent to the given FOLL formula.
Basis cases: If A is an atom or $\bot$, just take A as the equivalent clause. If $A = \top$ take $(\top \multimap \bot) \multimap \bot$.
Inductive cases: Let δ and δ' be clauses equivalent to formulae B and B' in Forum, respectively.

- $A = B \parr B'$. Consider $\delta \parr \delta'$ and use the commutative property of $\parr$. Use $A \parr \bot \equiv A$ if necessary.
- $A = B \mathbin{\&} B'$. Take $(\delta \mathbin{\&} \delta' \multimap \bot) \multimap \bot$.
- $A = B \multimap B'$. Consider $\delta \multimap \delta' \equiv \delta^\bot \parr \delta'$ and use the commutative property of $\parr$.
- $A = B \Rightarrow B'$. Consider $\delta \Rightarrow \delta' \equiv (!\delta)^\bot \parr \delta'$ and use the commutative property of $\parr$.
- $A = \forall \vec{x}.B$. Take $(\forall \vec{x}.\delta \multimap \bot) \multimap \bot$. □

One should be aware, though, that there are some concerns in Miller's [8] about clauses (similar to ours) with degenerate head $\bot$. Clauses of that kind, when at the left of $\vdash$, are always available to rewritings, what could be cause of explosion of the search space of proofs. How to translate formulae into goals and clauses is then a matter of careful judgment, to be exercised on the concrete situations one should deal with.

3.2. Deriving in the right context

We start here an analysis of the behaviour of goals and clauses in Forum. Our purpose is to isolate big chunks of derivations, whose shape is forced by the combined constraints of Forum inference rules and the restrictions on syntax we imposed in the previous subsection. We consider these big derivations as instances of (big) inference rules, whose operational meaning is reminiscent of traditional logic programming, but, of course, more general.

The analysis we perform is very straightforward: it only requires careful inspection of the rules. One way of looking at what we do here is the following: linear logic is a system with many rules, each of which performs a little operational task. Here, we head towards a system with only two rules, each of which has a somewhat complex behaviour. The point is that this behaviour is *manageable* in two senses:

- it corresponds to a generalised view of logic programming, as intuitive as Miller's one in Forum;
- it is possible to define a cut elimination procedure for the two-rule formalism that, even when spelled out in full detail, as we do here, is comparable to its analogue in unconstrained linear logic.

In a subsequent paper, we will see how these rules make sense in a third way, which is being in good correspondence to a sensible concurrent operational semantics.

3.2.1. Definition. Let δ_{R} be the following *clause reduction right* inference rule, shown in Fig. 3 in terms of Forum rules:

$$\delta_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi, \mathsf{cp}(\delta)\\ \Gamma, \mathsf{lp}(\delta)\end{bmatrix} \vdash \begin{bmatrix}\hfill \Xi\\ \mathsf{hd}(\delta), \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\delta, \Xi\\ \hfill \Lambda\end{bmatrix}}\,.$$

In the figure $k > 0$ and $k', k'' \geqslant 0$. Starred inference rule names mean repeated application of the rule, or no application at all; ($\parr_{\mathsf{R}}$ or a) stands for ‘application of one of either $\parr_{\mathsf{R}}$ or a.’ In the special case, where $k = 0$ the upper sequence of ($\parr_{\mathsf{R}}$ or a) rules is replaced by a single application of $\bot_{\mathsf{R}}$.

Clauses assume different meanings depending on whether they appear at the left or at the right of the entailment symbol $\vdash$. When a clause appears at the right of $\vdash$, we operationally interpret it as follows:

- classical premises are added to the classical program: they can be used at will (or not used at all) in the rest of the computation;
- linear premises are added to the linear program: they must be used exactly once in the rest of the computation;
- atoms in the head go into the atomic context: they are added to the current multiset of resources upon which the program will act.

We informally say that a δ_{R} rule *loads* the contexts, by *reducing* clauses. δ_{R} is nothing more than a shortening for a piece of a derivation. The following proposition justifies its introduction.

3.2.2. Proposition. *Every proof of* $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\delta, \Xi\\ \Lambda\end{bmatrix}$ *has shape*

$$\begin{array}{c} \bigtriangledown \\ \begin{bmatrix}\Psi'\\ \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ \Lambda'\end{bmatrix} \\ {\scriptstyle \delta_{\mathsf{R}}}\uparrow \\ \begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\delta, \Xi\\ \Lambda\end{bmatrix} \end{array} .$$

Proof. By reasoning bottom-up, each application of an inference rule is compulsory. □

All Forum inference rules applied in δ_{R} are right ones. This is of course an aspect of the fact that Forum produces only uniform proofs (see [8–10]). Rules $\top_{\mathsf{R}}$, $\&_{\mathsf{R}}$ and $\forall_{\mathsf{R}}$ are still missing: they will appear in the reduction of goals.

We can build on δ_{R} an inference rule which reduces goals in the right linear context.

3.2.3. Definition. Let G_{R} be the following *goal reduction right* inference rule, shown in Fig. 4 in terms of δ_{R} and Forum rules:

$$\mathsf{G}_{\mathsf{R}}\,\frac{\begin{bmatrix}\Psi, \mathsf{cp}(\delta_1\rho)\\ \Gamma, \mathsf{lp}(\delta_1\rho)\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ \mathsf{hd}(\delta_1\rho), \Lambda\end{bmatrix} \quad \cdots \quad \begin{bmatrix}\Psi, \mathsf{cp}(\delta_h\rho)\\ \Gamma, \mathsf{lp}(\delta_h\rho)\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ \mathsf{hd}(\delta_h\rho), \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\forall\vec{x}.(\delta_1 \& \cdots \& \delta_h), \Xi\\ \Lambda\end{bmatrix}} ,$$

where $\vec{x}$ can be empty, ρ is an appropriate renaming substitution and $h \geqslant 0$. In the figure only one choice among the possible associations of $\&$ connectives has been considered, but every choice leads to the same multiset of premises.

This whole reduction phase is deterministic: in the end a goal is reduced to pieces with no choice about the possible outcome, except for the rather immaterial choice of eigenvariables in G_{R} rules. The Forum system has been designed to reduce choices to a minimum, in a bottom-up construction of a proof. Still, some ‘not necessary’ sequentialisation exists: in the case above it resides in the binary treatment of associative connectives. We can consider the G_{R} rule at the abstraction level in which all premises are reached at the same time in a parallel way, thus hiding the sequentialisation at the Forum’s level of abstraction. In other words, we can consider every instance of the G_{R} rule a representative of an equivalence class of derivations, differing only in the associations of $\&$ connectives.

We can perform on the G_{R} rule the same kind of simple reasoning we did for δ_{R} in Proposition 3.2.2.

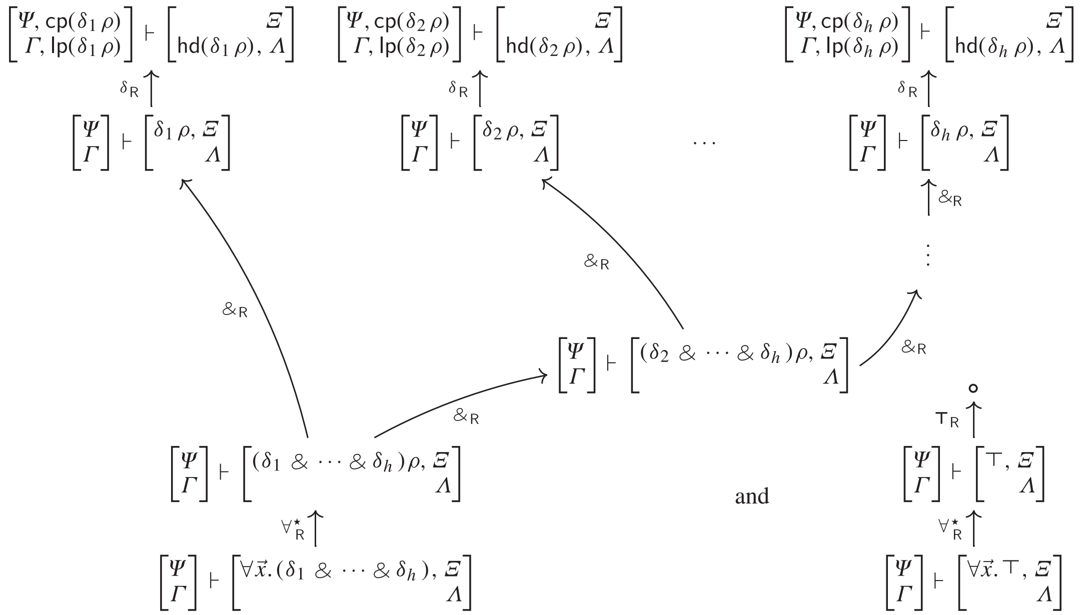
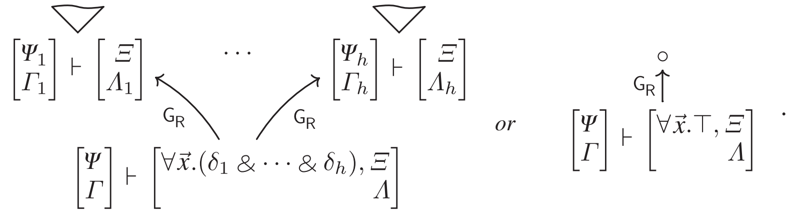

$\begin{bmatrix}\Psi, \mathsf{cp}(\delta_1\rho)\\ \Gamma, \mathsf{lp}(\delta_1\rho)\end{bmatrix} \vdash \begin{bmatrix} & \Xi\\ \mathsf{hd}(\delta_1\rho), & \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi, \mathsf{cp}(\delta_2\rho)\\ \Gamma, \mathsf{lp}(\delta_2\rho)\end{bmatrix} \vdash \begin{bmatrix} & \Xi\\ \mathsf{hd}(\delta_2\rho), & \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi, \mathsf{cp}(\delta_h\rho)\\ \Gamma, \mathsf{lp}(\delta_h\rho)\end{bmatrix} \vdash \begin{bmatrix} & \Xi\\ \mathsf{hd}(\delta_h\rho), & \Lambda\end{bmatrix}$
δ_R
δ_R
δ_R
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\delta_1\rho, \Xi & \\ & \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\delta_2\rho, \Xi & \\ & \Lambda\end{bmatrix}$
…
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\delta_h\rho, \Xi & \\ & \Lambda\end{bmatrix}$
$\&_\mathsf{R}$
$\&_\mathsf{R}$
⋮
$\&_\mathsf{R}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}(\delta_2 \mathbin{\&} \cdots \mathbin{\&} \delta_h)\rho, \Xi & \\ & \Lambda\end{bmatrix}$
$\&_\mathsf{R}$
$\&_\mathsf{R}$
$\circ$
$\top_\mathsf{R}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)\rho, \Xi & \\ & \Lambda\end{bmatrix}$
and
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\top, \Xi & \\ & \Lambda\end{bmatrix}$
$\forall^\star_\mathsf{R}$
$\forall^\star_\mathsf{R}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h), \Xi & \\ & \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\forall\vec{x}.\top, \Xi & \\ & \Lambda\end{bmatrix}$

Fig. 4. Goal reduction right inference rule $\mathsf{G_R}$ when $h > 0$ and $h = 0$.

3.2.4. Proposition. *Every proof of* $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G, \Xi & \\ & \Lambda\end{bmatrix}$ *has shape*


$\begin{bmatrix}\Psi_1\\ \Gamma_1\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ \Lambda_1\end{bmatrix}$
…
$\begin{bmatrix}\Psi_h\\ \Gamma_h\end{bmatrix} \vdash \begin{bmatrix}\Xi\\ \Lambda_h\end{bmatrix}$
$\mathsf{G_R}$
$\mathsf{G_R}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h), \Xi & \\ & \Lambda\end{bmatrix}$
or
$\circ$
$\mathsf{G_R}$
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\forall\vec{x}.\top, \Xi & \\ & \Lambda\end{bmatrix}$
.

Proof. By reasoning bottom-up, each application of an inference rule is compulsory. □

$\mathsf{G_R}$ defines the behaviour of goals when they appear at the right of $\vdash$. They generate as many branches in the computation as there are clauses in the conjunction. When $\mathsf{G_R}$ is applied to a $\forall\vec{x}.\top$, it just *terminates* a (thread of a) computation.

3.2.5. Definition. A *G-state sequent* is a state sequent of the kind $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}$, where all formulae in Ψ and Γ are goals.

By 3.1.2 and 3.2.4 we can always reduce provability of a Forum formula (therefore of a FOLL's one, by 2.11) to provability of some G-state sequents. Moreover, we can always reduce provability of a given formula to the provability of exactly one G-state sequent by employing the double negation equivalence $G \equiv (G \multimap \bot) \multimap \bot$: this last formula is a clause.

3.3. Deriving in the left context

Let us now turn our attention to left rules and the behaviour of goals and clauses when they appear at the left of $\vdash$, as left focused formulae.

G-state sequents embody a natural notion of state for our computations. To proceed computing from a G-state sequent, clauses from its program must be applied to its atomic context. Left rules come into play: application of clauses is mainly accomplished by $\multimap_\mathsf{L}$ and $\Rightarrow_\mathsf{L}$ rules. Rules $\parr_\mathsf{L}$ and $\bot_\mathsf{L}$ have a role in the applicability of clauses. Rule $\multimap_\mathsf{L}$ is also responsible for some non-deterministic choices about the splitting of state into multiple substates. Clauses are applicable to the atomic context Λ whenever their heads match a submultiset of Λ's atoms. Let us focus first on this matching aspect.

3.3.1. Definition. Let h be the following *head matching* inference rule, where $k \geqslant 0$:

$$\mathsf{h}\,\dfrac{}{\begin{bmatrix}\Psi\\ \ \end{bmatrix} a_1 \parr \cdots \parr a_k \vdash \begin{bmatrix} \ \\ a_1, \dots, a_k\end{bmatrix}}\ .$$

Fig. 5 shows how h corresponds to Forum inference rules. The same considerations made above about the associativity of $\&$ hold here for $\parr$.

