EPTCS 197

Proceedings of the

First International Workshop on

Focusing

Suva, Fiji, 23rd November 2015

Edited by: Iliano Cervesato and Carsten Schürmann

Published: 8th November 2015
DOI: 10.4204/EPTCS.197
ISSN: 2075-2180
Open Publishing Association

Table of Contents

Preface

This volume constitutes the proceedings of WoF'15, the First International Workshop on Focusing, held on November 23rd, 2015 in Suva, Fiji. The workshop was a half-day satellite event of LPAR-20, the 20th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning.

The program committee selected four papers for presentation at WoF'15, and inclusion in this volume. In addition, the program included an invited talk by Elaine Pimentel (UFRN, Brazil).

Focusing is a proof search strategy that alternates two phases: an inversion phase where invertible sequent rules are applied exhaustively and a chaining phase where it selects a formula and decomposes it maximally using non-invertible rules. Focusing is one of the most exciting recent developments in computational logic: it is complete for many logics of interest and provides a foundation for their use as programming languages and rewriting calculi.

This workshop had the purposes of bringing together researchers who work on or with focusing, to foster discussion and to report on recent advances. Topics of interest included: focusing in forward, backward and hybrid logic programming languages, focusing in theorem proving, focusing for substructural logics, focusing for epistemic logics, focused term calculi, implementation techniques, parallelism and concurrency, focusing in security, and pearls of focusing.

Many people helped make WoF'15 a success. We wish to thank the organizers of LPAR-20 for their support. We are indebted to the program committee members and the external referee for their careful and efficient work in the reviewing process. Finally we are grateful to the authors, the invited speaker and the attendees who made this workshop an enjoyable and fruitful event.

November, 2015

Iliano Cervesato
Carsten Schürmann

I. Cervesato and C. Schürmann (Eds.)
First Workshop on Focusing (WoF'15)
EPTCS 197, 2015, pp. ii–iii, doi:10.4204/EPTCS.197.0

Program Committee of WoF'15

- Iliano Cervesato (Carnegie Mellon University — co-chair)
- Kaustuv Chaudhuri (Inria)
- Paul Blain Levy (University of Birmingham)
- Chuck Liang (Hofstra University)
- Elaine Pimentel (Universidade Federal do Rio Grande do Norte)
- Carsten Schürmann (IT University of Copenhagen — co-chair)

Additional Reviewers

Giselle Reis.

Towards the Automated Generation of Focused Proof Systems

Vivek Nigam
Federal University of Paraíba, Brazil
vivek.nigam@gmail.com

Giselle Reis
Inria & LIX, France
giselle.reis@inria.fr

Leonardo Lima
Federal University of Paraíba, Brazil
leonardo.alfs@gmail.com

This paper tackles the problem of formulating and proving the completeness of focused-like proof systems in an automated fashion. Focusing is a discipline on proofs which structures them into phases in order to reduce proof search non-determinism. We demonstrate that it is possible to construct a complete focused proof system from a given un-focused proof system if it satisfies some conditions. Our key idea is to generalize the completeness proof based on permutation lemmas given by Miller and Saurin for the focused linear logic proof system. This is done by building a graph from the rule permutation relation of a proof system, called permutation graph. We then show that from the permutation graph of a given proof system, it is possible to construct a complete focused proof system, and additionally infer for which formulas contraction is admissible. An implementation for building the permutation graph of a system is provided. We apply our technique to generate the focused proof systems MALLF, LJF and LKF for linear, intuitionistic and classical logics, respectively.

1 Introduction

In spite of its widespread use, the proposition and completeness proofs of focused proof systems are still an *ad-hoc* and hard task, done for each individual system separately. For example, the original completeness proof for the focused linear logic proof system (LLF) [1] is very specific to linear logic. The completeness proof for many focused proof systems for intuitionistic logic, such as LJF [5], LKQ and LKT [3], are obtained by using non-trivial encodings of intuitionistic logic in linear logic.

One exception, however, is the work of Miller and Saurin [7], where they propose a modular way to prove the completeness of focused proof systems based on permutation lemmas and proof transformations. They show that a given focused proof system is complete with respect to its unfocused version by demonstrating that any proof in the unfocused system can be transformed into a proof in the focused system. Their proof technique has been successfully adapted to prove the completeness of a number of focused proof systems based on linear logic, such as ELL [11], μMALL [2] and SELLF [8].

This paper proposes a method for the automated generation of a sound and complete focused proof system from a given unfocused sequent calculus proof system. Our approach uses as theoretical foundations the modular proof given by Miller and Saurin [7]. There are, however, a number of challenges in automating such a proof for any given unfocused proof system: (1) Not all proof systems seem to admit a focused version. We define sufficient conditions based on the definitions in [7]; (2) Even if a proof system satisfies such conditions, there are many design choices when formulating a focused version for a system; (3) Miller and Saurin's proof cannot be directly applied to proof systems that have contraction and weakening rules, such as LJ; Focused proof systems, such as LJF and LKF, allow only the contraction of some formulas. This result was obtained by non-trivial encodings in linear logic [6]. Here, we demonstrate that this can be obtained in the system itself, *i.e.*, without a detour through linear logic; (4) Miller and Saurin did not formalize why their procedure or transforming an unfocused proof into a focused one terminates. It already seems challenging to do so for MALL as permutations are not

I. Cervesato and C. Schürmann (Eds.)
First Workshop on Focusing (WoF'15)
EPTCS 197, 2015, pp. 1–6, doi:10.4204/EPTCS.197.1

necessarily size preserving (with respect to the number of inferences). We are still investigating general conditions and this is left to future work.

In order to overcome these challenges, we introduce in Section 3 the notion of permutation graphs. Our previous work [9, 10] showed how to check whether a rule permutes over another in an automated fashion. We use these results to construct the permutation graph of a proof system. This paper then shows that, by analysing the permutation graph of an unfocused proof system, we can construct possibly different focused versions of this system, all sound and complete (provided a proof of termination is given). We sketch in Section 4 how to check the admissibility of contraction rules.

2 Permutation Graphs

In the following we assume that we are given a sequent calculus proof system $\mathbb{S}$ which is commutative, *i.e.*, sequents are formed by multi-sets of formulas, and whose non-atomic initial and cut rules are admissible. We will also assume that whenever contraction is allowed then weakening is also allowed, that is, our systems can be affine, but not relevant. Finally, we assume the reader is familiar with basic proof theory terminology, such as main and auxiliary formulas, formula ancestors.

Definition 1 (Permutability)**.** *Let α and β be two inference rules in a sequent calculus system $\mathbb{S}$. We will say that α* permutes up *β, denoted by $\alpha \uparrow \beta$, if for every $\mathbb{S}$ derivation of a sequent $\mathscr{S}$ in which α operates on $\mathscr{S}$ and β operates on one or more of α's premises (but not on auxiliary formulas of α), there exists another $\mathbb{S}$ derivation of $\mathscr{S}$ in which β operates on $\mathscr{S}$ and α operates on zero or more of β's premises (but not on β's auxiliary formulas). Consequently, β* permutes down *α ($\beta \downarrow \alpha$).*

Note that if there is no derivation in which β operates on α's premises without acting on its auxiliary formulas (e.g., $\vee_r$ and $\wedge_r$ in LJ), the permutation holds vacuously.

Definition 2 (Permutation graph)**.** *Let $\mathscr{R}$ be the set of inference rules of a sequent calculus system $\mathbb{S}$. We construct the (directed)* permutation graph *$P_{\mathbb{S}} = (V,E)$ for $\mathbb{S}$ by taking $V = \mathscr{R}$ and $E = \{(\alpha,\beta) \mid \alpha \uparrow \beta\}$.*

Definition 3 (Permutation cliques)**.** *Let $\mathbb{S}$ be a sequent calculus system and $P_{\mathbb{S}}$ its permutation graph. Consider $P^*_{\mathbb{S}} = (V^*,E^*)$ the undirected graph obtained from $P_{\mathbb{S}} = (V,E)$ by taking $V^* = V$ and $E^* = \{(\alpha,\beta) \mid (\alpha,\beta) \in E$ and $(\beta,\alpha) \in E\}$. Then the* permutation cliques *of $\mathbb{S}$ are the maximal cliques*[1] *of $P^*_{\mathbb{S}}$.*

For LJ, we obtain the following cliques $CL_1 = \{\wedge_l, \vee_l, \to_r, \vee_r\}$ and $CL_2 = \{\wedge_r, \vee_r, \to_l\}$.

Permutation cliques can be thought of as equivalence classes for inference rules. For example, the rule $\wedge_l$ permutes over all rules in CL_1. Permutation cliques are not always disjoint. For example, the rule $\vee_r$ appears in both cliques.

Definition 4 (Permutation partition)**.** *Let $\mathbb{S}$ be a proof system and $P_{\mathbb{S}}$ its permutation graph. Then a* permutation partition *$\mathscr{P}$ is a partition of $P_{\mathbb{S}}$ such that each component is a complete graph. We will call each component of such partitions a* permutation component*, motivated by the fact that inferences in the same component permute over each other.*

It is always possible to find such a partition by taking each component to be one single vertex, but we are mostly interested in bi-partitions.