3.3.2. Proposition. *If the sequent* $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} a_1 \parr \cdots \parr a_k \vdash \begin{bmatrix} \ \\ \Lambda\end{bmatrix}$ *is provable, then Γ is empty, $\Lambda = \{a_1, \dots, a_k\}_+$ and the only proof is*

$$\begin{array}{c} \circ \\ {}^{\mathsf{h}}\!\uparrow \\ \begin{bmatrix}\Psi\\ \ \end{bmatrix} a_1 \parr \cdots \parr a_k \vdash \begin{bmatrix} \ \\ a_1, \dots, a_k\end{bmatrix} \end{array}\ .$$

Proof. Consider Fig. 5: from the root to the leaves, all applications of inference rules are compelled by the left focused formula. Identity axioms force empty left linear contexts. By reading from the leaves to the root, $\parr_\mathsf{L}$ rules then constrain the conclusion. The case $k = 0$ is trivial. □

3.3.3. Definition. Let δ_L be the following *clause reduction left* inference rule, shown in Fig. 6 in terms of h and Forum rules:

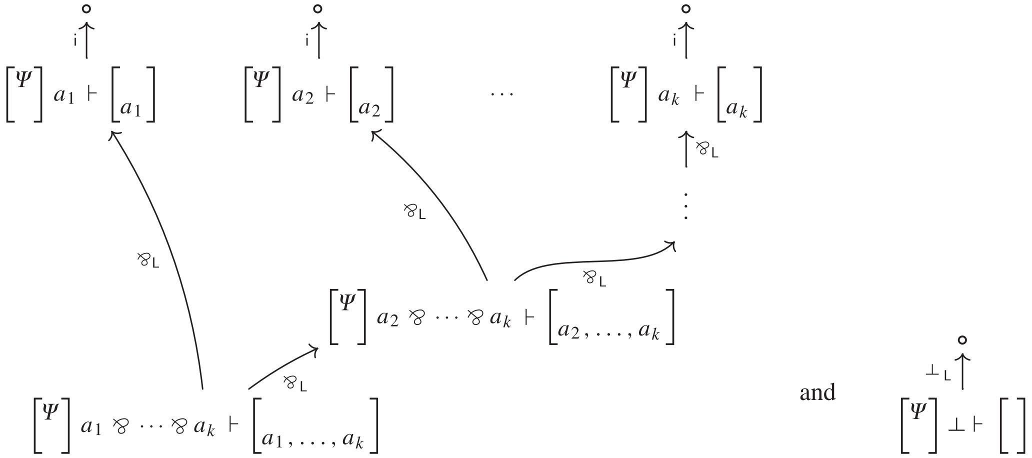
$$\delta_\mathsf{L}\,\dfrac{\begin{bmatrix}\Psi\\ \ \end{bmatrix} \vdash \begin{bmatrix}G_1\\ \ \end{bmatrix} \ \cdots\ \begin{bmatrix}\Psi\\ \ \end{bmatrix} \vdash \begin{bmatrix}G_{k''}\\ \ \end{bmatrix} \quad \begin{bmatrix}\Psi\\ \Gamma_1\end{bmatrix} \vdash \begin{bmatrix}H_1\\ \Lambda_1\end{bmatrix} \ \cdots\ \begin{bmatrix}\Psi\\ \Gamma_{k'}\end{bmatrix} \vdash \begin{bmatrix}H_{k'}\\ \Lambda_{k'}\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \delta \vdash \begin{bmatrix} \ \\ \Lambda\end{bmatrix}}\ ,$$

where $\delta = G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k$, and $k, k', k'' \geqslant 0$, and where $\Gamma_1 \uplus \cdots \uplus \Gamma_{k'} = \Gamma$ and $\Lambda_1 \uplus \cdots \uplus \Lambda_{k'} \uplus \{a_1, \dots, a_k\}_+ = \Lambda$.

Let us go through δ_L step by step; please note that it contains left rules only. This is a phase in a derivation in which a left focused clause is reduced and used in a rewriting.

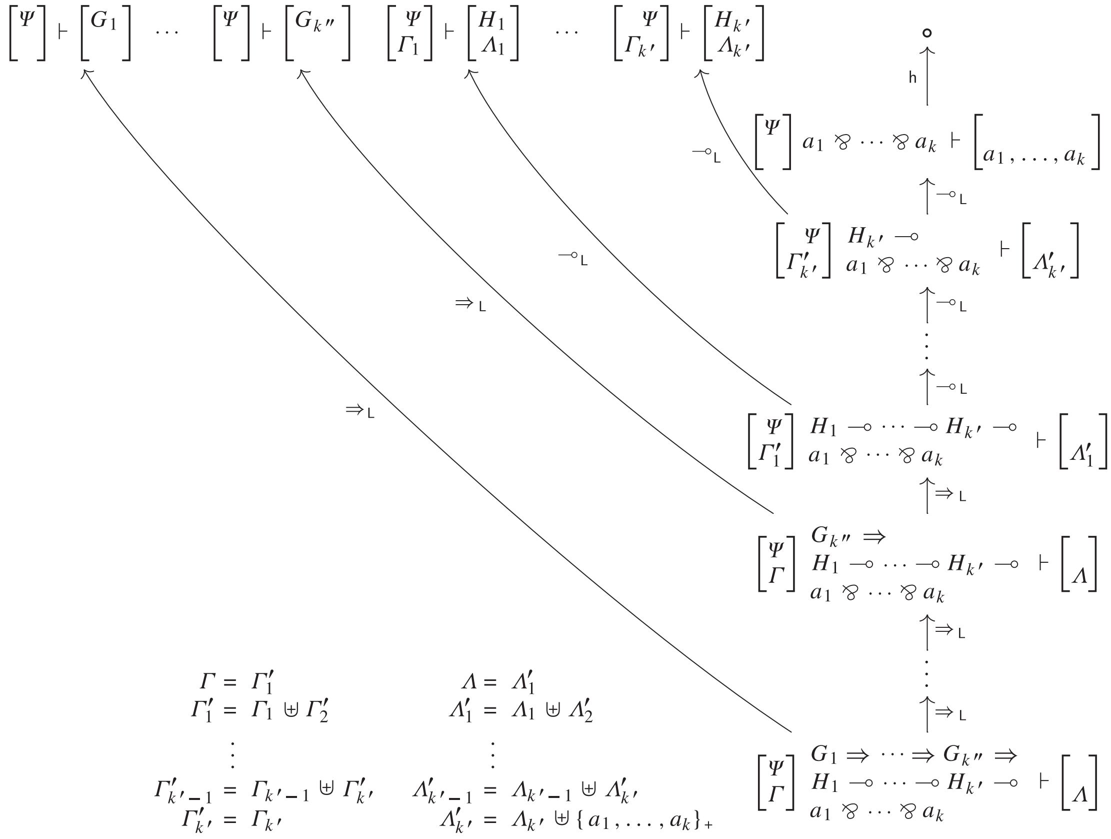
(1) All classical premises of δ are evaluated in classical context Ψ.
(2) All linear premises of δ are evaluated in classical context Ψ and in linear contexts which are non-deterministically obtained as indicated. Every linear premise gets a part of each piece of linear context: the left one is completely split among the premises; the atomic one is also split except for atoms which have to match the head of the selected clause; the right one is empty. In the figure some relations among contexts are noted for convenience.
(3) The head of δ is matched against the residual atomic context in an h rule.

As an outcome of the reduction of the left focused clause, we have a multiset of premises which will be further reduced by as many $\mathsf{G_R}$ rules. They, in turn, will produce G-state sequents. The most degenerate instances of δ_L have no premises. Special cases where there are no classical or linear premises are easily inferable from the general scheme



i
[Ψ] a1 ⊢ [a1]
[Ψ] a2 ⊢ [a2]
· · ·
[Ψ] ak ⊢ [ak]
⅋L
[Ψ] a2 ⅋ · · · ⅋ ak ⊢ [a2, . . . , ak]
[Ψ] a1 ⅋ · · · ⅋ ak ⊢ [a1, . . . , ak]
and
⊥L
[Ψ] ⊥ ⊢ [ ]

Fig. 5. Head matching inference rule h when $k > 0$ and $k = 0$.



[Ψ] ⊢ [G1] · · · [Ψ] ⊢ [Gk″]
[Ψ, Γ1] ⊢ [H1, Λ1] · · · [Ψ, Γk′] ⊢ [Hk′, Λk′]
h
[Ψ] a1 ⅋ · · · ⅋ ak ⊢ [a1, . . . , ak]
⊸L
[Ψ, Γ′k′] Hk′ ⊸ a1 ⅋ · · · ⅋ ak ⊢ [Λ′k′]
[Ψ, Γ′1] H1 ⊸ · · · ⊸ Hk′ ⊸ a1 ⅋ · · · ⅋ ak ⊢ [Λ′1]
⇒L
[Ψ, Γ] Gk″ ⇒ H1 ⊸ · · · ⊸ Hk′ ⊸ a1 ⅋ · · · ⅋ ak ⊢ [Λ]
[Ψ, Γ] G1 ⇒ · · · ⇒ Gk″ ⇒ H1 ⊸ · · · ⊸ Hk′ ⊸ a1 ⅋ · · · ⅋ ak ⊢ [Λ]
Γ = Γ′1   Λ = Λ′1
Γ′1 = Γ1 ⊎ Γ′2   Λ′1 = Λ1 ⊎ Λ′2
Γ′k′−1 = Γk′−1 ⊎ Γ′k′   Λ′k′−1 = Λk′−1 ⊎ Λ′k′
Γ′k′ = Γk′   Λ′k′ = Λk′ ⊎ {a1, . . . , ak}+

Fig. 6. Clause reduction left inference rule δ_{L}.

provided. Thanks to uniform provability, all non-determinism in searching for Forum proofs resides in left rules. Much of it can be concentrated into a decision rule, but one should notice that δ_{L} is also non-deterministic in the splitting of the linear contexts.

$$
\begin{array}{c}
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \delta_l\sigma \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle(\&_{\mathsf{LL}}\text{ or }\&_{\mathsf{LR}})^\star}\Big\uparrow\\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \delta_1\sigma \mathbin{\&} \dots \mathbin{\&} \delta_h\sigma \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle\forall^\star_{\mathsf{L}}}\Big\uparrow\\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} G \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle\mathsf{d}_{\mathsf{L}}}\Big\uparrow\\
\begin{bmatrix}\Psi\\ \Gamma, G\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}
\end{array}
\qquad\text{or}\qquad
\begin{array}{c}
\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix} \delta_l\sigma \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle(\&_{\mathsf{LL}}\text{ or }\&_{\mathsf{LR}})^\star}\Big\uparrow\\
\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix} \delta_1\sigma \mathbin{\&} \dots \mathbin{\&} \delta_h\sigma \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle\forall^\star_{\mathsf{L}}}\Big\uparrow\\
\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix} G \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle\mathsf{d}_{\mathsf{C}}}\Big\uparrow\\
\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}
\end{array}
$$

Fig. 7. Decision inference rule d in its two possibilities.

What we see here is a very similar situation to Andreoli's focusing notion [1]. Goals can be considered generalised connectives. As usual, connectives are defined by their inference rules, a right one and a left one. Following Andreoli, we could consider a goal on the right *asynchronous* and a goal on the left *synchronous*. While building a proof bottom-up, asynchronous connectives need no backtracking, while synchronous ones do, since a decision rule is involved at a finer level of abstraction. This, of course, is no coincidence, given that this entire investigation ultimately stems from Andreoli's work.

3.3.4. Definition. Let d be the *decision* inference rule, defined by the following two (non-mutually exclusive) cases, and shown in Fig. 7 in terms of Forum rules:

$$
\mathsf{d}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \delta_l\sigma \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma, G\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}}
\quad\text{or}\quad
\mathsf{d}\,\frac{\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix} \delta_l\sigma \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}}\,,
$$

where the conclusions are G-state sequents. In the first case Γ, G is the *selected context*, in the second it is Ψ, G. Goal $G = \forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)$, where $h > 0$ and $\vec{x}$ can be empty, is the *selected goal*; $\delta_l\sigma$ is the *selected clause*, $1 \leqslant l \leqslant h$, and σ is a substitution whose domain is $\vec{x}$.

3.3.5. Proposition. *All proofs of* $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}$ *have shape*

$$
\begin{array}{c}
\bigtriangledown\\
\begin{bmatrix}\Psi\\ \Gamma'\end{bmatrix} \delta \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}\\
{\scriptstyle\mathsf{d}}\Big\uparrow\\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}
\end{array}\qquad,
$$

for some Γ' and the inference rule above d *is* δ_{L}.

Proof. By reasoning bottom-up, each application of an inference rule is compulsory. □

3.3.6. Remark. There are no proofs for the sequent $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \forall\vec{x}.\top \vdash \begin{bmatrix}\\ \Lambda\end{bmatrix}$.

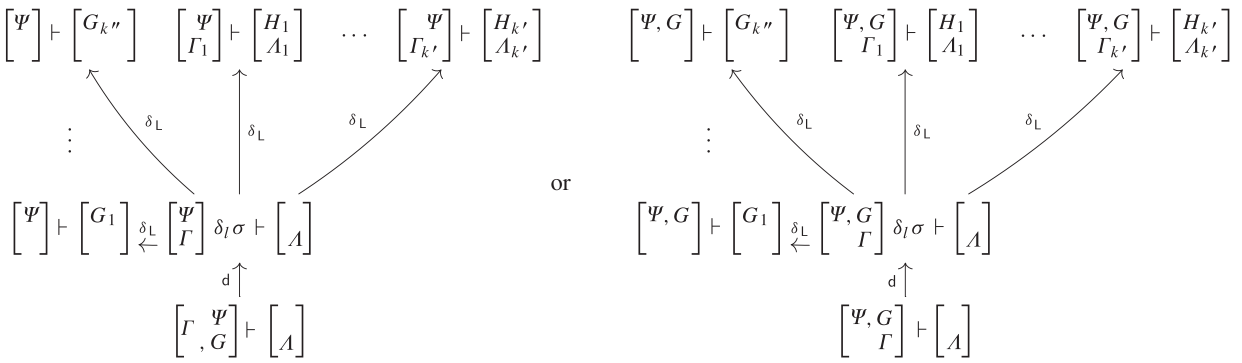


Fig. 8. Goal reduction left inference rule $\mathsf{G_L}$ in its two possibilities.

Each application of d involves the following choices:

(1) which goal $G = \forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)$ to select, and in which context; it could be the case that a certain G appears both in the classical and in the linear program;

(2) which substitution σ to apply;

(3) which clause $\delta_l\sigma$ to select among $\delta_1\sigma \mathbin{\&} \cdots \mathbin{\&} \delta_h\sigma$.

We can build on d and δ_L an inference rule which reduces goals in the program.

3.3.7. Definition. Let $\mathsf{G_L}$ be the *goal reduction left* inference rule, defined in the following two (non-mutually exclusive) cases and shown in Fig. 8 in terms of d and δ_L rules:

$$\mathsf{G_L}\ \frac{\begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G_1\\ \ \end{bmatrix}\ \cdots\ \begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G_{k''}\\ \ \end{bmatrix}\quad\begin{bmatrix}\Psi\\ \Gamma_1\end{bmatrix}\vdash\begin{bmatrix}H_1\\ \Lambda_1\end{bmatrix}\ \cdots\ \begin{bmatrix}\Psi\\ \Gamma_{k'}\end{bmatrix}\vdash\begin{bmatrix}H_{k'}\\ \Lambda_{k'}\end{bmatrix}}{\begin{bmatrix}\Psi\\ \Gamma, G\end{bmatrix}\vdash\begin{bmatrix}\ \\ \Lambda\end{bmatrix}}$$

or

$$\mathsf{G_L}\ \frac{\begin{bmatrix}\Psi, G\\ \ \end{bmatrix}\vdash\begin{bmatrix}G_1\\ \ \end{bmatrix}\ \cdots\ \begin{bmatrix}\Psi, G\\ \ \end{bmatrix}\vdash\begin{bmatrix}G_{k''}\\ \ \end{bmatrix}\quad\begin{bmatrix}\Psi, G\\ \Gamma_1\end{bmatrix}\vdash\begin{bmatrix}H_1\\ \Lambda_1\end{bmatrix}\ \cdots\ \begin{bmatrix}\Psi, G\\ \Gamma_{k'}\end{bmatrix}\vdash\begin{bmatrix}H_{k'}\\ \Lambda_{k'}\end{bmatrix}}{\begin{bmatrix}\Psi, G\\ \Gamma\end{bmatrix}\vdash\begin{bmatrix}\ \\ \Lambda\end{bmatrix}},$$

where $G = \forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)$, $\vec{x}$ can be empty, $\delta_l\sigma = G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k$, for $1 \leqslant l \leqslant h$ and $k, k', k'' \geqslant 0$, and where $\Gamma_1 \uplus \cdots \uplus \Gamma_{k'} = \Gamma$ and $\Lambda_1 \uplus \cdots \uplus \Lambda_{k'} \uplus \{a_1, \ldots, a_k\}_+ = \Lambda$.

3.4. The system

In the previous section, we built two big inference rules, a left one and a right one. Their definition is straightforward once one knows the (operational) meaning of linear logic connectives. One should notice that if the language of formulae were not restricted to goals exactly the way we did, such an enterprise would really be cumbersome and complex, and most probably pointless. Our point is, instead, that goals in Forum actually are made by a sort of generalised connective, in the same sense as a connective is defined in the sequent calculus by a left and a right rule. The harmony of the connective definition is then tested by a cut elimination procedure.

3.4.1. Definition. Let G-Forum be the formal system whose sequents are G-state sequents or sequents of the form $\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix}\vdash\begin{bmatrix}G\\ \Lambda\end{bmatrix}$, where Ψ and Γ contain goals, and whose inference rules are $\mathsf{G_L}$ and $\mathsf{G_R}$.

An important technical feature of a sequent system is that the rules allow for a cut elimination theorem. We know already cut elimination from linear logic, but that theorem is proved inside the sequent system where all the usual connectives are defined. Even if we know that cut is admissible in our system (since the system is complete), it is still necessary to prove that cuts can be constructively eliminated by a procedure that operates inside the system, i.e., for example, without recurring to inference rules at a different abstraction level. We all know that cut elimination is the principal combinatorial property of any deductive system, and this fundamental result usually paves the way to other, derived properties, like Herbrand-like theorems and interpolation theorems.

4. Cut elimination

Let us firstly define two natural cut rules for G-Forum.

4.1. Definition. The following inference rules $\bowtie_{\mathsf{L}}$ and $\bowtie_{\mathsf{C}}$ are, respectively, called *linear cut* and *classical cut*:

$$\bowtie_{\mathsf{L}}\,\frac{\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix}\vdash\begin{bmatrix}G\\ \Lambda\end{bmatrix}\quad\begin{bmatrix}\Psi'\\ G,\Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}}{\begin{bmatrix}\Psi,\Psi'\\ \Gamma,\Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda,\Lambda'\end{bmatrix}}\quad\text{and}\quad\bowtie_{\mathsf{C}}\,\frac{\begin{bmatrix}\Psi\\ \end{bmatrix}\vdash\begin{bmatrix}G\\ \end{bmatrix}\quad\begin{bmatrix}G,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}}{\begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}}\,.$$

Ξ' is either empty or a singleton. In both rules G is called the *principal formula*. System G-Forum$^{\bowtie_{\mathsf{L}},\bowtie_{\mathsf{C}}}$ is G-Forum where $\bowtie_{\mathsf{L}}$, $\bowtie_{\mathsf{C}}$ are allowed in proofs.

In the sequel, we prove that every proof in G-Forum$^{\bowtie_{\mathsf{L}},\bowtie_{\mathsf{C}}}$ can be transformed into an equivalent proof in G-Forum. The proof of the cut elimination theorem follows in part a traditional argument in which one deals with contraction by a generalised cut rule that cuts on several copies of the same principal formula (see for example [4]). We use the classical cut rule, in a certain generalisation $\bowtie'_{\mathsf{C}}$ together with a contraction rule $>$, to make G-Forum$^{\bowtie_{\mathsf{L}},\bowtie_{\mathsf{C}}}$ more general, and then we prove cut elimination on this more general system. The core of the proof is the elimination of the $\bowtie_{\mathsf{L}}$ rule. Actually, the design of the rules G_{L} and G_{R}, and the crucial decisions about the exact definition of goals, all come from a careful analysis of what is needed in this part of the cut elimination argument.

There are some original aspects (to the best of our knowledge) in the proof we offer below:

- all classical cuts are eliminated before linear ones,
- all contractions are eliminated at the end of the process.

The elimination of classical cuts essentially happens by transforming them into linear cuts. The two phases of elimination of classical cuts and contractions are best interpreted as bookkeeping phases around the central phase of elimination of linear cuts. We believe this technique might prove general and useful in similar situations.

We first introduce a contraction rule and a classical cut rule in a more general form as follows.

4.2. Definition. The following inference rule $>$ is called *contraction*:

$$>\,\frac{\begin{bmatrix}\Psi,G,G\\ \Gamma\end{bmatrix}\vdash\begin{bmatrix}\Xi\\ \Lambda\end{bmatrix}}{\begin{bmatrix}\Psi,G\\ \Gamma\end{bmatrix}\vdash\begin{bmatrix}\Xi\\ \Lambda\end{bmatrix}}\,.$$

G-Forum$^{>,\bowtie_{\mathsf{L}},\bowtie_{\mathsf{C}}}$, G-Forum$^{>,\bowtie_{\mathsf{L}}}$ and G-Forum$^{>}$ stand for G-Forum where, in addition to G_{L} and G_{R}, rules in the superscript are allowed.

4.3. Definition. We write lG to indicate the multiset $\{G,\dots,G\}_+$, where G appears l times; analogously, given the multiset of formulae Ψ, we write $l\Psi$ to indicate $\underbrace{\Psi\uplus\cdots\uplus\Psi}_{l\text{ times}}$; when $l=0$ both lG and $l\Psi$ stand for $\emptyset_+$.

4.4. Definition. The *generalised classical cut* rule $\bowtie'_{\mathsf{C}}$ is so defined:

$$\bowtie'_{\mathsf{C}}\ \frac{\begin{bmatrix}\Psi\\ \end{bmatrix} \vdash \begin{bmatrix}G\\ \end{bmatrix} \qquad \begin{bmatrix} lG,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}}{\begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}},$$

where $l \geqslant 0$. Again, G is called the *principal formula*.

We will prove that proofs in G-Forum$^{>,\bowtie_{\mathsf{L}},\bowtie'_{\mathsf{C}}}$ can be transformed into proofs in G-Forum$^{>,\bowtie_{\mathsf{L}}}$, which in turn can be transformed into proofs in G-Forum$^{>}$, which can be transformed into proofs in G-Forum, where every transformation preserves the conclusion:

$$\text{G-Forum}^{>,\bowtie_{\mathsf{L}},\bowtie'_{\mathsf{C}}} \to \text{G-Forum}^{>,\bowtie_{\mathsf{L}}} \to \text{G-Forum}^{>} \to \text{G-Forum}.$$

The first step and the third one are technically similar and eminently ‘structural’ in nature, meaning that they deal with bookkeeping of resources. The middle step is a typical cut elimination theorem which tests the symmetries of the system.

The cut elimination theorem relies on an induction measure that we call *cut-rank*, which is essentially a measure of the syntactical complexity of the principal formula.

4.5. Definition. The *cut-rank* $\mathsf{cr}(G)$ of goal G is a natural number inductively defined as follows:

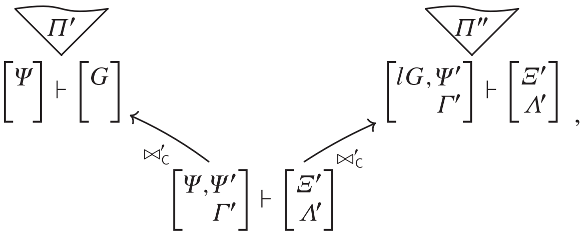
$$\begin{aligned}
&\mathsf{cr}'(\forall\vec{x}.\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h) = \max\{\mathsf{cr}'(\delta_1), \ldots, \mathsf{cr}'(\delta_h), 0\} + 1 \quad \text{where } h \geqslant 0,\\
&\mathsf{cr}'(G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \mathbin{\invamp} \cdots \mathbin{\invamp} a_k)\\
&\quad = \max\{\mathsf{cr}(G_1), \ldots, \mathsf{cr}(G_{k''}), \mathsf{cr}(H_1), \ldots, \mathsf{cr}(H_{k'}), 0\} + 1 \quad \text{where } k, k', k'' \geqslant 0.
\end{aligned}$$

The *cut-rank* of an instance of $\bowtie_{\mathsf{L}}$, $\bowtie_{\mathsf{C}}$ or $\bowtie'_{\mathsf{C}}$ rule is the cut-rank of its principal formula. The cut-rank $\mathsf{cr}(\Pi)$ of a proof Π is the maximum cut-rank of instances of $\bowtie_{\mathsf{L}}$, $\bowtie_{\mathsf{C}}$ and $\bowtie'_{\mathsf{C}}$ in Π; if Π is cut-free then we define $\mathsf{cr}(\Pi) = 0$.

4.1. Elimination of the generalised classical cut rule

To prove that $\bowtie'_{\mathsf{C}}$ occurrences may be eliminated from a proof, we prove that every instance of $\bowtie'_{\mathsf{C}}$ can be moved upwards in the proof, along the branch at the right in the rule, until it disappears.

4.1.1. Lemma. *In* G-Forum$^{>,\bowtie_{\mathsf{L}},\bowtie'_{\mathsf{C}}}$, *let* Π *be the proof*


Π'
$\begin{bmatrix}\Psi\\ \end{bmatrix} \vdash \begin{bmatrix}G\\ \end{bmatrix}$
Π''
$\begin{bmatrix} lG,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}$,
$\bowtie'_{\mathsf{C}}$
$\bowtie'_{\mathsf{C}}$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix}$

where $l \geqslant 0$ *and no instance of* $\bowtie'_{\mathsf{C}}$ *appears in* Π' *and* Π''. *Then, there exists a proof* $\overline{\Pi}$ *in* G-Forum$^{>,\bowtie_{\mathsf{L}}}$, *whose conclusion is the same as that of* Π, *and* $\mathsf{cr}(\overline{\Pi}) \leqslant \mathsf{cr}(\Pi)$.

Proof. We will prove the lemma by induction on the depth of Π''.

Basis cases: The following two cases are the only possible ones:

(1) $\mathsf{G_L}$. If

$$\Pi = \begin{tikzcd}
\Pi' \arrow[d, no head] & & \circ \\
\begin{bmatrix}\Psi \\ \ \end{bmatrix} \vdash \begin{bmatrix} G \\ \ \end{bmatrix} & & \begin{bmatrix} lG,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \ \\ \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G_L}"] \\
& \begin{bmatrix}\Psi,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \ \\ \Lambda' \end{bmatrix} \arrow[ul, "\bowtie'_{\mathsf{C}}"] \arrow[ur, "\bowtie'_{\mathsf{C}}"'] &
\end{tikzcd},$$

then two cases are possible. If G is not the selected goal in the $\mathsf{G_L}$ instance, or it is selected and either $G \Subset \Psi'$ or $G \Subset \Gamma'$, then take:

$$\overline{\Pi} = \begin{tikzcd}
\circ \\
\begin{bmatrix}\Psi,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \ \\ \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G_L}"]
\end{tikzcd}.$$

Otherwise, if G is the selected goal in the $\mathsf{G_L}$ instance and G does not appear neither in Ψ' nor in Γ', it must be $l > 0$ and $\Gamma' = \emptyset_+$; in this case take

$$\overline{\Pi} = \begin{tikzcd}
\Pi' \arrow[d, no head] & & \circ \\
\begin{bmatrix}\Psi \\ \ \end{bmatrix} \vdash \begin{bmatrix} G \\ \ \end{bmatrix} & & \begin{bmatrix} \Psi' \\ G \end{bmatrix} \vdash \begin{bmatrix} \ \\ \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G_L}"] \\
& \begin{bmatrix}\Psi,\Psi' \\ \ \end{bmatrix} \vdash \begin{bmatrix} \ \\ \Lambda' \end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd}.$$

(2) $\mathsf{G_R}$. If

$$\overline{\Pi} = \begin{tikzcd}
\Pi' \arrow[d, no head] & & \circ \\
\begin{bmatrix}\Psi \\ \ \end{bmatrix} \vdash \begin{bmatrix} G \\ \ \end{bmatrix} & & \begin{bmatrix} lG,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G_R}"] \\
& \begin{bmatrix}\Psi,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda' \end{bmatrix} \arrow[ul, "\bowtie'_{\mathsf{C}}"] \arrow[ur, "\bowtie'_{\mathsf{C}}"'] &
\end{tikzcd},$$

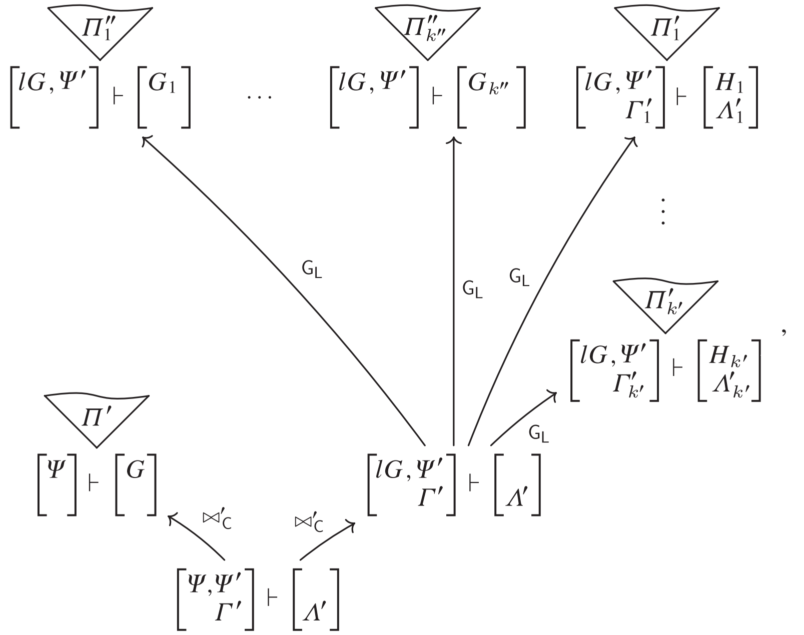
then take :

$$\overline{\Pi} = \begin{tikzcd}
\circ \\
\begin{bmatrix}\Psi,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G_R}"]
\end{tikzcd}.$$

In all cases it holds $\mathsf{cr}(\overline{\Pi}) \leqslant \mathsf{cr}(\Pi)$.