Although in general cliques are computed in exponential time, it is still feasible to compute them since the permutation graph is usually small. The partitions can be obtained simply by choosing at most one partition to those rules present in more than one clique. Therefore, there might be many possible

[1] A clique in a graph G is a set of vertices such that all vertices are pairwise connected by one edge.

ways to partition the rules of a system. In what follows (Definition 6) we will define which are the partitions that will yield a focused proof system. As we will see, the following partition will lead to LJF, restricted to multiplicative conjunctions: $C_1 = \{\wedge_l, \vee_l, \rightarrow_r\}$ and $C_2 = \{\wedge_r, \vee_r, \rightarrow_l\}$.

Definition 5 (Permutation partition hierarchy)**.** *Let $\mathbb{S}$ be a proof system, $P_{\mathbb{S}}$ its permutation graph and $\mathscr{P} = C_1, ..., C_n$ a permutation partition. We say that $C_i \downarrow C_j$ iff for every inference $\alpha_i \in C_i$ and $\alpha_j \in C_j$ we have that $\alpha_i \downarrow \alpha_j$,* i.e., *$\alpha_j \uparrow \alpha_i$ or equivalently $(\alpha_j, \alpha_i) \in P_{\mathbb{S}}$.*

Notice that the partition hierarchy can be easily computed from the permutation graph. For the partition used above, we have $C_1 \downarrow C_2$.

3 Focused Proof Systems Generation

We derive a focused proof system $\mathbb{S}^f$ from the permutation partitions of a given proof system $\mathbb{S}$ if some conditions are fulfilled. In this section we explain these conditions and prove that the induced focused system is sound and complete with respect to $\mathbb{S}$.

Definition 6 (Focusable permutation partition)**.** *Let $\mathbb{S}$ be a sequent calculus proof system and $C_1, ..., C_n$ a permutation partition of the rules in $\mathbb{S}$. We say that it is a* focusable permutation partition *if:*

- *$n = 2$ and $C_1 \downarrow C_2$;*
- *Every rule in component C_2 has at most one auxiliary formula in each premise;*
- *Every non-unary rule in component C_2 splits the context among the premises (*i.e.*, there is no implicit copying of context formulas on branching rules).*

We call C_1 the negative component and C_2 the positive component following usual terminology from the focusing literature (e.g. [5]) and classify formula occurrences in a proof as negative and positive according to their introduction rules.

Observe that, in contrast to the usual approach, we do not assign polarities to connectives on their own. Therefore the polarity of a formula can change depending on whether it occurs on the right or on the left side of the sequent. As for now, we will only define permutation partitions of logical inference rules. The structural inference rules will be treated separately. In particular, the role of contraction and its relation to the partitions is discussed in Section 4.

The partition $\{C_1, C_2\}$ for LJ is a focusable permutation partition. Interestingly, LJ allows for other focusable partitions, for example: $C_1 = \{\wedge_l, \vee_l, \rightarrow_r, \vee_r\}$ and $C_2 = \{\wedge_r, \rightarrow_l\}$.

We conjecture that all proof systems derived from a focusable permutation partition are sound and complete. It is not our goal here to justify which partition leads to a more suitable focused proof system, as this would depend on the context where the proof system would be used.

Based on the focusable permutation partition, we can define a focused proof system for $\mathbb{S}$. This definition is syntactically different from those usually present in the literature. It will, in particular, force the store and subsequent selection of a negative formula. This extra step is only for the sake of uniformity and clear separation between phases (there will always be a "no phase" state between two phases).

Definition 7 (Focused proof system)**.** *Let $\mathbb{S}$ be a sequent calculus proof system and $C_1 \downarrow C_2$ a focusable permutation partition of the rules in $\mathbb{S}$. Then we can define the* focused *system $\mathbb{S}^f$ in the following way:*

Sequents. *$\mathbb{S}^f$ sequents are of the shape $\Gamma; \Gamma' \vdash^p \Delta; \Delta'$, where $p \in \{+, -, 0\}$ indicates a positive, negative and neutral polarity sequents respectively. We will call Γ' and Δ' the* active *contexts.*

Inference Rules. *For each rule α in $\mathbb{S}$ belonging to the negative (positive) component, $\mathbb{S}^f$ will have a rule α with conclusion and premises being negative (positive) sequents and main and auxiliary formulas occurring in the active contexts.*

Structural rules. *The connection between the phases is done via the following structural rules.*

Selection rules *move a formula F to the active context. If F is negative, then $p = -$. If F is positive, then there is no negative $F' \in \Gamma \cup \Delta$ and $p = +$.* Store rules *remove a formula F from the active context if F is negative and $p = +$ or if F is positive and $p = -$. The* end rule *removes the label $p = \{+, -\}$ of a sequent by setting it to 0 if the active contexts are empty.*

$$\frac{\Gamma; F \vdash^{p} \Delta; \cdot}{\Gamma, F; \cdot \vdash^{0} \Delta; \cdot}\ sel_l \qquad \frac{\Gamma; \cdot \vdash^{p} \Delta; F}{\Gamma; \cdot \vdash^{0} \Delta, F; \cdot}\ sel_r \qquad \frac{\Gamma, F; \Lambda \vdash^{p} \Delta; \Pi}{\Gamma; \Lambda, F \vdash^{p} \Delta; \Pi}\ st_l \qquad \frac{\Gamma; \Lambda \vdash^{p} \Delta, F; \Pi}{\Gamma; \Lambda \vdash^{p} \Delta; \Pi, F}\ st_r \qquad \frac{\Gamma; \cdot \vdash^{0} \Delta; \cdot}{\Gamma; \cdot \vdash^{p} \Delta; \cdot}\ end$$

An $\mathbb{S}^f$ proof is characterized by sequences of inferences labeled with $+$ or $-$ which we will call phases. *Thus, we can say that selection rules are responsible for starting a phase and the end rule finishes a phase. Between any two phases there is always a "neutral" state, denoted by a sequent labeled with 0.*

We can prove using the machinery given in [7] that the focused proof system obtained is complete. There is one catch, however: one also needs to prove that the procedure to convert an unfocused proof into a focused proof using permutation lemmas terminates. This was not formalized in [7], although one can prove it. Finding general conditions is more challenging and is subject of current investigation.

Conjecture 1 (Completeness of focused proof systems)**.** *A sequent $\Gamma \vdash \Delta$ is provable in $\mathbb{S}$ iff the sequent $\Gamma; \cdot \vdash^{0} \Delta; \cdot$ is provable in $\mathbb{S}^f$.*

4 Admissibility of contraction

During proof search, it is desirable to avoid unnecessary copying of formulas at each rule application. Either by not copying the same context in all premises or by not auto-contracting the main formula of a rule application. The analysis of where the contraction rule lies in the permutation cliques can give us insights on when it can be avoided.

Definition 8 (Admissibility of contraction)**.** *Let $\mathbb{S}$ be a sequent calculus system with a set of rules $\mathscr{R}$. We say that contraction is* admissible *for $\mathscr{R}' \subseteq \mathscr{R}$ if for every $\mathbb{S}$ derivation φ there exists an $\mathbb{S}$ derivation φ' such that contraction is never applied to main formulas of inferences in $\mathscr{R}'$.*

The intuitionistic system LJ is an example of a calculus in which contraction is not admissible for all formulas. It is only complete if the main formula of the implication left rule is contracted [4].

The admissibility of contraction involves transformations which are similar to the rank reduction rewriting rules of reductive cut-elimination. This is a special case of permutation checking, since the upper inference *must* be applied to auxiliary formulas of the lower inference.

Definition 9 (Contraction permutation)**.** *Let $\mathbb{S}$ be a sequent calculus proof system, c one of its contraction rules and α a logical rule applied to a formula F_α. We say that $c \uparrow_c \alpha$ if a derivation composed by contraction of F_α followed by applications of α to the contracted formulas can be transformed into a derivation where α is applied to F_α and contraction is applied to the auxiliary formulas of α.*

It is worth noting that many of the cases for contraction permutation rely on α being applied to all contracted formulas in all premises where they occur. The proofs of such cases require a lemma stating that α can be "pushed down" until the correct location.

If $c \uparrow_c \alpha$ for some inference α, then it is admissible for that inference, as it can always be replaced by contraction on its ancestors To prove full admissibility of contraction in the calculus, it is necessary to prove that contraction on atoms can be eliminated. We will not address this issue in this paper, but we will analyse the behavior of contraction among the phases in a focused proof.

$$\frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_1 \& A_2 \vdash \Delta}\ \&_l \quad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\ \otimes_l \quad \frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \oplus B \vdash \Delta}\ \oplus_l \quad \frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \parr B \vdash \Delta, \Delta'}\ \parr_l$$

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \& B}\ \&_r \quad \frac{\Gamma \vdash \Delta, A \quad \Gamma' \vdash \Delta', B}{\Gamma, \Gamma' \vdash \Delta, \Delta', A \otimes B}\ \otimes_r \quad \frac{\Gamma \vdash \Delta, A_i}{\Gamma \vdash \Delta, A_1 \oplus A_2}\ \oplus_r \quad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \parr B}\ \parr_r$$

Figure 1: MALL logical inferences

$$\frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_1 \wedge A_2 \vdash \Delta}\ \wedge^a_l \quad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge^m_l \quad \frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee^a_l \quad \frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'}\ \vee^m_l$$

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\ \wedge^a_r \quad \frac{\Gamma \vdash \Delta, A \quad \Gamma' \vdash \Delta', B}{\Gamma, \Gamma' \vdash \Delta, \Delta', A \wedge B}\ \wedge^m_r \quad \frac{\Gamma \vdash \Delta, A_i}{\Gamma \vdash \Delta, A_1 \vee A_2}\ \vee^a_r \quad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}\ \vee^m_r$$

Figure 2: Additive and multiplicative logical inferences of the LK system.