Inductive cases:

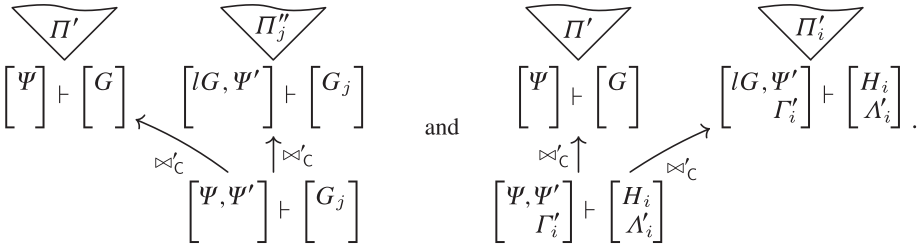
(3) $\mathsf{G_L}$. Let Π be



Π''_1
$\begin{bmatrix} lG,\Psi' \end{bmatrix} \vdash \begin{bmatrix} G_1 \end{bmatrix}$
$\dots$
$\Pi''_{k''}$
$\begin{bmatrix} lG,\Psi' \end{bmatrix} \vdash \begin{bmatrix} G_{k''} \end{bmatrix}$
Π'_1
$\begin{bmatrix} lG,\Psi' \\ \Gamma'_1 \end{bmatrix} \vdash \begin{bmatrix} H_1 \\ \Lambda'_1 \end{bmatrix}$
$\vdots$
$\Pi'_{k'}$
$\begin{bmatrix} lG,\Psi' \\ \Gamma'_{k'} \end{bmatrix} \vdash \begin{bmatrix} H_{k'} \\ \Lambda'_{k'} \end{bmatrix}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
Π'
$\begin{bmatrix} \Psi \end{bmatrix} \vdash \begin{bmatrix} G \end{bmatrix}$
$\begin{bmatrix} lG,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda' \end{bmatrix}$
$\bowtie'_\mathsf{C}$
$\bowtie'_\mathsf{C}$
$\begin{bmatrix} \Psi,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda' \end{bmatrix}$

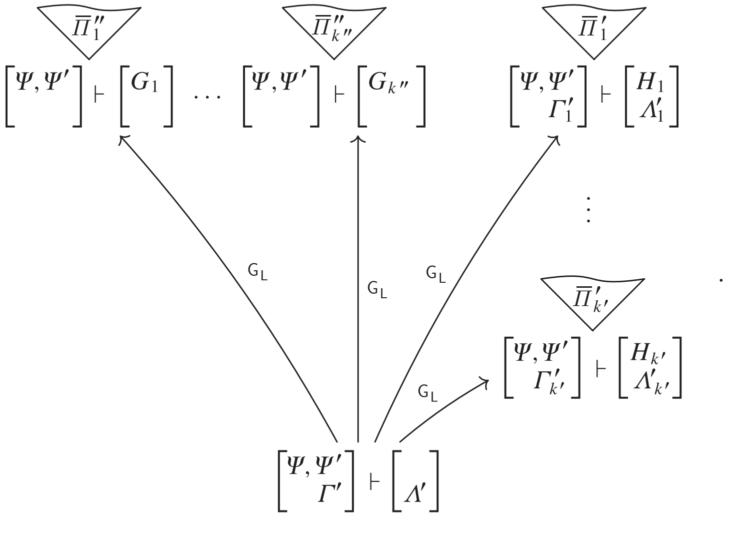
,

where $k' + k'' > 0$. By the induction hypothesis, for $1 \leqslant j \leqslant k''$ and $1 \leqslant i \leqslant k'$, there are $\bowtie'_\mathsf{C}$-free proofs $\overline{\Pi}''_j$ and $\overline{\Pi}'_i$, corresponding, respectively, to



Π'
$\begin{bmatrix} \Psi \end{bmatrix} \vdash \begin{bmatrix} G \end{bmatrix}$
Π''_j
$\begin{bmatrix} lG,\Psi' \end{bmatrix} \vdash \begin{bmatrix} G_j \end{bmatrix}$
$\bowtie'_\mathsf{C}$
$\bowtie'_\mathsf{C}$
$\begin{bmatrix} \Psi,\Psi' \end{bmatrix} \vdash \begin{bmatrix} G_j \end{bmatrix}$
and
Π'
$\begin{bmatrix} \Psi \end{bmatrix} \vdash \begin{bmatrix} G \end{bmatrix}$
Π'_i
$\begin{bmatrix} lG,\Psi' \\ \Gamma'_i \end{bmatrix} \vdash \begin{bmatrix} H_i \\ \Lambda'_i \end{bmatrix}$
$\bowtie'_\mathsf{C}$
$\bowtie'_\mathsf{C}$
$\begin{bmatrix} \Psi,\Psi' \\ \Gamma'_i \end{bmatrix} \vdash \begin{bmatrix} H_i \\ \Lambda'_i \end{bmatrix}$
.

Two cases are possible. If G is not the selected goal in the $\mathsf{G_L}$ instance, or if it is selected and either $G \Subset \Psi'$ or $G \Subset \Gamma'$, then take $\overline{\Pi}$ as:



$\overline{\Pi}''_1$
$\begin{bmatrix} \Psi,\Psi' \end{bmatrix} \vdash \begin{bmatrix} G_1 \end{bmatrix}$
$\dots$
$\overline{\Pi}''_{k''}$
$\begin{bmatrix} \Psi,\Psi' \end{bmatrix} \vdash \begin{bmatrix} G_{k''} \end{bmatrix}$
$\overline{\Pi}'_1$
$\begin{bmatrix} \Psi,\Psi' \\ \Gamma'_1 \end{bmatrix} \vdash \begin{bmatrix} H_1 \\ \Lambda'_1 \end{bmatrix}$
$\vdots$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\overline{\Pi}'_{k'}$
$\begin{bmatrix} \Psi,\Psi' \\ \Gamma'_{k'} \end{bmatrix} \vdash \begin{bmatrix} H_{k'} \\ \Lambda'_{k'} \end{bmatrix}$
$\mathsf{G_L}$
$\begin{bmatrix} \Psi,\Psi' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda' \end{bmatrix}$
.

Otherwise, if G is selected and G does not appear neither in Ψ' nor in Γ', it must be $l > 0$; in this case take $\overline{\Pi}$ as

$$\begin{tikzcd}
\begin{matrix}\overline{\Pi}''_1\\ \begin{bmatrix}\Psi,\Psi'\\ \ \end{bmatrix}\vdash\begin{bmatrix}G_1\\ \ \end{bmatrix}\end{matrix} & \cdots & \begin{matrix}\overline{\Pi}''_{k''}\\ \begin{bmatrix}\Psi,\Psi'\\ \ \end{bmatrix}\vdash\begin{bmatrix}G_{k''}\\ \ \end{bmatrix}\end{matrix} & \begin{matrix}\overline{\Pi}'_1\\ \begin{bmatrix}\Psi,\Psi'\\ \Gamma'_1\end{bmatrix}\vdash\begin{bmatrix}H_1\\ \Lambda'_1\end{bmatrix}\end{matrix} \\
 & & & \vdots \\
 & & & \begin{matrix}\overline{\Pi}'_{k'}\\ \begin{bmatrix}\Psi,\Psi'\\ \Gamma'_{k'}\end{bmatrix}\vdash\begin{bmatrix}H_{k'}\\ \Lambda'_{k'}\end{bmatrix}\end{matrix} \quad . \\
\begin{matrix}\Pi'\\ \begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G\\ \ \end{bmatrix}\end{matrix} & & \begin{bmatrix}\Psi,\Psi'\\ G,\Gamma'\end{bmatrix}\vdash\begin{bmatrix}\ \\ \Lambda'\end{bmatrix} \arrow[uuull, "\mathsf{G_L}"'] \arrow[uuu, "\mathsf{G_L}"] \arrow[uuur, "\mathsf{G_L}"] \arrow[ur, "\mathsf{G_L}"'] & \\
 & \begin{bmatrix}\Psi,\Psi,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\ \\ \Lambda'\end{bmatrix} \arrow[ul, "\bowtie_\mathsf{L}"] \arrow[ur, "\bowtie_\mathsf{L}"] & & \\
 & \begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\ \\ \Lambda'\end{bmatrix} \arrow[u, ">^\star"'] & &
\end{tikzcd}$$

(4) $\mathsf{G_R}$. Let Π be

$$\begin{tikzcd}
\begin{matrix}\Pi'\\ \begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G\\ \ \end{bmatrix}\end{matrix} & \begin{matrix}\Pi_1\\ \begin{bmatrix}lG,\Psi',\mathsf{cp}(\delta_1\rho)\\ \Gamma',\mathsf{lp}(\delta_1\rho)\end{bmatrix}\vdash\begin{bmatrix}\ \\ \mathsf{hd}(\delta_1\rho),\Lambda'\end{bmatrix}\end{matrix} & \cdots & \begin{matrix}\Pi_h\\ \begin{bmatrix}lG,\Psi',\mathsf{cp}(\delta_h\rho)\\ \Gamma',\mathsf{lp}(\delta_h\rho)\end{bmatrix}\vdash\begin{bmatrix}\ \\ \mathsf{hd}(\delta_h\rho),\Lambda'\end{bmatrix}\end{matrix} \\
 & & \begin{bmatrix}lG,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}G'\\ \Lambda'\end{bmatrix} \arrow[ul, "\mathsf{G_R}"] \arrow[ur, "\mathsf{G_R}"'] & \quad , \\
 & & \begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}G'\\ \Lambda'\end{bmatrix} \arrow[u, "\bowtie'_\mathsf{C}"'] \arrow[uull, "\bowtie'_\mathsf{C}"] &
\end{tikzcd}$$

where $G' = \forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)$, for $h > 0$, and ρ is a renaming substitution over domain $\vec{x}$; we can assume that no variable in the range of ρ is free in Ψ. By the induction hypothesis, for $1 \leqslant i \leqslant h$, there are $\bowtie'_\mathsf{C}$-free proofs $\overline{\Pi}_i$, corresponding to:

$$\begin{tikzcd}
\begin{matrix}\Pi'\\ \begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G\\ \ \end{bmatrix}\end{matrix} & \begin{matrix}\Pi_i\\ \begin{bmatrix}lG,\Psi',\mathsf{cp}(\delta_i\rho)\\ \Gamma',\mathsf{lp}(\delta_i\rho)\end{bmatrix}\vdash\begin{bmatrix}\ \\ \mathsf{hd}(\delta_i\rho),\Lambda'\end{bmatrix}\end{matrix} & \quad . \\
 & \begin{bmatrix}\Psi,\Psi',\mathsf{cp}(\delta_i\rho)\\ \Gamma',\mathsf{lp}(\delta_i\rho)\end{bmatrix}\vdash\begin{bmatrix}\ \\ \mathsf{hd}(\delta_i\rho),\Lambda'\end{bmatrix} \arrow[ul, "\bowtie'_\mathsf{C}"] \arrow[u, "\bowtie'_\mathsf{C}"'] &
\end{tikzcd}$$

Take

$$\overline{\Pi} = \begin{tikzcd}[column sep=small]
\overline{\Pi}_1 & & \overline{\Pi}_h \\
\begin{bmatrix}\Psi,\Psi',\mathsf{cp}(\delta_1\,\rho)\\ \Gamma',\mathsf{lp}(\delta_1\,\rho)\end{bmatrix} \vdash \begin{bmatrix} \\ \mathsf{hd}(\delta_1\,\rho),\,\Lambda'\end{bmatrix} & \cdots & \begin{bmatrix}\Psi,\Psi',\mathsf{cp}(\delta_h\,\rho)\\ \Gamma',\mathsf{lp}(\delta_h\,\rho)\end{bmatrix} \vdash \begin{bmatrix} \\ \mathsf{hd}(\delta_h\,\rho),\,\Lambda'\end{bmatrix} \\
& & \begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda'\end{bmatrix} \arrow[ull, "\mathsf{G_R}"'] \arrow[u, "\mathsf{G_R}"']
\end{tikzcd} .$$

(5) $\bowtie_{\mathsf{L}}$. Let Π be

$$\begin{tikzcd}[column sep=small]
\Pi' & \Pi_1 & \Pi_h \\
\begin{bmatrix}\Psi\\ {}\end{bmatrix} \vdash \begin{bmatrix}G\\ {}\end{bmatrix} & \begin{bmatrix}l_1\,G,\Psi_1'\\ \Gamma_1'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda_1'\end{bmatrix} & \begin{bmatrix}l_2\,G,\Psi_2'\\ G',\Gamma_2'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda_2'\end{bmatrix} \\
& \begin{bmatrix}(l_1+l_2)G,\Psi_1',\Psi_2'\\ \Gamma_1',\Gamma_2'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda_1',\Lambda_2'\end{bmatrix} \arrow[u, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] & \\
& \begin{bmatrix}\Psi,\Psi_1',\Psi_2'\\ \Gamma_1',\Gamma_2'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda_1',\Lambda_2'\end{bmatrix} \arrow[uul, "\bowtie_{\mathsf{C}}'"] \arrow[u, "\bowtie_{\mathsf{C}}'"'] &
\end{tikzcd} ,$$

where $l_1, l_2 \geqslant 0$. By the induction hypothesis, consider $\bowtie_{\mathsf{C}}'$-free proofs $\overline{\Pi}_1$ and $\overline{\Pi}_2$ corresponding, respectively, to

$$\begin{tikzcd}[column sep=small]
\Pi' & & \Pi_1 \\
\begin{bmatrix}\Psi\\ {}\end{bmatrix} \vdash \begin{bmatrix}G\\ {}\end{bmatrix} & & \begin{bmatrix}l_1\,G,\Psi_1'\\ \Gamma_1'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda_1'\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi_1'\\ \Gamma_1'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda_1'\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{C}}'"] \arrow[ur, "\bowtie_{\mathsf{C}}'"'] &
\end{tikzcd}$$

and

$$\begin{tikzcd}[column sep=small]
\Pi' & & \Pi_2 \\
\begin{bmatrix}\Psi\\ {}\end{bmatrix} \vdash \begin{bmatrix}G\\ {}\end{bmatrix} & & \begin{bmatrix}l_2\,G,\Psi_2'\\ G',\Gamma_2'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda_2'\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi_2'\\ G',\Gamma_2'\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda_2'\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{C}}'"] \arrow[ur, "\bowtie_{\mathsf{C}}'"'] &
\end{tikzcd} .$$

Take

$$\overline{\Pi} = \begin{tikzcd}
\overline{\Pi}_1 & & \overline{\Pi}_2 \\
\begin{bmatrix}\Psi,\Psi'_1\\ \Gamma'_1\end{bmatrix}\vdash\begin{bmatrix}G'\\ \Lambda'_1\end{bmatrix} & & \begin{bmatrix}\Psi,\Psi'_2\\ G',\Gamma'_2\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'_2\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi,\Psi'_1,\Psi'_2\\ \Gamma'_1,\Gamma'_2\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[ul, bend right, "\bowtie_{\mathsf{L}}"'] \arrow[ur, bend left, "\bowtie_{\mathsf{L}}"] & \\
& \begin{bmatrix}\Psi,\Psi'_1,\Psi'_2\\ \Gamma'_1,\Gamma'_2\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[u, ">^{\star}"'] &
\end{tikzcd} .$$

(6) $>$. There are two distinct cases. If Π is

$$\begin{tikzcd}
& \Pi'' \\
\Pi' & \begin{bmatrix}(l+1)G,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \\
\begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G\\ \ \end{bmatrix} & \begin{bmatrix}lG,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \arrow[u, ">"] \\
& \begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \arrow[ul, bend right, "\bowtie'_{\mathsf{C}}"'] \arrow[u, "\bowtie'_{\mathsf{C}}"']
\end{tikzcd} ,$$

where $l > 0$, then, by induction hypothesis, there is a $\bowtie'_{\mathsf{C}}$-free proof $\overline{\Pi}$ corresponding to

$$\begin{tikzcd}
\Pi' & & \Pi'' \\
\begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G\\ \ \end{bmatrix} & & \begin{bmatrix}(l+1)G,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi'\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \arrow[ul, bend right, "\bowtie'_{\mathsf{C}}"'] \arrow[ur, bend left, "\bowtie'_{\mathsf{C}}"] &
\end{tikzcd} .$$

If Π is

$$\begin{tikzcd}
& \Pi'' \\
\Pi' & \begin{bmatrix}lG,G',G',\Psi''\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \\
\begin{bmatrix}\Psi\\ \ \end{bmatrix}\vdash\begin{bmatrix}G\\ \ \end{bmatrix} & \begin{bmatrix}lG,G',\Psi''\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \arrow[u, ">"] \\
& \begin{bmatrix}\Psi,G',\Psi''\\ \Gamma'\end{bmatrix}\vdash\begin{bmatrix}\Xi'\\ \Lambda'\end{bmatrix} \arrow[ul, bend right, "\bowtie'_{\mathsf{C}}"'] \arrow[u, "\bowtie'_{\mathsf{C}}"']
\end{tikzcd} ,$$

by the induction hypothesis, consider the $\bowtie'_{\mathsf{C}}$-free proof $\overline{\Pi}''$ corresponding to

$$\begin{tikzcd}
\Pi' \arrow[d, phantom] & & \Pi'' \arrow[d, phantom] \\
\begin{bmatrix}\Psi \\ \end{bmatrix} \vdash \begin{bmatrix}G \\ \end{bmatrix} & & \begin{bmatrix} lG, G', G', \Psi'' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix} \\
& \begin{bmatrix} \Psi, G', G', \Psi'' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix} \arrow[ul, "\bowtie'_{\mathsf{C}}"] \arrow[ur, "\bowtie'_{\mathsf{C}}"'] &
\end{tikzcd} .$$

Take

$$\overline{\Pi} = \begin{array}{c} \overline{\Pi}'' \\ \begin{bmatrix} \Psi, G', G', \Psi'' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix} \\ >\uparrow \\ \begin{bmatrix} \Psi, G', \Psi'' \\ \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix} \end{array} .$$

In all cases it is easy to check that $\mathsf{cr}(\overline{\Pi}) \leqslant \mathsf{cr}(\Pi)$. □

4.1.2. Lemma. *For every proof Π in* G-Forum$^{>,\bowtie_{\mathsf{L}},\bowtie'_{\mathsf{C}}}$ *a proof Π' exists in* G-Forum$^{>,\bowtie_{\mathsf{L}}}$ *whose conclusion is the same and* $\mathsf{cr}(\Pi') \leqslant \mathsf{cr}(\Pi)$.

Proof. If a $\bowtie'_{\mathsf{C}}$ instance appears in the proof Π, there must be in Π a subproof Π' which satisfies the hypotheses of Lemma 4.1.1. Replace Π' with its associated $\bowtie'_{\mathsf{C}}$-free proof $\overline{\Pi}'$ produced by the lemma. Proceed by induction on the number of $\bowtie'_{\mathsf{C}}$ instances in Π: at each step the cut-rank of the proof does not increase. □

4.2. Elimination of the linear cut rule

In this section, we see the core argument of the whole proof.

4.2.1. Lemma. *If there is in* G-Forum$^{>,\bowtie_{\mathsf{L}}}$ *a proof with conclusion* $\begin{bmatrix}\Psi \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix}$, *there is also a proof with conclusion* $\begin{bmatrix}\Psi, \Psi' \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix}$ *and whose cut-rank is the same, for any multiset Ψ'.*

Proof. Add Ψ' to the classical context in the conclusion of the proof and proceed recursively upwards: when a cut rule is reached, proceed recursively on one of the two proofs ending in its premises; when a rule different from a cut is reached, proceed recursively on all the proofs ending in its premises. The cut-rank is unchanged. □

4.2.2. Lemma. *In* G-Forum$^{>,\bowtie_{\mathsf{L}}}$ *let Π be the proof*

$$\begin{tikzcd}
\Pi' \arrow[d, phantom] & & \Pi'' \arrow[d, phantom] \\
\begin{bmatrix}\Psi \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G \\ \Lambda\end{bmatrix} & & \begin{bmatrix} \Psi' \\ G, \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix} \\
& \begin{bmatrix} \Psi, \Psi' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda, \Lambda'\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd} ,$$

such that $\mathsf{cr}(G) > \mathsf{cr}(\Pi')$ *and* $\mathsf{cr}(G) > \mathsf{cr}(\Pi'')$. *A proof $\overline{\Pi}$ exists in* G-Forum$^{>,\bowtie_{\mathsf{L}}}$, *whose conclusion is the same as that of Π, and* $\mathsf{cr}(\overline{\Pi}) < \mathsf{cr}(\Pi)$.

Proof. The proof is by induction on $\mathsf{d}(\Pi') + \mathsf{d}(\Pi'')$.

Basis case: The only basis case occurs when Π is

$$\begin{tikzcd}
\circ & & \circ \\
\begin{bmatrix} \Psi \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda \end{bmatrix} \arrow[u, "\mathsf{G}_{\mathsf{R}}"] & & \begin{bmatrix} \Psi' \\ \forall\vec{x}.\top, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G}_{\mathsf{R}}"] \\
& \begin{bmatrix} \Psi, \Psi' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda, \Lambda' \end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd} ;$$

in this case consider

$$\bar{\Pi} = \begin{tikzcd}
\circ \\
\begin{bmatrix} \Psi, \Psi' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \forall\vec{x}.\top \\ \Lambda, \Lambda' \end{bmatrix} \arrow[u, "\mathsf{G}_{\mathsf{R}}"]
\end{tikzcd} .$$

Inductive cases: We consider the bottommost rule instances in Π' and Π''. The bottommost rule in Π' cannot be a G_{L}, so we can make an exhaustive case analysis by considering the following couples 'left bottommost, right bottommost':

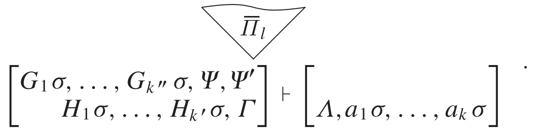
(1) G_{R}, G_{L} (where the principal formula is active);
(2) any rule, G_{L} (where the principal formula is not active);
(3) any rule, G_{R};
(4) any rule, $\bowtie_{\mathsf{L}}$ (principal formula goes to the left);
(5) any rule, $\bowtie_{\mathsf{L}}$ (principal formula goes to the right);
(6) any rule, $>$;
(7) $\bowtie_{\mathsf{L}}$, any rule;
(8) $>$, any rule.

In detail:

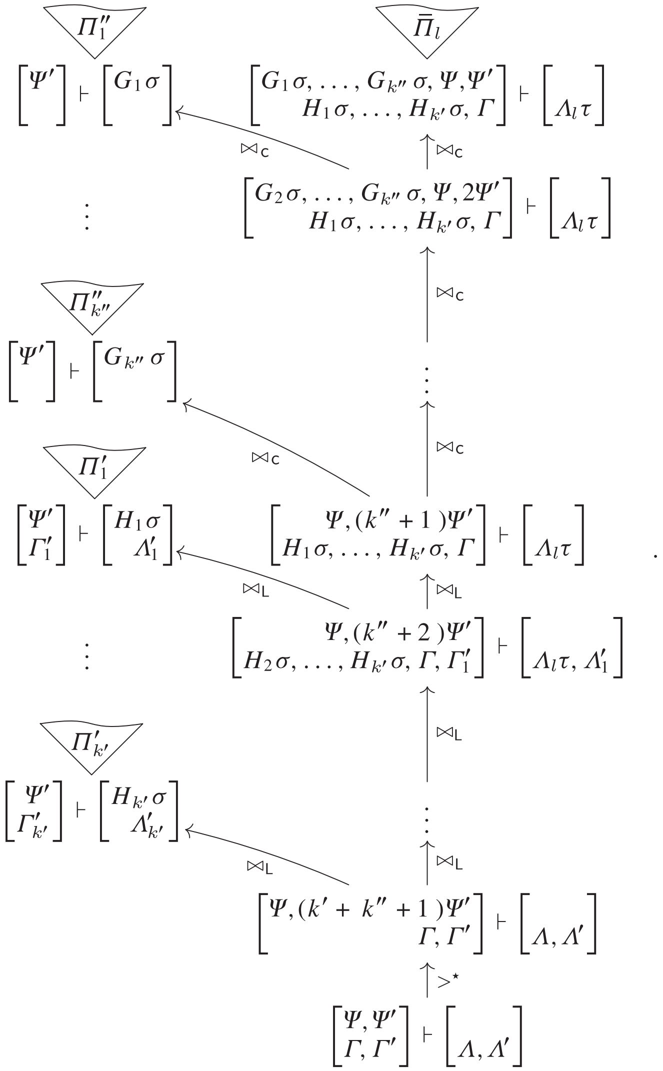
(1) G_{R}, G_{L} (where the principal formula is active). Let Π be

$$\begin{tikzcd}
\begin{array}{c} \overset{\Pi_1}{\bigtriangledown} \\ \begin{bmatrix} \Psi_1 \\ \Gamma_1 \end{bmatrix} \vdash \begin{bmatrix} \Lambda_1 \end{bmatrix} \end{array} & & \begin{array}{c} \overset{\Pi''_1}{\bigtriangledown} \\ \begin{bmatrix} \Psi' \end{bmatrix} \vdash \begin{bmatrix} G_1\sigma \end{bmatrix} \end{array} & \cdots & \begin{array}{c} \overset{\Pi''_{k''}}{\bigtriangledown} \\ \begin{bmatrix} \Psi' \end{bmatrix} \vdash \begin{bmatrix} G_{k''}\sigma \end{bmatrix} \end{array} \\
& \ddots & & & \\
& & \begin{array}{c} \overset{\Pi_l}{\bigtriangledown} \\ \begin{bmatrix} \Psi_l \\ \Gamma_l \end{bmatrix} \vdash \begin{bmatrix} \Lambda_l \end{bmatrix} \end{array} & & \begin{array}{c} \overset{\Pi'_1}{\bigtriangledown} \\ \begin{bmatrix} \Psi' \\ \Gamma'_1 \end{bmatrix} \vdash \begin{bmatrix} H_1\sigma \\ \Lambda'_1 \end{bmatrix} \end{array} \\
& & \vdots & & \vdots \\
& & \begin{array}{c} \overset{\Pi_h}{\bigtriangledown} \\ \begin{bmatrix} \Psi_h \\ \Gamma_h \end{bmatrix} \vdash \begin{bmatrix} \Lambda_h \end{bmatrix} \end{array} & & \begin{array}{c} \overset{\Pi'_{k'}}{\bigtriangledown} \\ \begin{bmatrix} \Psi' \\ \Gamma'_{k'} \end{bmatrix} \vdash \begin{bmatrix} H_{k'}\sigma \\ \Lambda'_{k'} \end{bmatrix} \end{array} \\
\begin{bmatrix} \Psi \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} G \\ \Lambda \end{bmatrix} \arrow[uuuuu, "\mathsf{G}_{\mathsf{R}}"] \arrow[uuurr, bend left, "\mathsf{G}_{\mathsf{R}}"] \arrow[urr, bend left, "\mathsf{G}_{\mathsf{R}}"'] & & & \begin{bmatrix} \Psi' \\ G, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Lambda' \end{bmatrix} \arrow[uuuuul, "\mathsf{G}_{\mathsf{L}}"] \arrow[uuuuur, "\mathsf{G}_{\mathsf{L}}"] \arrow[uuur, bend left, "\mathsf{G}_{\mathsf{L}}"] \arrow[ur, bend left, "\mathsf{G}_{\mathsf{L}}"'] & \\
& & \begin{bmatrix} \Psi, \Psi' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Lambda, \Lambda' \end{bmatrix} \arrow[ull, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] & &
\end{tikzcd} ;$$

where $G = \forall\vec{x}.(\delta_1 \mathbin{\&} \cdots \mathbin{\&} \delta_h)$ is the selected goal in the $\mathsf{G_L}$ instance and $h > 0$ (the case $h = 0$ is impossible); the selected clause in $\mathsf{G_L}$ is $\delta_l\sigma = (G_1 \Rightarrow \cdots \Rightarrow G_{k''} \Rightarrow H_1 \multimap \cdots \multimap H_{k'} \multimap a_1 \parr \cdots \parr a_k)\sigma$, where $k, k', k'' \geqslant 0$ and $1 \leqslant l \leqslant h$; for some renaming substitution ρ they hold $\Psi_l = \Psi \uplus \{G_1\rho, \ldots, G_{k''}\rho\}_+$, $\Gamma_l = \Gamma \uplus \{H_1\rho, \ldots, H_{k'}\rho\}_+$ and $\Lambda_l = \Lambda \uplus \{a_1\rho, \ldots, a_k\rho\}_+$; $\Gamma' = \Gamma'_1 \uplus \cdots \uplus \Gamma'_{k'}$ and $\Lambda' = \Lambda'_1 \uplus \cdots \uplus \Lambda'_{k'} \uplus \{a_1\sigma, \ldots, a_k\sigma\}_+$. Let τ be a substitution whose domain is the range of ρ and such that $\rho\tau = \sigma$. Consider the proof $\overline{\Pi}_l$ obtained from Π_l by applying τ to every formula in every sequent (this operation may of course involve the renaming of some eigenvariable); moreover Ψ' is added to the classical context (Lemma 4.2.1):


$\overline{\Pi}_l$
$\begin{bmatrix} G_1\sigma, \ldots, G_{k''}\sigma, \Psi, \Psi' \\ H_1\sigma, \ldots, H_{k'}\sigma, \Gamma \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda, a_1\sigma, \ldots, a_k\sigma \end{bmatrix}$.