Definition 10 (Admissibility of contraction in a phase)**.** *Let $\mathbb{S}$ be a sequent calculus proof system and C_1, C_2 a focusable permutation partition. We say that contraction is admissible in phase i if every $\mathbb{S}$ proof can be transformed into a proof where contraction is never applied to main formulas of rules $\alpha \in C_i$.*

Theorem 1. *Let $\mathbb{S}$ be a sequent calculus system, C its contraction rules, C_1, C_2 a focusable permutation partition. If for all $c \in C$ and $\alpha \in C_i$, $c \uparrow \alpha$ and $c \uparrow_c \alpha$, then contraction is admissible in phase i.*

The focused proof is obtained by only contracting formulas that can be introduced by C_2 rules. It is easy to extend Definition 7 to enforce this as done in LJF and LKF [5].

5 Case studies

Given the permutation cliques, it is up to the user to analyse them and decide which partition to use for the focused proof system. As case studies we will show how the focused proof systems LKF, LJF and MALLF can be obtained from LK, LJ and MALL respectively using the permutation cliques.

MALL MALL stands for multiplicative additive linear logic (without exponentials) and its rules are depicted in Figure 1. A focused system, MALLF, for this calculus was proposed in [1].

Given the logical inferences of MALL, the permutation cliques found were the following: $CL_1 = \{\otimes_l, \oplus_l, \parr_r, \&_r, \&_l, \oplus_r\}$ and $CL_2 = \{\otimes_r, \oplus_r, \parr_l, \&_l\}$, with the relation $CL_1 \downarrow CL_2$. The following focusable permutation partition corresponds to MALLF: $C_1 = \{\otimes_l, \oplus_l, \parr_r, \&_r\}$ and $C_2 = \{\otimes_r, \oplus_r, \parr_l, \&_l\}$. Notice that all invertible rules are in C_1, while all positive rules are in C_2 as expected.

LK and LJ In order to derive the focused system LKF for classical logic from LK, all variations of inferences must be considered. We need to take into account the additive and multiplicative versions of each conjunction and disjunction, as depicted in Figure 2. In principle an analysis could be made with the usual presentation of the LK system, but it would certainly not result in LKF. Asserting that we can generate a well-known focused system serves as a validation of our method.

The permutation cliques for the inferences in Figure 2 are: $CL_1 = \{\wedge^a_r, \wedge^m_l, \vee^m_r, \vee^a_l, \wedge^a_l, \vee^a_r\}$ and $CL_2 = \{\wedge^m_r, \wedge^a_l, \vee^a_r, \vee^m_l\}$, where $CL_1 \downarrow CL_2$. Analogous to MALL, we can drop the two last rules from CL_1 and obtain a focusable permutation partition which corresponds to the propositional fragment of LKF.

By analysing the permutation relation of contraction to the rules in the partitions, we observe that it permutes up ($\uparrow$ and $\uparrow_c$) all the inferences in CL_1 $\{\wedge^a_l, \vee^a_r\}$. Therefore, it is admissible in the negative phase. For the positive phase, on the other hand, contraction will not permute up, for example, $\wedge^a_l$. We can thus conclude that such a system must have contraction for positive formulas[2].

[2]This contraction is implicit on the *decide* rule and the positive rules for the usual presentation of LKF.

The case for LJ is completely analogous as that of LK when considering the partition: $CL_1 = \{\wedge_l^m, \wedge_r^a, \vee_l, \rightarrow_r, \wedge_l^a, \vee_r\}$ and $CL_2 = \{\wedge_l^a, \wedge_r^m, \vee_r, \rightarrow_l\}$.

6 Conclusion

This paper proposed a method for automatically devising focused proof systems for sequent calculi. Our aim was to provide a uniform and automated way to obtain the sound and complete systems without using an encoding in linear logic. The main element in our solution is the permutation graph of a sequent calculus system. By using this graph we can separate the inferences into positives and negatives and also reason on the admissibility of contraction. The permutation graph represents the permutation lemmas used in the proof in [7]. We extended the method developed in [7] to handle contraction.

For future work, we plan to apply/extend our technique to other proof systems in order to obtain sensible focused proof systems. There are, however, some more foundational challenges in doing so. We would need to extend the conditions used here for determining whether a partition is focusable. For example, non-commutative and bunched proof systems have even more complicated structural restrictions. It is not even clear how would be the focusing discipline for these proof systems. We expect that our existing machinery may help make some of these decisions by investigating different partitions.

Although we can deduce in which phase the contraction of formulas is admissible, it is still unclear if the position of this rule in the permutation graph can indicate exactly which rules do not admit contraction. We expect to further investigate the permutation graphs of other systems to find out if this and other properties can be discovered.

References

[1] Jean-Marc Andreoli (1992): *Logic Programming with Focusing Proofs in Linear Logic*. *Journal of Logic and Computation* 2(3), pp. 297–347, doi:10.1093/logcom/2.3.297.

[2] David Baelde (2008): *A linear approach to the proof-theory of least and greatest fixed points*. Ph.D. thesis, Ecole Polytechnique.

[3] V. Danos, J.-B. Joinet & H. Schellinx (1995): *LKT and LKQ: sequent calculi for second order logic based upon dual linear decompositions of classical implication*. In: *Advances in Linear Logic*, pp. 211–224, doi:10.1017/CBO9780511629150.011.

[4] Roy Dyckhoff (1992): *Contraction-free sequent calculi for intuitionistic logic*. *Journal of Symbolic Logic* 57(3), pp. 795–807, doi:10.2307/2275431.

[5] Chuck Liang & Dale Miller (2007): *Focusing and Polarization in Intuitionistic Logic*. In: *Computer Science Logic, LNCS* 4646, Springer, pp. 451–465, doi:10.1007/978-3-540-74915-8_34.

[6] Chuck Liang & Dale Miller (2009): *Focusing and Polarization in Linear, Intuitionistic, and Classical Logics*. *Theoretical Computer Science* 410(46), pp. 4747–4768, doi:10.1016/j.tcs.2009.07.041.

[7] Dale Miller & Alexis Saurin (2007): *From proofs to focused proofs: a modular proof of focalization in Linear Logic*. In: *CSL 2007*, LNCS, pp. 405–419, doi:10.1007/978-3-540-74915-8_31.

[8] Vivek Nigam (2009): *Exploiting non-canonicity in the sequent calculus*. Ph.D. thesis, Ecole Polytechnique.

[9] Vivek Nigam, Giselle Reis & Leonardo Lima (2013): *Checking Proof Transformations with ASP*. In: *ICLP (Technical Communications)*, 13.

[10] Vivek Nigam, Giselle Reis & Leonardo Lima (2014): *Quati: An Automated Tool for Proving Permutation Lemmas*. In: *7th IJCAR Proceedings*, pp. 255–261, doi:10.1007/978-3-319-08587-6_18.

[11] Alexis Saurin (2008): *Une étude logique du contrôle (appliquée à la programmation fonctionnelle et logique)*. Ph.D. thesis, Ecole Polytechnique.

Proof Outlines as Proof Certificates: A System Description

Roberto Blanco
Inria & LIX, École Polytechnique

Dale Miller
Inria & LIX, École Polytechnique

We apply the *foundational proof certificate* (FPC) framework to the problem of designing high-level outlines of proofs. The FPC framework provides a means to formally define and check a wide range of proof evidence. A focused proof system is central to this framework and such a proof system provides an interesting approach to proof reconstruction during the process of proof checking (relying on an underlying logic programming implementation). Here, we illustrate how the FPC framework can be used to design proof outlines and then to exploit proof checkers as a means for expanding outlines into fully detailed proofs. In order to validate this approach to proof outlines, we have built the ACheck system that allows us to take a sequence of theorems and apply the proof outline "do the obvious induction and close the proof using previously proved lemmas".

1 Introduction

Inference rules, as used in, say, Frege proofs (a.k.a. Hilbert proofs) are usually greatly restricted by limitations of human psychology and by what skeptics are willing to trust. Typically, checking the application of inference rules involves simple syntactic checks, such as deciding on whether or not two premises have the structure A and $A \supset B$ and the conclusion has the structure B. The introduction of automation into theorem proving has allowed us to engineer inference steps that are more substantial and include both computation and search. In recent years, a number of proof theoretic results allow us to extend that literature from being a study of minuscule inference rules (such as *modus ponens* or Gentzen's introduction rules) to a study of large scale and formally defined "synthetic" inference rules. In this paper, we briefly describe the ACheck system in which we build and check *proof outlines* as combinations of such synthetic rules.

Consider the following inductive specification of addition of natural numbers

```
Define plus : nat -> nat -> nat -> prop by
  plus z N N ;
  plus (s N) M (s P) := plus N M P.
```

where `z` and `s` denote zero and successor, respectively. (Examples will be displayed using the syntax of the Abella theorem prover [2]: this syntax should be familiar to users of other systems, such as Coq.) When this definition is introduced, we should establish several properties immediately, e.g., that the addition relation is determinate and total.

```
Theorem plustotal :
  forall N, nat N -> forall M, nat M -> exists S, plus N M S.

Theorem plusdeterm : forall N, nat N -> forall M, nat M ->
  forall S, plus N M S  -> forall T, plus N M T -> S = T.
```

Anyone familiar with proving such theorems knows that their proofs are simple: basically, the obvious induction leads quickly to a final proof. Of course, if we wish to prove more results about addition, one may need to invent and prove some lemma before simple inductions will work. For example, proving the commutativity of addition makes use of two additional lemmas.