Of course, $\mathsf{cr}(\overline{\Pi}_l) = \mathsf{cr}(\Pi_l)$. Let $\overline{\Pi}'$ be the following proof:

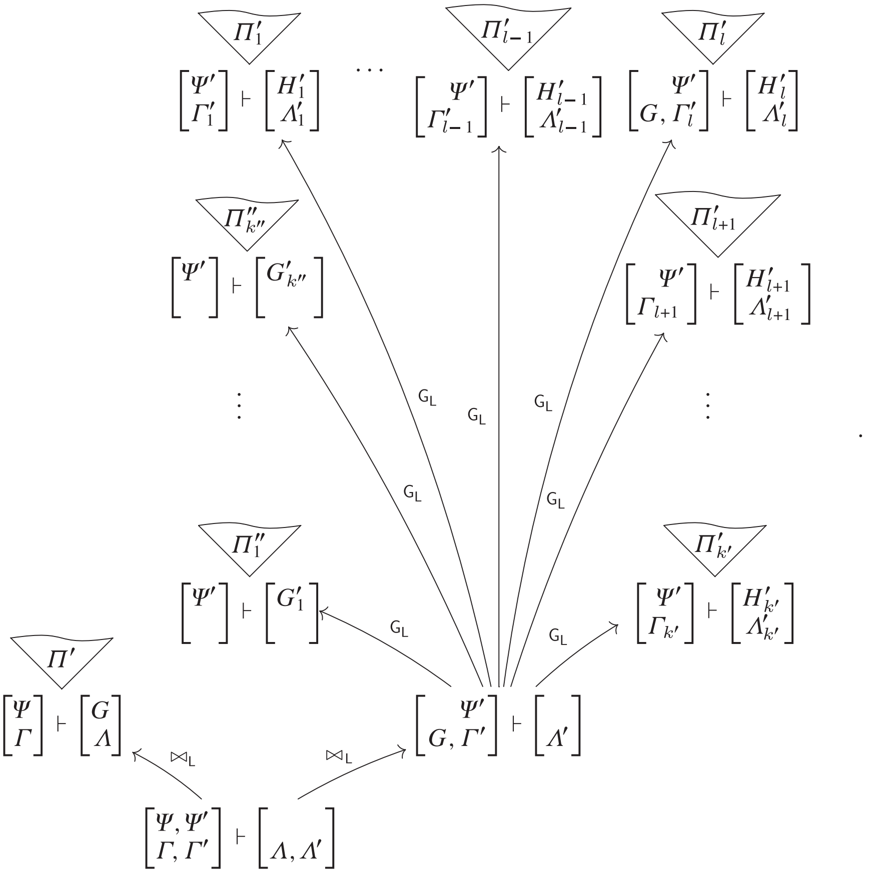

Π''_1
$\begin{bmatrix} \Psi' \\ \end{bmatrix} \vdash \begin{bmatrix} G_1\sigma \\ \end{bmatrix}$
$\overline{\Pi}_l$
$\begin{bmatrix} G_1\sigma, \ldots, G_{k''}\sigma, \Psi, \Psi' \\ H_1\sigma, \ldots, H_{k'}\sigma, \Gamma \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda_l\tau \end{bmatrix}$
$\bowtie_{\mathsf{C}}$
$\bowtie_{\mathsf{C}}$
$\begin{bmatrix} G_2\sigma, \ldots, G_{k''}\sigma, \Psi, 2\Psi' \\ H_1\sigma, \ldots, H_{k'}\sigma, \Gamma \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda_l\tau \end{bmatrix}$
$\bowtie_{\mathsf{C}}$
$\Pi''_{k''}$
$\begin{bmatrix} \Psi' \\ \end{bmatrix} \vdash \begin{bmatrix} G_{k''}\sigma \\ \end{bmatrix}$
$\bowtie_{\mathsf{C}}$
$\bowtie_{\mathsf{C}}$
Π'_1
$\begin{bmatrix} \Psi' \\ \Gamma'_1 \end{bmatrix} \vdash \begin{bmatrix} H_1\sigma \\ \Lambda'_1 \end{bmatrix}$
$\begin{bmatrix} \Psi, (k''+1)\Psi' \\ H_1\sigma, \ldots, H_{k'}\sigma, \Gamma \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda_l\tau \end{bmatrix}$.
$\bowtie_{\mathsf{L}}$
$\bowtie_{\mathsf{L}}$
$\begin{bmatrix} \Psi, (k''+2)\Psi' \\ H_2\sigma, \ldots, H_{k'}\sigma, \Gamma, \Gamma'_1 \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda_l\tau, \Lambda'_1 \end{bmatrix}$
$\bowtie_{\mathsf{L}}$
$\Pi'_{k'}$
$\begin{bmatrix} \Psi' \\ \Gamma'_{k'} \end{bmatrix} \vdash \begin{bmatrix} H_{k'}\sigma \\ \Lambda'_{k'} \end{bmatrix}$
$\bowtie_{\mathsf{L}}$
$\bowtie_{\mathsf{L}}$
$\begin{bmatrix} \Psi, (k'+k''+1)\Psi' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda, \Lambda' \end{bmatrix}$
$>^\star$
$\begin{bmatrix} \Psi, \Psi' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda, \Lambda' \end{bmatrix}$

We have

$$\begin{aligned}\mathsf{cr}(\overline{\Pi}') = &\max\{\mathsf{cr}(\overline{\Pi}_l), \mathsf{cr}(\Pi''_1), \ldots, \mathsf{cr}(\Pi''_{k''}), \mathsf{cr}(\Pi'_1), \ldots, \mathsf{cr}(\Pi'_{k'}),\\ &\quad \mathsf{cr}(G_1\sigma), \ldots, \mathsf{cr}(G_{k''}\sigma), \mathsf{cr}(H_1\sigma), \ldots, \mathsf{cr}(H_{k'}\sigma)\}\\ &< \mathsf{cr}(G).\end{aligned}$$

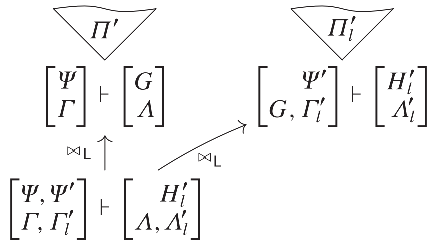
Apply Lemma 4.1.2 to $\overline{\Pi}'$ and obtain $\overline{\Pi}$: the cut-rank does not increase.

(2) Any rule, $\mathsf{G_L}$ (where the principal formula is not active). Let Π be



Π'_1
$\begin{bmatrix}\Psi'\\ \Gamma'_1\end{bmatrix} \vdash \begin{bmatrix}H'_1\\ \Lambda'_1\end{bmatrix}$
$\ldots$
Π'_{l-1}
$\begin{bmatrix}\Psi'\\ \Gamma'_{l-1}\end{bmatrix} \vdash \begin{bmatrix}H'_{l-1}\\ \Lambda'_{l-1}\end{bmatrix}$
Π'_l
$\begin{bmatrix}\Psi'\\ G, \Gamma'_l\end{bmatrix} \vdash \begin{bmatrix}H'_l\\ \Lambda'_l\end{bmatrix}$
$\Pi''_{k''}$
$\begin{bmatrix}\Psi'\end{bmatrix} \vdash \begin{bmatrix}G'_{k''}\end{bmatrix}$
Π'_{l+1}
$\begin{bmatrix}\Psi'\\ \Gamma_{l+1}\end{bmatrix} \vdash \begin{bmatrix}H'_{l+1}\\ \Lambda'_{l+1}\end{bmatrix}$
$\vdots$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\vdots$
Π''_1
$\begin{bmatrix}\Psi'\end{bmatrix} \vdash \begin{bmatrix}G'_1\end{bmatrix}$
$\Pi'_{k'}$
$\begin{bmatrix}\Psi'\\ \Gamma_{k'}\end{bmatrix} \vdash \begin{bmatrix}H'_{k'}\\ \Lambda'_{k'}\end{bmatrix}$
$\mathsf{G_L}$
$\mathsf{G_L}$
Π'
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G\\ \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi'\\ G, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda'\end{bmatrix}$
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi, \Psi'\\ \Gamma, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda, \Lambda'\end{bmatrix}$
.

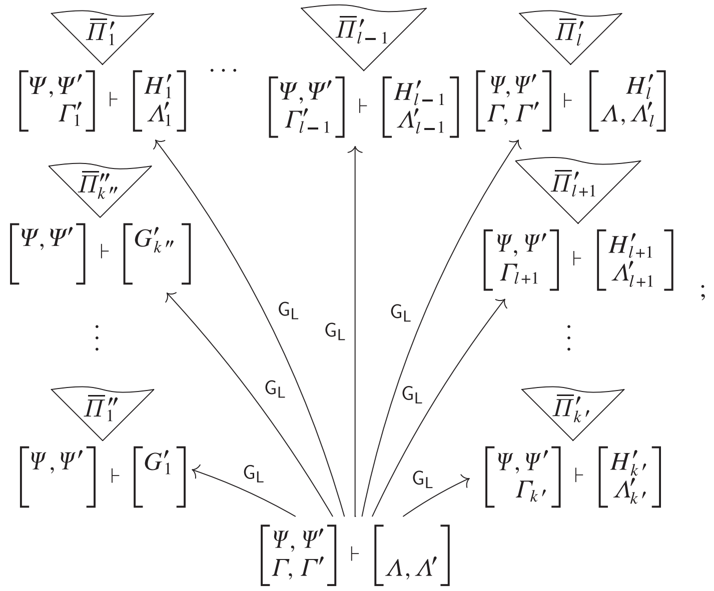
where $k'' \geqslant 0$, goal G is not selected, and then $k' > 0$ and $1 \leqslant l \leqslant k'$. Let $\overline{\Pi}''_1, \ldots, \overline{\Pi}''_{k''}, \overline{\Pi}'_1, \ldots, \overline{\Pi}'_{l-1}, \overline{\Pi}'_{l+1}, \ldots, \overline{\Pi}'_{k'}$, respectively, be obtained by $\Pi''_1, \ldots, \Pi''_{k''}, \Pi'_1, \ldots, \Pi'_{l-1}, \Pi'_{l+1}, \ldots, \Pi'_{k'}$ by adding Ψ to the classical context in the conclusion (by Lemma 4.2.1). By the induction hypothesis, there is a proof $\overline{\Pi}'_l$ with the same conclusion of



Π'
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G\\ \Lambda\end{bmatrix}$
Π'_l
$\begin{bmatrix}\Psi'\\ G, \Gamma'_l\end{bmatrix} \vdash \begin{bmatrix}H'_l\\ \Lambda'_l\end{bmatrix}$
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi, \Psi'\\ \Gamma, \Gamma'_l\end{bmatrix} \vdash \begin{bmatrix}H'_l\\ \Lambda, \Lambda'_l\end{bmatrix}$

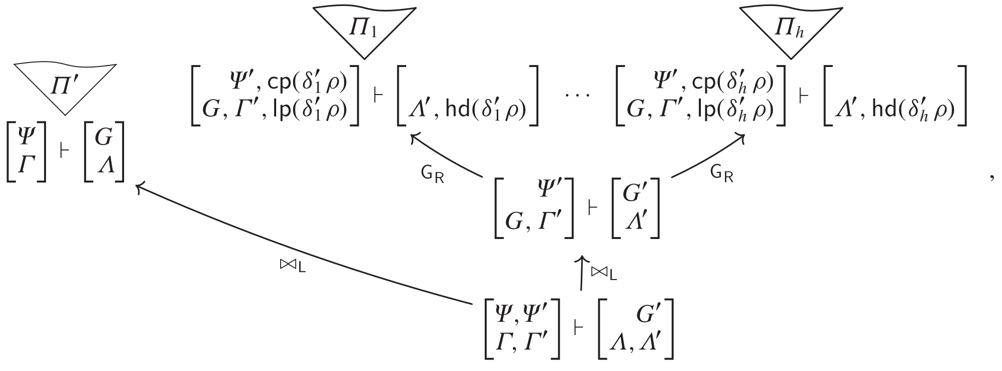
,

and such that $\mathsf{cr}(\overline{\Pi}'_l) < \mathsf{cr}(G)$. Take $\overline{\Pi}$ as

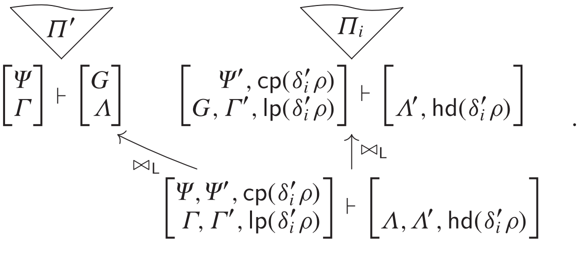

$\overline{\Pi}'_1$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma'_1\end{bmatrix} \vdash \begin{bmatrix}H'_1\\ \Lambda'_1\end{bmatrix}$
$\cdots$
$\overline{\Pi}'_{l-1}$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma'_{l-1}\end{bmatrix} \vdash \begin{bmatrix}H'_{l-1}\\ \Lambda'_{l-1}\end{bmatrix}$
$\overline{\Pi}'_l$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}H'_l\\ \Lambda,\Lambda'_l\end{bmatrix}$
$\overline{\Pi}''_{k''}$
$\begin{bmatrix}\Psi,\Psi'\\ \end{bmatrix} \vdash \begin{bmatrix}G'_{k''}\\ \end{bmatrix}$
$\overline{\Pi}'_{l+1}$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma_{l+1}\end{bmatrix} \vdash \begin{bmatrix}H'_{l+1}\\ \Lambda'_{l+1}\end{bmatrix}$
;
$\vdots$
$\vdots$
$\overline{\Pi}''_1$
$\begin{bmatrix}\Psi,\Psi'\\ \end{bmatrix} \vdash \begin{bmatrix}G'_1\\ \end{bmatrix}$
$\overline{\Pi}'_{k'}$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma_{k'}\end{bmatrix} \vdash \begin{bmatrix}H'_{k'}\\ \Lambda'_{k'}\end{bmatrix}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\mathsf{G_L}$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda,\Lambda'\end{bmatrix}$

$\mathsf{cr}(\overline{\Pi}) < \mathsf{cr}(\Pi)$ is easily verified.