I. Cervesato and C. Schürmann (Eds.)
First Workshop on Focusing (WoF'15)
EPTCS 197, 2015, pp. 7–14, doi:10.4204/EPTCS.197.2

```
Theorem plus0com : forall N, nat N ->  plus N z N.

Theorem plusscom : forall M, nat M -> forall N, nat N ->
                   forall P, plus M N P -> plus M (s N) (s P).

Theorem pluscom : forall N, nat N -> forall M, nat M ->
                  forall S, plus N M S  -> plus M N S.
```

These three lemmas have the same high-level proof outline: use the obvious induction invariant, apply some previously proved lemmas and the inductive hypothesis, and deal with any remaining case analysis.

The fact that many theorems can be proved by using induction-lemmas-cases is well-known and built into existing theorem provers. For example, the waterfall model of the Boyer-Moore prover [6] proves such theorems in a similar fashion (but for inductive definitions of functions). Twelf [11] can often prove that some relations are total and functional using a series of similar steps [12]. The tactics and tacticals of LCF have been used to implement procedures that attempt to find proofs using these steps [13]. Finally, TAC [4] attempts to follow such a procedure as well but in a rather fixed and inflexible fashion.

In this paper, we present an approach to describing the simple rules that can prove a given lemma based on previously proved lemmas. Specifically, we define proof certificates that describe the structure of a proof outline that we expect and then we run a proof checker on that certificate to see if the certificate can be elaborated into a full proof of the lemma.

2 A focused proof system

Consider the two introduction rules in Figure 1. If one attempts to prove sequents by reading these rules from conclusion to premises, then these rules need either information from some external source (e.g., an oracle providing the $i \in \{1,2\}$ or the term t) or some implementation support for non-determinism (e.g., unification and backtracking search).

$$\frac{\Gamma \vdash B_i}{\Gamma \vdash B_1 \vee B_2} \qquad \frac{\Gamma \vdash B[t/x]}{\Gamma \vdash \exists x.B}$$

Figure 1: From the LJ calculus

$$\frac{\Gamma \vdash B_i \Downarrow}{\Gamma \vdash B_1 \vee B_2 \Downarrow} \qquad \frac{\Gamma \vdash B[t/x] \Downarrow}{\Gamma \vdash \exists x.B \Downarrow}$$

Figure 2: Focusing annotations

It is meaningless to use Gentzen's sequent calculus to directly support proof automation. Consider, for example, attempting to prove the sequent $\Gamma \vdash \exists x \exists y[(p\ x\ y) \vee ((q\ x\ y) \vee (r\ x\ y))]$, where Γ contains, say, a hundred formulas. The search for a (cut-free) proof of this sequent can confront the need to choose from among a hundred-and-one introduction rules. If we choose the right-side introduction rule, we will then be left with, again, a hundred-and-one introduction rules to apply to the premise. Thus, reducing this sequent to, say, $\Gamma \vdash (q\ t\ s)$ requires picking one path of choices in a space of 101^4 choices.

Focused proof systems address this explosion by organizing rules into two phases. For example, the rules in Figure 1 are written instead as Figure 2. Here, the formula on which one is introducing a connective is marked with the $\Downarrow$: as a result, reducing the sequent $\Gamma \vdash \exists x \exists y[(p\,x\,y) \vee ((q\,x\,y) \vee (r\,x\,y))] \Downarrow$ to $\Gamma \vdash (q\ t\ s) \Downarrow$ involves only those choices related to the formula marked for focus: no interleaving of other choices needs to be considered.

While the $\Downarrow$ phase involves rules that may not be invertible, the $\Uparrow$ phase involves rules that must be invertible. For example, the left rules for $\vee$ and $\exists$ are invertible and their introduction rule is listed as

$$\frac{\Gamma \Uparrow B_1, \Theta \vdash \mathscr{R} \quad \Gamma \Uparrow B_2, \Theta \vdash \mathscr{R}}{\Gamma \Uparrow B_1 \vee^+ B_2, \Theta \vdash \mathscr{R}} \qquad \frac{\Gamma \Uparrow [y/x]B, \Theta \vdash \mathscr{R}}{\Gamma \Uparrow \exists x.B, \Theta \vdash \mathscr{R}}$$

These rules need no external information (in particular, any new variable y will work in the $\exists$ introduction rule). In these last two rules, the zone between $\Uparrow$ and $\vdash$ contains a *list* of formulas. When there are no more invertible rules that can be applied to that first formula, that formula is moved to (i.e., stored in) the zone written as Γ, using the following *store-left* rule

$$\frac{C, \Gamma \Uparrow \Theta \vdash \mathscr{R}}{\Gamma \Uparrow C, \Theta \vdash \mathscr{R}}\, S_l.$$

Finally, when the zone between the $\Uparrow$ and the $\vdash$ is empty (i.e., all invertible inference rules have been completed), it is time to select a (possibly non-invertible) introduction rule to attempt. For that, we have the two *decide* rules

$$\frac{\Gamma, N \Downarrow N \vdash E}{\Gamma, N \Uparrow \cdot \vdash \cdot \Uparrow E}\, D_l \qquad \text{and} \qquad \frac{\Gamma \vdash P \Downarrow}{\Gamma \Uparrow \cdot \vdash \cdot \Uparrow P}\, D_r.$$

Although we cannot show all focused inference rules, we will present those that deal with the least fixed point operator. Formally speaking, when we define a predicate, such as `plus` in the previous section, we are actually naming a least fixed point expression. In the case of `plus`, that expression is

$$\mu\lambda P \lambda n \lambda m \lambda p.(n = z \wedge m = p) \vee \exists n' \exists p'.[n = (s\ n') \wedge p = (s\ p') \wedge (P\ n'\ m\ p')].$$

For the treatment of least fixed points, we follow the μLJ proof system and its focused variant [1, 3]. The treatment of least fixed point expressions in the $\Uparrow$ phase and the $\Downarrow$ phase is given by the three rules

$$\frac{\mu B \overline{t}, \Gamma \Uparrow \Theta \vdash \mathscr{R}}{\Gamma \Uparrow \mu B \overline{t}, \Theta \vdash \mathscr{R}} \qquad \frac{\Gamma \Uparrow S\overline{t}, \Theta \vdash N \quad \Uparrow BS\overline{x} \vdash S\overline{x}}{\Gamma \Uparrow \mu B \overline{t}, \Theta \vdash N} \qquad \frac{\Gamma \vdash B(\mu B)\overline{t} \Downarrow}{\Gamma \vdash \mu B \overline{t} \Downarrow}$$

Notice that the right introduction rule is just an unfolding of the fixed point. There are two ways to treat the least fixed point on the left: one can either perform a store operation or one can do an induction using, in this case, the invariant S. The right premise of the induction rule shows that S is a prefixed point (i.e., $BS \subseteq S$). In general, supplying an invariant can be tedious so we shall also identify two admissible rules for unfolding (also on the left) and *obvious induction*, meaning that the invariant to use is nothing more than the immediately surrounding sequent as the invariant S. In the case of the obvious induction, the left premise sequent will be trivial.

3 Foundational proof certificates

The main idea behind using the *foundational proof certificate* (FPC) [9] approach to defining proof evidence is that theorem provers output their proof evidence to some persistent memory (e.g., a file) and that independent and trustable proof checkers examine this evidence for validity. In order for such a scheme to work, the semantics of the output proof evidence must be formally defined. The FPC framework provides ways to formally define such proof semantics which is also executable when interpreted on top of a suitable logic engine.

There are four key ingredients to providing such a formal definition and they are all described via their relationship to focused proof systems. In fact, consider the following *augmented* versions of inference rules we have seen in the previous section.

$$\frac{\Xi_1 : \Gamma \vdash B_i \Downarrow \quad \vee_e(\Xi_0, \Xi_1, i)}{\Xi_0 : \Gamma \vdash B_1 \vee B_2 \Downarrow} \qquad \frac{\Xi_1 : \Gamma \vdash B[t/x] \Downarrow \quad \exists_e(\Xi_0, \Xi_1, t)}{\Xi_0 : \Gamma \vdash \exists x.B \Downarrow}$$

These two augmented rules contain two of the four ingredients of an FPC: the schema variable Ξ ranges over terms that denote the actual proof evidence comprising a certificate. The additional premises involve *experts* which are predicates that relate the concluding certificate Ξ_0 to a continuation certificate Ξ_1 and some additional information. The expert predicate for the disjunction can provide an indication of which disjunct to pick and the expert for the existential quantifier can provide an indication of which instance of the quantifier to use. Presumably, these expert predicates are capable of digging into a certificate and extracting such information. It is not required, however, for an expert to definitively extract such information. For example, the disjunction expert might guess both 1 and 2 for i: the proof checker will thus need to handle such non-determinism during the checking of certificates.

Another ingredient of an FPC is illustrated by the augmented inference rules in Figure 3. The store-left (S_l) inference rule is augmented with an extra premise that invokes a *clerk* predicate which is responsible for computing an index l that is associated to the stored formula (here, C). The augmented decide-left (D_l) rule is given an extra premise that uses an expert predicate: that premise can be used to compute the index of the formula that is to be selected for focus. The indexing mechanism does not need to be functional (i.e., different formulas can have the same index) in which case the decide rule must also be interpreted as non-deterministic. In earlier work [9], indexes have been identified with structures as diverse as formula occurrences and de Bruijn numerals. In this paper, the only role planned by indexes will be as the names given to lemmas and to hypotheses (i.e., formulas that are stored on the left using the S_l inference rule).

$$\frac{\Xi_1 : \langle l, C\rangle, \Gamma \Uparrow \Theta \vdash \mathscr{R} \quad \mathit{store}_c(\Xi_0, \Xi_1, l)}{\Xi_0 : \Gamma \Uparrow C, \Theta \vdash \mathscr{R}}\ S_l$$

$$\frac{\Xi_1 : \Gamma, \langle l, N\rangle \Downarrow N \vdash E \quad \mathit{decide}_e(\Xi_0, \Xi_1, l)}{\Xi_0 : \Gamma, \langle l, N\rangle \Uparrow \cdot \vdash \cdot \Uparrow E}\ D_l$$

Figure 3: Two augmented rules

As indicated above, there are essentially three operations that we can perform to treat a least fixed point formula in the left-hand context. In fact, we shall expand these into the following four augmented inference rules.

1. The fixed point can be "frozen" in the sense that the store-left (S_l) rule is applied to it. As a result of such a store operation, the stored occurrence of the fixed point will never be unfolded and will not be the site of an induction. Such a frozen fixed point can only be used later in proof construction within an instance of the initial rule.
2. The fixed point can be unfolded just as it can be on the right-hand side of the sequent. (Unfolding on the left is a consequence of the more general induction rule.) The following augmented inference rule can be used to control when such a fixed point is unfolded.

$$\frac{\Xi_1 : \mathscr{N} \Uparrow B(\mu B)\,\overline{t}, \Gamma \vdash \mathscr{R} \qquad \mathit{unfold}(\Xi_0, \Xi_1)}{\Xi_0 : \mathscr{N} \Uparrow \mu B\overline{t}, \Gamma \vdash \mathscr{R}}$$

3. The most substantial inference rule related to the least fixed point is the induction rule. In its most general form, this inference rule involves proving premises that involve an invariant. A proof certificate term Ξ_0 could include such an invariant explicitly and the following augmented rule could be used to extract and use that invariant.

$$\frac{\Xi_1 : \mathscr{N} \Uparrow S\overline{t}, \Gamma \vdash \mathscr{R} \qquad \Xi_2\,\overline{y} : \mathscr{N} \Uparrow BS\overline{y} \vdash S\overline{y} \qquad \mathit{ind}(\Xi_0, \Xi_1, \Xi_2, S)}{\Xi_0 : \mathscr{N} \Uparrow \mu B\overline{t}, \Gamma \vdash \mathscr{R}}$$

4. In general, it appears that invariants are often complex, large, and tedious structures to build and use. Thus, it is most likely that we need to develop a number of techniques by which invariants

are not built directly but are rather implied by alternative reasoning principles. For example, the Abella theorem prover [2], allows the user to do induction not by explicitly entering an invariant but rather by performing a certain kind of guarded, circular reasoning. In the context of this paper, we consider, however, only one approach to specifying induction and that involves taking the sequent $\mathscr{N} \Uparrow \mu B\bar{t}, \Gamma \vdash \mathscr{R}$ and abstracting out the fixed point expression to yield the "obvious" invariant $\hat{S}$ so that the premise $\mathscr{N} \Uparrow S\bar{t}, \Gamma \vdash \mathscr{R}$ has an easy proof. As a result, only the second premise related to the induction rule needs to be proved. The following augmented rule is used to generate and check whether or not the obvious induction invariant can be used.

$$\frac{\Xi_1\, \bar{y} : \mathscr{N} \Uparrow B\hat{S}\bar{y} \vdash \hat{S}\bar{y} \qquad \mathit{obvious}(\Xi_0, \Xi_1)}{\Xi_0 : \mathscr{N} \Uparrow \mu B\bar{t}, \Gamma \vdash \mathscr{R}}$$

An FPC definition is then a collection of the following: (i) the term constructors for the term encoding proof evidence (the Ξ schema variable above), (ii) the constructors for building indexes, and (iii) the definitions of predicates for describing how the clerks and experts behave in different inference rules. Given these definitions, we then check whether or not a sequent of the form $\Xi : \Gamma \vdash B$ is provable. This latter check can be done using a logic programming engine since such an engine should support both unification and backtracking search (thereby allowing one to have a trade-off between large certificates with a great deal of explicit information and small certificates where details are elided and reconstructed as needed). In our own work, we have used both λProlog [10] and Bedwyr [5] as that logic-based engine.

4 Certificate design

Imagine telling a colleague "The proof of this theorem follows by a simple induction and the three lemmas we just proved." You may or may not be correct in such an assertion since (a) the proposed theorem may not be provable and (b) the simple proof you describe may not exist. In any case, it is clear that there is a rather simple, high-level algorithm to follow that will search for such a proof. In this section, we translate that algorithm into an FPC.

Following the paradigm of focused proof systems for first-order logic, there is a clear, high-level outline to follow for doing proof search for cut-free proofs: first do all invertible inference rules and then, select a formula on which to do a series of non-invertible choices. This latter phase ends when one encounters invertible inference rules again or the proof ends. In the setting we describe here, there are two significant complicating features with which to be concerned.

Treating the induction rule. The invertible ($\Uparrow$) phase is generally a place where no important choices in the search for a proof appear. When dealing with a formula that is a fixed point, however, this is no longer true. As described in the previous section, we treat that fixed point expression either by freezing (see also [1]), unfolding, or using an explicitly supplied invariant or the "obvious" induction.

Lemmas must be invoked. Although the focusing framework can work with any provable formula as a lemma, we shall only consider lemmas that are Horn clauses (for example, the lemmas at the end of Section 2). Consider applying a lemma of the form $\forall\bar{x}[A_1 \supset A_2 \supset A_3]$ in proving the sequent $\Gamma \vdash E$. Since the formulas A_1, A_2, and A_3 are polarized positively, we can design the proof outline (simply by only authorizing fixed points to be frozen during this part of the proof) so that $\Gamma \Downarrow \forall\bar{x}[A_1 \supset A_2 \supset A_3] \vdash E$ is provable if and only if there is a substitution θ for the variables in the list of variables $\bar{x}$ such that θA_1 and θA_2 are in Γ and the sequent $\Gamma, \theta A_3 \vdash E$ is provable. The application of such a lemma is then seen as forward chaining: if the context Γ contains two atoms (frozen fixed points) then add a third.

The main issue that a certificate-as-proof-outline therefore needs to provide is some indication of what lemmas should be used during the construction of a proof. The following collections of supporting lemmas—starting from the least explicit to the most explicit—have been tested within our framework: (i) a bound on the number of lemmas that can be used to finish the proof; (ii) a list of possible lemmas to use in finishing the proof; and (iii) a tree of lemmas, indicating which lemmas are applied following the conjunctive structure of the remaining proof.

5 The proof checker

A direct and natural way to implement the FPC approach to proof checking is to use a logic programming language: by turning the augmented inference rules sideways, one gets logic programming clauses. In this way, the rule's conclusion becomes the head of the clause and the rule's premises become the body of the clause. When proof checking in (either classical or intuitionistic) first-order logic without fixed points, the λProlog programming language [10] is a good choice since its treatment of bindings allows for elegant and effective implementations of first-order quantification in formulas and of eigenvariables in proofs [9]. When the logic itself contains fixed points, as is the case in this paper, λProlog is no longer a good choice: instead, a stronger logic that incorporates aspects of the *closed world assumption* is needed. In particular, the ACheck system has two parts. The first is a theorem prover; we have used Abella since it was easy to modify it for exporting theorems and certificates for use by the second part. The second part is the proof checker, FPCcheck, that verifies the output of a session of the theorem prover, suitably translated. This checker is written in the Bedwyr model checking system [5] and is the new component underlying this particular paper: its code is available from `https://github.com/proofcert/fpccheck`. The documentation at that address explains where to find Bedwyr and our modified Abella system.

To illustrate here the kinds of examples available on the web page, the Abella theory files can have a `ship` command that is followed by a string describing a certificate to use to prove the proposed theorem: the checking of this certificate is shipped to the Bedwyr-based kernel for checking. In this particular case, the `induction` certificate constructor is given three arguments: the first is the maximal number of decides that can be used in the proof and the second and third are bounds on the number of unfoldings in the $\Uparrow$ and $\Downarrow$ phases respectively.

```
Theorem plus0com : forall N, is_nat N ->  plus N zero N.
   ship "(induction␣1␣0␣1)".
Theorem plusscom : forall M, is_nat M -> forall N, is_nat N ->
             forall P, plus M N P -> plus M (succ N) (succ P).
   ship "(induction␣1␣0␣1)".
Theorem pluscom : forall N, is_nat N -> forall M, is_nat M ->
                  forall S, plus N M S  -> plus M N S.
   ship "(induction␣2␣1␣0)".
```

The bound on the number of decide rules (first argument) is also a bound on the number of lemmas that can be used on any given branch of the proof.