(3) Any rule, $\mathsf{G_R}$. Let Π be


Π'
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G\\ \Lambda\end{bmatrix}$
Π_1
$\begin{bmatrix}\Psi', \mathsf{cp}(\delta'_1\rho)\\ G,\Gamma',\mathsf{lp}(\delta'_1\rho)\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda',\mathsf{hd}(\delta'_1\rho)\end{bmatrix}$
$\cdots$
Π_h
$\begin{bmatrix}\Psi', \mathsf{cp}(\delta'_h\rho)\\ G,\Gamma',\mathsf{lp}(\delta'_h\rho)\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda',\mathsf{hd}(\delta'_h\rho)\end{bmatrix}$
$\mathsf{G_R}$
$\mathsf{G_R}$
,
$\begin{bmatrix}\Psi'\\ G,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda'\end{bmatrix}$
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi,\Psi'\\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda,\Lambda'\end{bmatrix}$

where $G' = \forall\vec{x}.(\delta'_1 \mathbin{\&} \cdots \mathbin{\&} \delta'_h)$ and $h > 0$. Obtain, using the induction hypothesis, proofs $\overline{\Pi}_i$, where $1 \leqslant i \leqslant h$, and such that $\mathsf{cr}(\overline{\Pi}_i) < \mathsf{cr}(G)$, from proofs


Π'
$\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G\\ \Lambda\end{bmatrix}$
Π_i
$\begin{bmatrix}\Psi', \mathsf{cp}(\delta'_i\rho)\\ G,\Gamma',\mathsf{lp}(\delta'_i\rho)\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda',\mathsf{hd}(\delta'_i\rho)\end{bmatrix}$
.
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi,\Psi', \mathsf{cp}(\delta'_i\rho)\\ \Gamma,\Gamma',\mathsf{lp}(\delta'_i\rho)\end{bmatrix} \vdash \begin{bmatrix}\\ \Lambda,\Lambda',\mathsf{hd}(\delta'_i\rho)\end{bmatrix}$

Take $\overline{\overline{\Pi}}$ as

$$\begin{tikzcd}
\overline{\overline{\Pi}}_1 & & \overline{\overline{\Pi}}_h \\
\begin{bmatrix}\Psi,\Psi',\mathsf{cp}(\delta'_1\,\rho)\\ \Gamma,\Gamma',\mathsf{lp}(\delta'_1\,\rho)\end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda,\Lambda',\mathsf{hd}(\delta'_1\,\rho)\end{bmatrix} & \dots & \begin{bmatrix}\Psi,\Psi',\mathsf{cp}(\delta'_h\,\rho)\\ \Gamma,\Gamma',\mathsf{lp}(\delta'_h\,\rho)\end{bmatrix} \vdash \begin{bmatrix} \\ \Lambda,\Lambda',\mathsf{hd}(\delta'_h\,\rho)\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi'\\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda,\Lambda'\end{bmatrix} \arrow[ul, "\mathsf{G_R}"] \arrow[ur, "\mathsf{G_R}"'] &
\end{tikzcd} .$$

When $h = 0$ take $\overline{\overline{\Pi}}$ as

$$\begin{tikzcd}
\circ \\
\begin{bmatrix}\Psi,\Psi'\\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}\forall\vec{y}.\top\\ \Lambda,\Lambda'\end{bmatrix} \arrow[u, "\mathsf{G_R}"]
\end{tikzcd} .$$

(4) Any rule, $\bowtie_{\mathsf{L}}$ (principal formula goes to the left). Let Π be

$$\begin{tikzcd}
& \Pi_1 & \Pi_2 \\
\Pi' & \begin{bmatrix}\Psi'_1\\ G,\Gamma'_1\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda'_1\end{bmatrix} & \begin{bmatrix}\Psi'_2\\ G',\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda'_2\end{bmatrix} \\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G\\ \Lambda\end{bmatrix} & \begin{bmatrix}\Psi'_1,\Psi'_2\\ G,\Gamma'_1,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[u, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] & \\
& \begin{bmatrix}\Psi,\Psi'_1,\Psi'_2\\ \Gamma,\Gamma'_1,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix}\Xi'\\ \Lambda,\Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"] \arrow[u, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd} .$$

Apply the induction hypothesis on

$$\begin{tikzcd}
\Pi' & & \Pi_1 \\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G\\ \Lambda\end{bmatrix} & & \begin{bmatrix}\Psi'_1\\ G,\Gamma'_1\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda'_1\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi'_1\\ \Gamma,\Gamma'_1\end{bmatrix} \vdash \begin{bmatrix}G'\\ \Lambda,\Lambda'_1\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd}$$

and obtain $\overline{\Pi}_1$. Then take $\overline{\Pi}$ as

$$\begin{tikzcd}
\overline{\Pi}_1 & & \Pi_2 \\
\begin{bmatrix}\Psi,\Psi'_1\\ \Gamma,\Gamma'_1\end{bmatrix} \vdash \begin{bmatrix} G'\\ \Lambda,\Lambda'_1\end{bmatrix} & & \begin{bmatrix}\Psi'_2\\ G',\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda'_2\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi'_1,\Psi'_2\\ \Gamma,\Gamma'_1,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda,\Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"'] \arrow[ur, "\bowtie_{\mathsf{L}}"] &
\end{tikzcd} .$$

(5) Any rule, $\bowtie_{\mathsf{L}}$ (principal formula goes to the right). Let Π be

$$\begin{tikzcd}
& \Pi_1 & & \Pi_2 \\
\Pi' & \begin{bmatrix}\Psi'_1\\ \Gamma'_1\end{bmatrix} \vdash \begin{bmatrix} G'\\ \Lambda'_1\end{bmatrix} & & \begin{bmatrix}\Psi'_2\\ G',G,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda'_2\end{bmatrix} \\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix} G\\ \Lambda\end{bmatrix} & & \begin{bmatrix}\Psi'_1,\Psi'_2\\ G,\Gamma'_1,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"'] \arrow[ur, "\bowtie_{\mathsf{L}}"] & \\
& & \begin{bmatrix}\Psi,\Psi'_1,\Psi'_2\\ \Gamma,\Gamma'_1,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda,\Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[ull, "\bowtie_{\mathsf{L}}"'] \arrow[u, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd} .$$

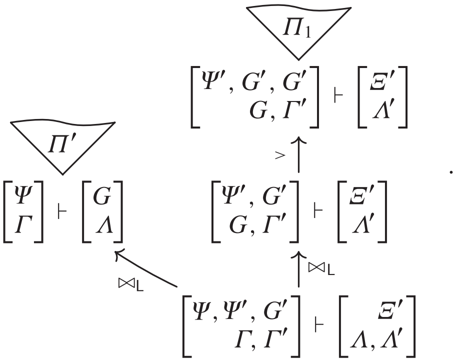
Apply the induction hypothesis on

$$\begin{tikzcd}
\Pi' & & \Pi_2 \\
\begin{bmatrix}\Psi\\ \Gamma\end{bmatrix} \vdash \begin{bmatrix} G\\ \Lambda\end{bmatrix} & & \begin{bmatrix}\Psi'_2\\ G',G,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda'_2\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi'_2\\ G',\Gamma,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda,\Lambda'_2\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"'] \arrow[ur, "\bowtie_{\mathsf{L}}"] &
\end{tikzcd}$$

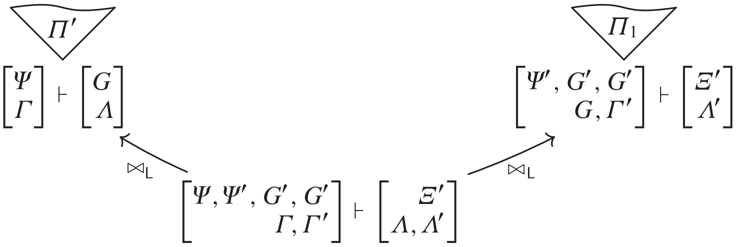
and obtain $\overline{\Pi}_2$. Then take $\overline{\Pi}$ as

$$\begin{tikzcd}
\Pi_1 & & \overline{\Pi}_2 \\
\begin{bmatrix}\Psi'_1\\ \Gamma'_1\end{bmatrix} \vdash \begin{bmatrix} G'\\ \Lambda'_1\end{bmatrix} & & \begin{bmatrix}\Psi,\Psi'_2\\ G',\Gamma,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda,\Lambda'_2\end{bmatrix} \\
& \begin{bmatrix}\Psi,\Psi'_1,\Psi'_2\\ \Gamma,\Gamma'_1,\Gamma'_2\end{bmatrix} \vdash \begin{bmatrix} \Xi'\\ \Lambda,\Lambda'_1,\Lambda'_2\end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"'] \arrow[ur, "\bowtie_{\mathsf{L}}"] &
\end{tikzcd} .$$

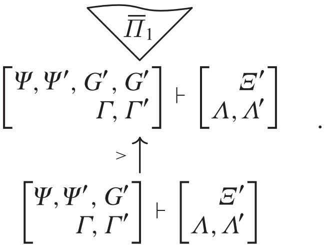
(6) Any rule, $>$. Let Π be


Π_1
$\begin{bmatrix}\Psi', G', G' \\ G, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix}$
$>$
Π'
$\begin{bmatrix}\Psi \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G \\ \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi', G' \\ G, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix}$
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi, \Psi', G' \\ \Gamma, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda, \Lambda'\end{bmatrix}$
.

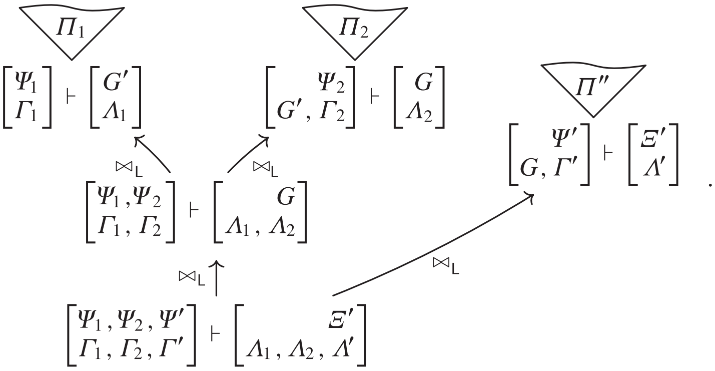
Apply the induction hypothesis on


Π'
Π_1
$\begin{bmatrix}\Psi \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}G \\ \Lambda\end{bmatrix}$
$\begin{bmatrix}\Psi', G', G' \\ G, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix}$
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi, \Psi', G', G' \\ \Gamma, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda, \Lambda'\end{bmatrix}$

and obtain $\overline{\Pi}_1$. Then take $\overline{\Pi}$ as


$\overline{\Pi}_1$
$\begin{bmatrix}\Psi, \Psi', G', G' \\ \Gamma, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda, \Lambda'\end{bmatrix}$
$>$
$\begin{bmatrix}\Psi, \Psi', G' \\ \Gamma, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda, \Lambda'\end{bmatrix}$
.

(7) $\bowtie_\mathsf{L}$, Any rule. Let Π be


Π_1
Π_2
$\begin{bmatrix}\Psi_1 \\ \Gamma_1\end{bmatrix} \vdash \begin{bmatrix}G' \\ \Lambda_1\end{bmatrix}$
$\begin{bmatrix}\Psi_2 \\ G', \Gamma_2\end{bmatrix} \vdash \begin{bmatrix}G \\ \Lambda_2\end{bmatrix}$
Π''
$\begin{bmatrix}\Psi' \\ G, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda'\end{bmatrix}$
.
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi_1, \Psi_2 \\ \Gamma_1, \Gamma_2\end{bmatrix} \vdash \begin{bmatrix}G \\ \Lambda_1, \Lambda_2\end{bmatrix}$
$\bowtie_\mathsf{L}$
$\bowtie_\mathsf{L}$
$\begin{bmatrix}\Psi_1, \Psi_2, \Psi' \\ \Gamma_1, \Gamma_2, \Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda_1, \Lambda_2, \Lambda'\end{bmatrix}$

Apply the induction hypothesis on

$$\begin{tikzcd}
\Pi_2 & & \Pi'' \\
\begin{bmatrix} \Psi_2 \\ G', \Gamma_2 \end{bmatrix} \vdash \begin{bmatrix} G \\ \Lambda_2 \end{bmatrix} & & \begin{bmatrix} \Psi' \\ G, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda' \end{bmatrix} \\
\begin{bmatrix} \Psi_2, \Psi' \\ G', \Gamma_2, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda_2, \Lambda' \end{bmatrix} \arrow[u, "\bowtie_{\mathsf{L}}"] \arrow[urr, "\bowtie_{\mathsf{L}}"']
\end{tikzcd}$$

and obtain $\overline{\Pi}_2$. Then take $\overline{\Pi}$ as

$$\begin{tikzcd}
\Pi_1 & & \overline{\Pi}_2 \\
\begin{bmatrix} \Psi_1 \\ \Gamma_1 \end{bmatrix} \vdash \begin{bmatrix} G' \\ \Lambda_1 \end{bmatrix} & & \begin{bmatrix} \Psi_2, \Psi' \\ G', \Gamma_2, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda_2, \Lambda' \end{bmatrix} \\
\begin{bmatrix} \Psi_1, \Psi_2, \Psi' \\ \Gamma_1, \Gamma_2, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda_1, \Lambda_2, \Lambda' \end{bmatrix} \arrow[u, "\bowtie_{\mathsf{L}}"] \arrow[urr, "\bowtie_{\mathsf{L}}"']
\end{tikzcd} .$$

(8) $>$, any rule. Let Π be

$$\begin{tikzcd}
\Pi_1 & \\
\begin{bmatrix} \Psi, G', G' \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} G \\ \Lambda \end{bmatrix} & \Pi'' \\
\begin{bmatrix} \Psi, G' \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} G \\ \Lambda \end{bmatrix} \arrow[u, ">"] & \begin{bmatrix} \Psi' \\ G, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda' \end{bmatrix} \\
\begin{bmatrix} \Psi, \Psi', G' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda, \Lambda' \end{bmatrix} \arrow[u, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"']
\end{tikzcd} .$$

Apply the induction hypothesis on

$$\begin{tikzcd}
\Pi_1 & & \Pi'' \\
\begin{bmatrix} \Psi, G', G' \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} G \\ \Lambda \end{bmatrix} & & \begin{bmatrix} \Psi' \\ G, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda' \end{bmatrix} \\
& \begin{bmatrix} \Psi, \Psi', G', G' \\ \Gamma, \Gamma' \end{bmatrix} \vdash \begin{bmatrix} \Xi' \\ \Lambda, \Lambda' \end{bmatrix} \arrow[ul, "\bowtie_{\mathsf{L}}"] \arrow[ur, "\bowtie_{\mathsf{L}}"'] &
\end{tikzcd}$$

and obtain $\overline{\Pi}_1$. Then take $\overline{\Pi}$ as

$$\begin{array}{c} \overline{\Pi}_1 \\ \begin{bmatrix}\Psi,\Psi',G',G' \\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda,\Lambda'\end{bmatrix} \\ {\scriptstyle >}\uparrow \\ \begin{bmatrix}\Psi,\Psi',G' \\ \Gamma,\Gamma'\end{bmatrix} \vdash \begin{bmatrix}\Xi' \\ \Lambda,\Lambda'\end{bmatrix} \end{array} . \qquad \square$$

4.2.3. Lemma. *For every proof in* G-Forum$^{>,\bowtie_{\mathsf{L}}}$ *there is a proof in* G-Forum$^{>}$ *whose conclusion is the same.*

Proof. Given a proof in G-Forum$^{>,\bowtie_{\mathsf{L}}}$, we say that a certain $\bowtie_{\mathsf{L}}$ instance is *maximal* if its cut-rank is higher than that of any other $\bowtie_{\mathsf{L}}$ instance above it. Apply repeatedly Lemma 4.2.2 on maximal $\bowtie_{\mathsf{L}}$ instances until all $\bowtie_{\mathsf{L}}$ instances disappear: they do because the cut-rank in the subproofs whose root is a maximal $\bowtie_{\mathsf{L}}$ decreases each time. □

4.3. Elimination of the contraction rule

4.3.1. Lemma. *In* G-Forum$^{>}$, *let* Π *be the proof*

$$\begin{array}{c} \Pi' \\ \begin{bmatrix}\Psi,G,G \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix} \\ {\scriptstyle >}\uparrow \\ \begin{bmatrix}\Psi,G \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix} \end{array} ,$$

where no instance of $>$ *appears in* Π'. *A proof* $\overline{\Pi}$ *exists in* G-Forum *whose conclusion is the same as that of* Π.