6 System architecture

Our system for producing and checking proof outlines follows a simple linear work-flow: first, state a theorem and obtain a proof outline, next, attempt to check the theorem with the outline given as a proof

certificate. We have structured this process in three computation steps, involving a theorem prover, a translator, and a proof checker. Their roles are given here.

The theorem prover. An existing theorem prover is principally the source of the concrete syntax of definitions and theorems. It may not be directly involved in the work-flow, particularly if the proof language is extended to support `ship` tactics that enable the omission of detailed proof scripts. At the end of the phase a *theorem file* with all relevant definitions, theorem statements and proof scripts is obtained.

The translator. For each theorem prover, we need to furnish a translator that can convert the concrete syntax of the theorem prover into that of the proof checker. It may export explicit certificates given by the `ship` tactic or derive certificates automatically, possibly making use of proof scripts and execution traces in the theorem file. These translation facilities may be built into the theorem prover itself, or an instrumented version of it, thus encapsulating the first two stages. The translator outputs translation files usable directly by a general-purpose, universal proof checker.

The proof checker. The proof checker, as described in Section 5, implements a focused version of the μLJ logic in the Bedwyr model checking system, augmented to further implement the FPC framework. The translated theorems and their certificates plug into this kernel and constitute a full instantiation of the system, which Bedwyr can execute to verify that the certificates can be elaborated into complete proofs of their associated theorems.

Translators are the only addition needed to enable a new theorem prover to use the FPC framework. Such translators are free to implement sophisticated mechanisms to produce efficient certificates from proof scripts but, in fact, only a more shallow syntactic translation is strictly required: in this latter case, sophistication must be built into the FPC definitions. For the present study with the Abella interactive theorem prover, the translator has been integrated in an instrumented version of Abella, a system whose proximity with Bedwyr syntax is naturally well-suited to the approach.

The work-flow structure just described makes no explicit reference to the FPCs that could be shipped, constructed, and checked. These can conform to the definition of proof outlines used throughout this paper or be tailored to each specific problem. The translator module can choose to use proof outlines as the default, as is the case in our examples. It could also let a user of `ship` specify an FPC definition to be included in the resulting translation, or generate tentative certificates with rules involving other FPC definitions.

7 Conclusion

We have described a methodology for elaborating proof outlines using a framework for checking proof certificates. We have illustrated this approach with “simple inductive proofs” that are applicable in a wide range of situations involving inductively defined datatypes. We plan to scale and study significantly more complex examples in the near future. Finally, it would be interesting to see how our use of high-level descriptions of proofs and proof reconstruction might be related to the work of Bundy and his colleagues on proof plans and rippling [8, 7].

Acknowledgments: We thank anonymous reviewers and K. Chaudhuri for comments on a draft of this paper. This work was funded by the ERC Advanced Grant ProofCert.

References

[1] David Baelde (2012): *Least and greatest fixed points in linear logic*. *ACM Trans. on Computational Logic* 13(1), doi:10.1145/2071368.2071370. Available at http://tocl.acm.org/accepted/427baelde.pdf.

[2] David Baelde, Kaustuv Chaudhuri, Andrew Gacek, Dale Miller, Gopalan Nadathur, Alwen Tiu & Yuting Wang (2014): *Abella: A System for Reasoning about Relational Specifications*. *Journal of Formalized Reasoning* 7(2), doi:10.6092/issn.1972-5787/4650. Available at http://jfr.unibo.it/article/download/4650/4137.

[3] David Baelde & Dale Miller (2007): *Least and greatest fixed points in linear logic*. In N. Dershowitz & A. Voronkov, editors: *International Conference on Logic for Programming and Automated Reasoning (LPAR), LNCS* 4790, pp. 92–106. Available at http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lpar07final.pdf.

[4] David Baelde, Dale Miller & Zachary Snow (2010): *Focused Inductive Theorem Proving*. In J. Giesl & R. Hähnle, editors: *Fifth International Joint Conference on Automated Reasoning, LNCS* 6173, pp. 278–292, doi:10.1007/978-3-642-14203-1_24. Available at http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/ijcar10.pdf.

[5] (2015): *The Bedwyr system*. Available at http://slimmer.gforge.inria.fr/bedwyr/.

[6] Robert S. Boyer & J. Strother Moore (1979): *A Computational Logic*. Academic Press.

[7] A. Bundy, A. Stevens, F. van Harmelen, A. Ireland & A. Smaill (1993): *Rippling: A Heuristic for Guiding Inductive Proofs*. *Artificial Intelligence* 62(2), pp. 185–253, doi:10.1016/0004-3702(93)90079-Q.

[8] Alan Bundy (1987): *The use of explicit plans to guide inductive proofs*. In: *Conf. on Automated Deduction (CADE 9)*, pp. 111–120.

[9] Zakaria Chihani, Dale Miller & Fabien Renaud (2013): *Foundational proof certificates in first-order logic*. In Maria Paola Bonacina, editor: *CADE 24: Conference on Automated Deduction 2013, LNAI* 7898, pp. 162–177, doi:10.1007/978-3-642-38574-2_11.

[10] Dale Miller & Gopalan Nadathur (2012): *Programming with Higher-Order Logic*. Cambridge University Press, doi:10.1017/CBO9781139021326.

[11] Frank Pfenning & Carsten Schürmann (1999): *System Description: Twelf — A Meta-Logical Framework for Deductive Systems*. In H. Ganzinger, editor: *16th Conf. on Automated Deduction (CADE), LNAI* 1632, Springer, Trento, pp. 202–206, doi:10.1007/3-540-48660-7_14.

[12] Carsten Schürmann & Frank Pfenning (2003): *A Coverage Checking Algorithm for LF*. In: *Theorem Proving in Higher Order Logics: 16th International Conference, TPHOLs 2003, LNCS* 2758, Springer, pp. 120–135, doi:10.1007/10930755_8.

[13] Sean Wilson, Jacques Fleuriot & Alan Smaill (2010): *Inductive Proof Automation for Coq*. In: *Second Coq Workshop*. Available at http://hal.archives-ouvertes.fr/inria-00489496/en/.

Realisability semantics of abstract focussing, formalised

Stéphane Graham-Lengrand
CNRS, École Polytechnique, INRIA, SRI International

We present a sequent calculus for abstract focussing, equipped with proof-terms: in the tradition of Zeilberger's work, logical connectives and their introduction rules are left as a parameter of the system, which collapses the synchronous and asynchronous phases of focussing as macro rules. We go further by leaving as a parameter the operation that extends a context of hypotheses with new ones, which allows us to capture both classical and intuitionistic focussed sequent calculi.

We then define the realisability semantics of (the proofs of) the system, on the basis of Munch-Maccagnoni's orthogonality models for the classical focussed sequent calculus, but now operating at the higher level of abstraction mentioned above. We prove, at that level, the Adequacy Lemma, namely that if a term is of type A, then in the model its denotation is in the (set-theoretic) interpretation of A. This exhibits the fact that the universal quantification involved when taking the orthogonal of a set, reflects in the semantics Zeilberger's universal quantification in the macro rule for the asynchronous phase.

The system and its semantics are all formalised in Coq.

1 Introduction

The objective of this paper is to formalise a strong connection between *focussing* and *realisability*.

Focussing is a concept from proof theory that arose from the study of Linear Logic [Gir87, And92] with motivations in proof-search and logic programming, and was then used for studying the proof theory of classical and intuitionistic logics [Gir91, DJS95, DL07, LM09].

Realisability is a concept used since Kleene [Kle45] to witness provability of formulae and build models of their proofs. While originally introduced in the context of constructive logics, the methodology received a renewed attention with the concept of *orthogonality*, used by Girard to build models of Linear Logic proofs, and then used to define the realisability semantics of classical proofs [DK00].

Both focussing and realisability exploit the notion of *polarity* for formulae, with an asymmetric treatment of positive and negative formulae.

In realisability, a primitive interpretation of a positive formula such as $\exists xA$ (resp. $A_1 \vee A_2$) is given as a set of pairs (t,π), where t is a witness of existence and π is in the interpretation of $\{^t/_x\}A$ (resp. a set of injections $\mathsf{inj}_i(\pi)$, where π is in the interpretation of A_i). In other words, the primitive interpretation of positive formulae expresses a very contructive approach to provability. In contrast, a negative formula (such as $\forall xA$ or $A_1 \Rightarrow A_2$) is interpreted "by duality", as the set that is *orthogonal* to the interpretation of its (positive) negation. In classical realisability, a bi-orthogonal set completion provides denotations for all classical proofs of positive formulae.

In focussing, introduction rules for connectives of the same polarity can be chained without loss of generality: for instance if we decide to prove $(A_1 \vee A_2) \vee A_3$ by proving $A_1 \vee A_2$, then we can assume (without losing completeness) that a proof of this in turn consists in a proof of A_1 or a proof of A_2 (rather than uses a hypothesis, uses a lemma or reasons by contradiction).

I. Cervesato and C. Schürmann (Eds.)
First Workshop on Focusing (WoF'15)
EPTCS 197, 2015, pp. 15–28, doi:10.4204/EPTCS.197.3

Such a grouping of introduction steps for positives (resp. negatives) is called a *synchronous* phase (resp. *asynchronous* phase). In order to express those phases as (or collapse them into) single *macro steps*, some formalisms for *big-step focussing* (as in Zeilberger's work [Zei08a, Zei08b]) abstract away from logical connectives and simply take as a parameter the mechanism by which a positive formula can be decomposed into a collection of positive atoms and negative formulae. While the proof of a positive needs to exhibit such a decomposition, the proof of a negative needs to range over such decompositions and perform a case analysis.

This asymmetry, and this universal quantification over decompositions, are reminiscent of the orthogonal construction of realisability models. To make the connection precise, we formalise the construction of realisability models for (big-step) focussed systems in Zeilberger's style.

In [MM09], the classical realisability semantics was studied for the classical sequent calculus, with an extended notion of cut-elimination that produced focussed proofs. Here we want to avoid relying on the specificities of particular logical connectives, and therefore lift this realisability semantics to the more abstract level of Zeilberger's systems, whose big-step approach also reveals the universal quantification that we want to connect to the orthogonality construction. We even avoid relying on the specificities of a particular logic as follows:

- We do not assume that formulae are syntax, i.e. have an inductive structure, nor do we assume that "positive atoms" are particular kinds of formulae; positive atoms and formulae can now literally be two arbitrary sets, of elements called *atoms* and *molecules*, respectively.
- The operation that extends a context of hypotheses with new ones, is usually taken to be set or multiset union. We leave this extension operation as another parameter of the system, since tweaking it will allow the capture of different logics.

In Section 2 we present our abstract system called LAF, seen through the Curry-Howard correspondence as a typing system for (proof-)terms. Section 3 describes how to tune (i.e. instantiate) the notions of atoms, molecules, and context extensions, so as to capture the big-step versions of standard focussed sequent calculi, both classical and intuitionistic. Section 4 gives the definition of realisability models, where terms and types are interpreted, and proves the Adequacy Lemma. In very generic terms, if t is a proof(-term) for A then in the model the interpretation of t is in the interpretation of A. Finally, Section 5 exhibits a simple model from which we derive the consistency of LAF.

2 The abstract focussed sequent calculus LAF

An instance of LAF is given by a tuple of parameters $(\mathsf{Lab}_+, \mathsf{Lab}_-, \mathbb{A}, \mathbb{M}, \mathsf{Co}, \mathsf{Pat}, \Vdash)$ where each parameter is described below. We use $\to$ (resp. $\rightharpoonup$) for the total (resp. partial) function space constructor.

Since our abstract focussing system is a typing system, we use a notion of typing contexts, i.e. those structures denoted Γ in a typing judgement of the form $\Gamma \vdash \ldots$. Two kinds of "types" are used (namely atoms and molecules), and what is declared as having such types in a typing context, is two corresponding kinds of labels:[1] *positive labels* and *negative labels*, respectively ranging over Lab_+ and Lab_-. These two sets are the first two parameters of LAF.

[1] We choose to call them "labels", rather than "variables", because "variable" suggests an object identified by a name that "does not matter" and somewhere subject to α-conversion. Our labels form a deterministic way of indexing atoms and molecules in a context, and could accommodate De Bruijn's indices or De Bruijn's levels.

DEFINITION 1 (Generic contexts, generic decomposition algebras)

Given two sets $\mathscr{A}$ and $\mathscr{B}$, an $(\mathscr{A},\mathscr{B})$-*context algebra* is an algebra of the form

$$\left(\mathscr{G}, \begin{pmatrix} \mathscr{G}\times \mathsf{Lab}_+ \rightharpoonup \mathscr{A} \\ (\Gamma, x^+) \mapsto \Gamma\left[x^+\right] \end{pmatrix}, \begin{pmatrix} \mathscr{G}\times \mathsf{Lab}_- \rightharpoonup \mathscr{B} \\ (\Gamma, x^-) \mapsto \Gamma\left[x^-\right] \end{pmatrix}, \begin{pmatrix} \mathscr{G}\times \mathbb{D}_{\mathscr{A},\mathscr{B}} \to \mathscr{G} \\ (\Gamma,\Delta) \mapsto \Gamma;\Delta \end{pmatrix}\right)$$

whose elements are called $(\mathscr{A},\mathscr{B})$-*contexts*, and where $\mathbb{D}_{\mathscr{A},\mathscr{B}}$, which we call the $(\mathscr{A},\mathscr{B})$-*decomposition algebra* and whose elements are called $(\mathscr{A},\mathscr{B})$-*decompositions*, is the free algebra defined by the following grammar:

$$\Delta, \Delta_1, \ldots ::= \; a \,\big|\, {\sim}b \,\big|\, \bullet \,\big|\, \Delta_1, \Delta_2$$

where a (resp. b) ranges over $\mathscr{A}$ (resp. $\mathscr{B}$).

Let $\mathbb{D}_{\mathsf{st}}$ abbreviate $\mathbb{D}_{\mathtt{unit},\mathtt{unit}}$, whose elements we call *decomposition structures*.

The *(decomposition) structure* of an $(\mathscr{A},\mathscr{B})$-decomposition Δ, denoted $|\Delta|$, is its obvious homomorphic projection in $\mathbb{D}_{\mathsf{st}}$.

We denote by $\mathsf{dom}^+(\Gamma)$ (resp. $\mathsf{dom}^-(\Gamma)$) the subset of Lab_+ (resp. Lab_-) where $\Gamma[x^+]$ (resp. $\Gamma[x^-]$) is defined, and say that Γ is *empty* if both $\mathsf{dom}^+(\Gamma)$ and $\mathsf{dom}^-(\Gamma)$ are.

Intuitively, an $(\mathscr{A},\mathscr{B})$-decomposition Δ is simply the packaging of elements of $\mathscr{A}$ and elements of $\mathscr{B}$; we could flatten this packaging by seeing $\bullet$ as the empty set (or multiset), and Δ_1,Δ_2 as the union of the two sets (or multisets) Δ_1 and Δ_2. Note that the coercion from $\mathscr{B}$ into $\mathbb{D}_{\mathscr{A},\mathscr{B}}$ is denoted with $\sim$. It helps distinguishing it from the coercion from $\mathscr{A}$ (e.g. when $\mathscr{A}$ and $\mathscr{B}$ intersect each other), and in many instances of LAF it will remind us of the presence of an otherwise implicit negation. But so far it has no logical meaning, and in particular $\mathscr{B}$ is not equipped with an operator $\sim$ of syntactical or semantical nature.

Now we can specify the nature of the LAF parameters:

DEFINITION 2 (LAF parameters) Besides Lab_+ and Lab_-, LAF is parameterised by

- two sets $\mathbb{A}$ and $\mathbb{M}$, whose elements are respectively called *atoms* (denoted a, a',...) and *molecules* (denoted M, M',...);
- an $(\mathbb{A},\mathbb{M})$-context algebra Co, whose elements are called *typing contexts*;
- a *pattern algebra*, an algebra of the form

$$\left(\mathsf{Pat}, \begin{pmatrix} \mathsf{Pat}\to\mathbb{D}_{\mathsf{st}} \\ p \mapsto |p| \end{pmatrix}\right)$$

  whose elements are called *patterns*,
- a *decomposition relation*, i.e. a set of elements

$$(\_ \Vdash \_ : \_) : (\mathbb{D} \times \mathsf{Pat} \times \mathbb{M})$$

  such that if $\Delta \Vdash p : M$ then the structure of Δ is $|p|$.

The $(\mathbb{A},\mathbb{M})$-decomposition algebra, whose elements are called *typing decompositions*, is called the *typing decomposition algebra* and is denoted $\mathbb{D}$.

The group of parameters $(\mathbb{A},\mathbb{M})$ specifies what the instance of LAF, as a logical system, talks about. A typical example is when $\mathbb{A}$ and $\mathbb{M}$ are respectively the sets of (positive) atoms and the set of (positive) formulae from a polarised logic. Section 3 shows how our level of abstraction allows for some interesting variants. In the Curry-Howard view, $\mathbb{A}$ and $\mathbb{M}$ are our sets of types.

The last group of parameters $(\mathsf{Pat}, \Vdash)$ specifies the structure of molecules. If $\mathbb{M}$ is a set of formulae featuring logical connectives, those parameters specify the introduction rules for the connectives. The

intuition behind the terminology is that the decomposition relation $\Vdash$ decomposes a molecule, according to a pattern, into a typing decomposition which, as a first approximation, can be seen as a "collection of atoms and (hopefully smaller) molecules".

DEFINITION 3 (LAF system) Proof-terms are defined by the following syntax:

Positive terms	Terms^+	$t^+ ::= pd$
Decomposition terms	$\mathsf{Terms}^{\mathsf{d}}$	$d ::= x^+ \mid f \mid \bullet \mid d_1, d_2$
Commands	Terms	$c ::= \langle x^- \mid t^+ \rangle \mid \langle f \mid t^+ \rangle$

where p ranges over Pat, x^+ ranges over Lab_+, x^- ranges over Lab_-, and f ranges over the partial function space $\mathsf{Pat} \rightharpoonup \mathsf{Terms}$.

LAF is the inference system of Fig. 1 defining the derivability of three kinds of sequents

$$\begin{array}{lcl} (\_ \vdash [\_ : \_]) & : & (\mathsf{Co} \times \mathsf{Terms}^+ \times \mathbb{M}) \\ (\_ \vdash \_ : \_) & : & (\mathsf{Co} \times \mathsf{Terms}^{\mathsf{d}} \times \mathbb{D}) \\ (\_ \vdash \_) & : & (\mathsf{Co} \times \mathsf{Terms}) \end{array}$$

We further impose in rule async that the domain of function f be exactly those patterns that can decompose M ($p \in \mathsf{Dom}(f)$ if and only if there exists Δ such that $\Delta \Vdash p : M$).