Proof. By induction on the depth of Π'.

Basis case: If Π is

$$\begin{array}{c} \circ \\ {\scriptstyle r}\uparrow \\ \begin{bmatrix}\Psi,G,G \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix} \\ {\scriptstyle >}\uparrow \\ \begin{bmatrix}\Psi,G \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix} \end{array} ,$$

then take

$$\overline{\Pi} = \begin{array}{c} \circ \\ {\scriptstyle r}\uparrow \\ \begin{bmatrix}\Psi,G \\ \Gamma\end{bmatrix} \vdash \begin{bmatrix}\Xi \\ \Lambda\end{bmatrix} \end{array} ,$$

where r is $\mathsf{G_L}$ or $\mathsf{G_R}$.

Inductive cases:

(1) $\mathsf{G_L}$. Let Π be

$$\begin{tikzcd}
\Pi''_1 \arrow[d, phantom] & & \Pi''_{k''} \arrow[d, phantom] & \Pi'_1 \arrow[d, phantom] & & \Pi'_{k'} \arrow[d, phantom] \\
\begin{bmatrix}\Psi, G, G \\ {} \end{bmatrix} \vdash \begin{bmatrix} G_1 \\ {} \end{bmatrix} & \cdots & \begin{bmatrix}\Psi, G, G \\ {} \end{bmatrix} \vdash \begin{bmatrix} G_{k''} \\ {} \end{bmatrix} & \begin{bmatrix}\Psi, G, G \\ \Gamma_1 \end{bmatrix} \vdash \begin{bmatrix} H_1 \\ \Lambda_1 \end{bmatrix} & \cdots & \begin{bmatrix}\Psi, G, G \\ \Gamma_{k'} \end{bmatrix} \vdash \begin{bmatrix} H_{k'} \\ \Lambda_{k'} \end{bmatrix} \\
& & \begin{bmatrix}\Psi, G, G \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} {} \\ \Lambda \end{bmatrix} \arrow[ull, "\mathsf{G_L}"] \arrow[u, "\mathsf{G_L}"'] \arrow[ur, "\mathsf{G_L}"] \arrow[urrr, "\mathsf{G_L}"'] & & & \\
& & \begin{bmatrix}\Psi, G \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} {} \\ \Lambda \end{bmatrix} \arrow[u, ">"'] & & &
\end{tikzcd}$$

where $k' + k'' > 0$. By the induction hypothesis, for $1 \leqslant j \leqslant k''$ and $1 \leqslant i \leqslant k'$, there are $>$-free proofs $\overline{\Pi}''_j$ and $\overline{\Pi}'_i$, corresponding, respectively, to

$$\begin{tikzcd}
\Pi''_j \arrow[d, phantom] & & \Pi'_i \arrow[d, phantom] \\
\begin{bmatrix}\Psi, G, G \\ {} \end{bmatrix} \vdash \begin{bmatrix} G_j \\ {} \end{bmatrix} & \text{and} & \begin{bmatrix}\Psi, G, G \\ \Gamma_i \end{bmatrix} \vdash \begin{bmatrix} H_i \\ \Lambda_i \end{bmatrix} \\
\begin{bmatrix}\Psi, G \\ {} \end{bmatrix} \vdash \begin{bmatrix} G_j \\ {} \end{bmatrix} \arrow[u, ">"] & & \begin{bmatrix}\Psi, G \\ \Gamma_i \end{bmatrix} \vdash \begin{bmatrix} H_i \\ \Lambda_i \end{bmatrix} \arrow[u, ">"]
\end{tikzcd} .$$

Then take $\overline{\Pi}$ as

$$\begin{tikzcd}
\overline{\Pi}''_1 \arrow[d, phantom] & & \overline{\Pi}''_{k''} \arrow[d, phantom] & \overline{\Pi}'_1 \arrow[d, phantom] & & \overline{\Pi}'_{k'} \arrow[d, phantom] \\
\begin{bmatrix}\Psi, G \\ {} \end{bmatrix} \vdash \begin{bmatrix} G_1 \\ {} \end{bmatrix} & \cdots & \begin{bmatrix}\Psi, G \\ {} \end{bmatrix} \vdash \begin{bmatrix} G_{k''} \\ {} \end{bmatrix} & \begin{bmatrix}\Psi, G \\ \Gamma_1 \end{bmatrix} \vdash \begin{bmatrix} H_1 \\ \Lambda_1 \end{bmatrix} & \cdots & \begin{bmatrix}\Psi, G \\ \Gamma_{k'} \end{bmatrix} \vdash \begin{bmatrix} H_{k'} \\ \Lambda_{k'} \end{bmatrix} \\
& & \begin{bmatrix}\Psi, G \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} {} \\ \Lambda \end{bmatrix} \arrow[ull, "\mathsf{G_L}"] \arrow[u, "\mathsf{G_L}"'] \arrow[ur, "\mathsf{G_L}"] \arrow[urrr, "\mathsf{G_L}"'] & & &
\end{tikzcd} .$$

(2) $\mathsf{G_R}$. Let Π be

$$\begin{tikzcd}
\Pi_1 \arrow[d, phantom] & & \Pi_h \arrow[d, phantom] \\
\begin{bmatrix}\Psi, G, G, \mathsf{cp}(\delta_1 \rho) \\ \Gamma, \mathsf{lp}(\delta_1 \rho) \end{bmatrix} \vdash \begin{bmatrix} {} \\ \mathsf{hd}(\delta_1 \rho),\, \Lambda \end{bmatrix} & \cdots & \begin{bmatrix}\Psi, G, G, \mathsf{cp}(\delta_h \rho) \\ \Gamma, \mathsf{lp}(\delta_h \rho) \end{bmatrix} \vdash \begin{bmatrix} {} \\ \mathsf{hd}(\delta_h \rho),\, \Lambda \end{bmatrix} \\
& \begin{bmatrix}\Psi, G, G \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} G' \\ \Lambda \end{bmatrix} \arrow[ul, "\mathsf{G_R}"] \arrow[ur, "\mathsf{G_R}"'] & \\
& \begin{bmatrix}\Psi, G \\ \Gamma \end{bmatrix} \vdash \begin{bmatrix} G' \\ \Lambda \end{bmatrix} \arrow[u, ">"] &
\end{tikzcd}$$

where $h > 0$. By the induction hypothesis, for $1 \leqslant i \leqslant h$, there are $>$-free proofs $\overline{\Pi}_i$, corresponding to

$$\begin{tikzcd}
\Pi_i \\
\left[\begin{array}{r}\Psi, G, G, \mathsf{cp}(\delta_i\,\rho)\\ \Gamma, \mathsf{lp}(\delta_i\,\rho)\end{array}\right] \vdash \left[\begin{array}{l} \\ \mathsf{hd}(\delta_i\,\rho), \Lambda\end{array}\right] \\
\left[\begin{array}{r}\Psi, G, \mathsf{cp}(\delta_i\,\rho)\\ \Gamma, \mathsf{lp}(\delta_i\,\rho)\end{array}\right] \vdash \left[\begin{array}{l} \\ \mathsf{hd}(\delta_i\,\rho), \Lambda\end{array}\right] \arrow[u, ">"]
\end{tikzcd} .$$

Then take $\overline{\Pi}$ as

$$\begin{tikzcd}
\overline{\Pi}_1 & & \overline{\Pi}_h \\
\left[\begin{array}{r}\Psi, G, \mathsf{cp}(\delta_1\,\rho)\\ \Gamma, \mathsf{lp}(\delta_1\,\rho)\end{array}\right] \vdash \left[\begin{array}{l} \\ \mathsf{hd}(\delta_1\,\rho), \Lambda\end{array}\right] & \cdots & \left[\begin{array}{r}\Psi, G, \mathsf{cp}(\delta_h\,\rho)\\ \Gamma, \mathsf{lp}(\delta_h\,\rho)\end{array}\right] \vdash \left[\begin{array}{l} \\ \mathsf{hd}(\delta_h\,\rho), \Lambda\end{array}\right] \\
& \left[\begin{array}{r}\Psi, G\\ \Gamma\end{array}\right] \vdash \left[\begin{array}{r}G'\\ \Lambda\end{array}\right] \arrow[ul, "\mathsf{G_R}"] \arrow[ur, "\mathsf{G_R}"'] &
\end{tikzcd} .$$

□

4.3.2. Lemma. *For every proof in* G-Forum$^{>}$ *there is a proof in* G-Forum *whose conclusion is the same.*

Proof. Apply repeatedly Lemma 4.3.1 until all $>$ instances disappear. □

4.4. The cut elimination theorem

4.4.1. Theorem. *For every proof in* G-Forum$^{\bowtie_\mathsf{L}, \bowtie_\mathsf{C}}$ *there exists a proof in* G-Forum *with the same conclusion.*

Proof. G-Forum$^{>, \bowtie_\mathsf{L}, \bowtie'_\mathsf{C}}$ is more general than G-Forum$^{\bowtie_\mathsf{L}, \bowtie_\mathsf{C}}$, so we can proceed by the scheme:

G-Forum$^{>, \bowtie_\mathsf{L}, \bowtie'_\mathsf{C}}$ $\rightarrow$ G-Forum$^{>, \bowtie_\mathsf{L}}$ $\rightarrow$ G-Forum$^{>}$ $\rightarrow$ G-Forum,

by applying successively Lemmas 4.1.2, 4.2.3 and 4.3.2. □

5. Conclusions

In this paper, we showed G-Forum, a non-left-right symmetric sequent system for linear logic, and we proved cut elimination for it. Like in the case of Forum, the left-right asymmetry of sequents is motivated by the necessity of limiting proof search to uniform proofs. We limited the freedom of composing formulae in such a way that their structure matches that of sequents; in a certain sense, formulae become asymmetric, too. We consider this situation where two asymmetries match aesthetically pleasant, and actually more symmetric than the same in Forum, where the structure of formulae is at odds with that of sequents. Our design is motivated by the desire of structuring proofs by easily definable, big building blocks, suitable to semantic understanding.

The result is a system for which it is natural to define cut rules and for which it is possible to prove cut elimination by a procedure that rewrites proofs inside the system, without resorting to Forum or plain linear logic. This guarantees that the new system has a good proof theoretical standing, which usually means that it is a good basis for further, fruitful research. The technique used for proving cut elimination improves on the standard one by a careful separation of bookkeeping phases around a central one directly concerned with the exploitation of the symmetries of the system.

In [9], Miller shows an equivalent but richer syntax for Forum. Even if it does not make a difference for provability, a different syntax has of course a potentially big impact on proof theory. It should be possible to use our techniques and extend our results to that case, but this is not trivial and has not been attempted in this paper.

In a forthcoming paper, we will show how to associate to G-Forum a labelled event structure semantics, i.e., a behavioural model of computation, along the lines initiated in [6]. In another paper, we will apply G-Forum and its semantics to problems of partial-order planning.

The methods in this paper are about studying the structure of proofs at a coarser abstraction level than the one provided by the sequent calculus. In another research project we are pursuing about the calculus of structures [3,7,2,12] (see also http://alessio.guglielmi.name/res/cos), we study proofs at a *finer* level than provided by the sequent calculus. We do so for exploring properties of locality and modularity, which are important for concurrency, that are otherwise not available.

In the future, we plan to adapt the techniques in this paper to the calculus of structures (for example, of linear logic), in order to cover the full range of granularity: from the finer, suitable for distributed implementation, to the coarser, suitable for semantics.

References

[1] J.-M. Andreoli, Logic programming with focusing proofs in linear logic, J. Logic Comput. 2 (3) (1992) 297–347.

[2] K. Brünnler, A.F. Tiu, A local system for classical logic, in: R. Nieuwenhuis, A. Voronkov (Eds.), LPAR 2001, Lecture Notes in Artificial Intelligence, Vol. 2250, Springer, Berlin, 2001, pp. 347–361 URL: ⟨http://www.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf⟩.

[3] P. Bruscoli, A purely logical account of sequentiality in proof search, in: P.J. Stuckey (Ed.), Logic Programming, 18th Internat. Conf., Lecture Notes in Computer Science, Vol. 2401, Springer, Berlin, 2002, pp. 302–316 URL: ⟨http://www.ki.inf.tu-dresden.de/~paola/bvl/bvl.pdf⟩.

[4] J. Gallier, Constructive logics. Part I: a tutorial on proof systems and typed λ-calculi, Theoret. Comput. Sci. 110 (1993) 249–339.

[5] J.-Y. Girard, Linear logic, Theoret. Comput. Sci. 50 (1987) 1–102.

[6] A. Guglielmi, Abstract logic programming in linear logic—independence and causality in a first order calculus, Ph.D. Thesis, Università di Pisa, 1996.

[7] A. Guglielmi, A system of interaction and structure, Technical Report WV-02-10, Technische Universität Dresden, 2002, ACM Trans. Comput. Logic, to appear, URL: ⟨http://www.ki.inf.tu-dresden.de/~guglielm/p/SystIntStr.pdf⟩.

[8] D. Miller, A multiple-conclusion meta-logic, in: S. Abramsky (Ed.), Ninth Annu. IEEE Symp. on Logic in Computer Science, Paris, July 1994, pp. 272–281.

[9] D. Miller, Forum: a multiple-conclusion specification logic, Theoret. Comput. Sci. 165 (1996) 201–232.

[10] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov, Uniform proofs as a foundation for logic programming, Ann. Pure Appl. Logic 51 (1991) 125–157.

[11] G. Nadathur, Uniform provability in classical logic, J. Logic Comput. 8 (2) (1998) 209–229.

[12] L. Straßburger, MELL in the calculus of structures, Theoret. Comput. Sci. 309 (2003) 213–285 URL: ⟨http://www.ps.uni-sb.de/~lutz/papers/els.pdf⟩.