$\mathsf{LAF}^{\mathsf{cf}}$ is the inference system LAF without the cut-rule.

$$\frac{\Delta \Vdash p : M \quad \Gamma \vdash d : \Delta}{\Gamma \vdash [pd : M]}\ \mathsf{sync}$$

$$\frac{}{\Gamma \vdash \bullet : \bullet} \qquad \frac{\Gamma \vdash d_1 : \Delta_1 \quad \Gamma \vdash d_2 : \Delta_2}{\Gamma \vdash d_1, d_2 : \Delta_1, \Delta_2}$$

$$\frac{\Gamma\left[x^+\right] = a}{\Gamma \vdash x^+ : a}\ \mathsf{init} \qquad \frac{\forall p, \forall \Delta, \quad \Delta \Vdash p : M \quad \Rightarrow \quad \Gamma; \Delta \vdash f(p)}{\Gamma \vdash f : \sim M}\ \mathsf{async}$$

$$\frac{\Gamma \vdash \left[t^+ : \Gamma\left[x^-\right]\right]}{\Gamma \vdash \left\langle x^- \mid t^+ \right\rangle}\ \mathsf{select} \qquad \frac{\Gamma \vdash f : \sim M \qquad \Gamma \vdash [t^+ : M]}{\Gamma \vdash \left\langle f \mid t^+ \right\rangle}\ \mathsf{cut}$$

Figure 1: LAF

An intuition of LAF can be given in terms of proof-search:

When we want to "prove" a molecule, we first need to decompose it into a collection of atoms and (refutations of) molecules (rule sync). Each of those atoms must be found in the current typing context (rule init). Each of those molecules must be refuted, and the way to do this is to consider all the possible ways that this molecule could be decomposed, and for each of those decompositions, prove the inconsistency of the current typing context extended with the decomposition (rule async). This can be done by proving one of the molecules refuted in the typing context (rule select) or refuted by a complex proof (rule cut). Then a new cycle starts.

Now it will be useful to formalise the idea that, when a molecule M is decomposed into a collection of atoms and (refutations of) molecules, the latter are "smaller" than M:

DEFINITION 4 (Well-founded LAF instance)

We write $M' \triangleleft M$ if there are Δ and p such that $\Delta \Vdash p : M$ and $\sim M'$ is a leaf of Δ.

The LAF instance is *well-founded* if (the smallest order containing) $\triangleleft$ is well-founded.

Well-foundedness is a property that a LAF instance may or may not have, and which we will require to construct its realisability semantics. A typical situation where it holds is when $M' \triangleleft M$ implies that M' is a sub-formula of M.

The above intuitions may become clearer when we instantiate the parameters of LAF with actual literals, formulae, etc in order to capture existing systems: we shall therefore we illustrate system LAF by specifying different instances, providing each time the long list of parameters, that capture different focussed sequent calculus systems.

While LAF is defined as a typing system (in other words with proof-terms decorating proofs in the view of the Curry-Howard correspondence), the traditional systems that we capture below are purely logical, with no proof-term decorations. When encoding the former into the latter, we therefore need to erase proof-term annotation, and for this it is useful to project the notion of typing context as follows:

DEFINITION 5 (Referable atoms and molecules) Given a typing context Γ, let $\mathsf{Im}^+(\Gamma)$ (resp. $\mathsf{Im}^-(\Gamma)$) be the image of function $x^+ \mapsto \Gamma[x^+]$ (resp. $x^- \mapsto \Gamma[x^-]$), i.e. the set of atoms (resp. molecules) that can be refered to, in Γ, by the use of a positive (resp. negative) label.

3 Examples in propositional logic

The parameters of LAF will be specified so as to capture: the classical focussed sequent calculus LKF and the intuitionistic one LJF [LM09].

3.1 Polarised classical logic - one-sided

In this sub-section we define the instance LAF_{K1} corresponding to the (one-sided) calculus LKF:

DEFINITION 6 (Literals, formulae, patterns, decomposition)

Let $\mathbb{A}$ be a set of elements called *atoms* and ranged over by $a, a', \ldots$.

Negations of atoms $a^\perp, a'^\perp, \ldots$ range over a set isomorphic to, but disjoint from, $\mathbb{A}$.

Let $\mathbb{M}$ be the set defined by the first line of the following grammar for (polarised) formulae of classical logic:

$$\begin{array}{lll} \text{Positive formulae} & P, \ldots & ::= a \mid \top^+ \mid \perp^+ \mid A \wedge^+ B \mid A \vee^+ B \\ \text{Negative formulae} & N, \ldots & ::= a^\perp \mid \top^- \mid \perp^- \mid A \wedge^- B \mid A \vee^- B \\ \text{Unspecified formulae} & A & ::= P \mid N \end{array}$$

Negation is extended to formulae as usual by De Morgan's laws (see e.g. [LM09]).

The set Pat of *pattern* is defined by the following grammar:

$$p, p_1, p_2, \ldots ::= \_^+ \mid \_^- \mid \bullet \mid (p_1, p_2) \mid \mathsf{inj}_i(p)$$

The decomposition relation $(\_ \Vdash \_ : \_) : (\mathbb{D} \times \mathsf{Pat} \times \mathbb{M})$ is the restriction to molecules of the relation defined inductively for all formulae by the inference system of Fig. 2.

The map $p \mapsto |p|$ can be inferred from the decomposition relation.

$$\frac{}{\bullet \Vdash \bullet : \top^+} \qquad \frac{}{\sim N^\perp \Vdash \_^- : N} \qquad \frac{}{a \Vdash \_^+ : a}$$

$$\frac{\Delta_1 \Vdash p_1 : A_1 \quad \Delta_2 \Vdash p_2 : A_2}{\Delta_1, \Delta_2 \Vdash (p_1, p_2) : A_1 \wedge^+ A_2} \qquad \frac{\Delta \Vdash p : A_i}{\Delta \Vdash \mathsf{inj}_i(p) : A_1 \vee^+ A_2}$$

Figure 2: Decomposition relation for LAF_{K1}

Keeping the sync rule of LAF_{K1} in mind, we can already see in Fig. 2 the traditional introduction rules of positive connectives in polarised classical logic. Note that these rules make LAF_{K1} a well-founded LAF instance, since $M' \triangleleft M$ implies that M' is a sub-formula of M. The rest of this sub-section formalises that intuition and explains how LAF_{K1} manages the introduction of negative connectives, etc. But in order to finish the instantiation of LAF capturing LKF, we need to define typing contexts, i.e. give Lab_+, Lab_-, and Co. In particular, we have to decide how to refer to elements of the typing context. To avoid getting into aspects that may be considered as implementation details (in [GL14a] we present two implementations based on De Bruijn's indices and De Bruijn's levels), we will stay rather generic and only assume the following property:

DEFINITION 7 (Typing contexts) We assume that context extensions satisfy:

$$\begin{array}{llll} \mathsf{Im}^+(\Gamma; a) & = \mathsf{Im}^+(\Gamma) \cup \{a\} & \mathsf{Im}^-(\Gamma; a) & = \mathsf{Im}^-(\Gamma) \\ \mathsf{Im}^+(\Gamma; \sim M) & = \mathsf{Im}^+(\Gamma) & \mathsf{Im}^-(\Gamma; \sim M) & = \mathsf{Im}^-(\Gamma) \cup \{M\} \\ \mathsf{Im}^\pm(\Gamma; \bullet) & = \mathsf{Im}^\pm(\Gamma) & \mathsf{Im}^\pm(\Gamma; (\Delta_1, \Delta_2)) & = \mathsf{Im}^\pm(\Gamma; \Delta_1; \Delta_2) \end{array}$$

where $\pm$ stands for either $+$ or $-$.

We now relate (cut-free) $\mathsf{LAF}^{\mathsf{cf}}_{K1}$ and the LKF system of [LM09] by mapping sequents:

DEFINITION 8 (Mapping sequents)

We encode the sequents of LAF_{K1} (regardless of derivability) to those of LKF as follows:

$$\begin{array}{lll} \phi(\Gamma \vdash c) & := & \vdash \mathsf{Im}^+(\Gamma)^\perp, \mathsf{Im}^-(\Gamma) \Uparrow \\ \phi(\Gamma \vdash x^+ : a) & := & \vdash \mathsf{Im}^+(\Gamma)^\perp, \mathsf{Im}^-(\Gamma) \Downarrow a \\ \phi(\Gamma \vdash f : \sim P) & := & \vdash \mathsf{Im}^+(\Gamma)^\perp, \mathsf{Im}^-(\Gamma) \Downarrow P^\perp \\ \phi(\Gamma \vdash [t^+ : P]) & := & \vdash \mathsf{Im}^+(\Gamma)^\perp, \mathsf{Im}^-(\Gamma) \Downarrow P \end{array}$$

THEOREM 1 ϕ satisfies structural adequacy between $\mathsf{LAF}^{\mathsf{cf}}_{K1}$ and LKF.

The precise notion of adequacy used here is formalised in [GL14a]; let us just say here that it preserves the derivability of sequents in a compositional way (a derivation π in one system is mapped to a derivation π' in the other system, and its subderivations are mapped to subderivations of π').

3.2 Polarised intuitionistic logic

In this sub-section we define the instance LAF_J corresponding to the (two-sided) calculus LJF. Because it it two-sided, and the LAF framework itself does not formalise the notion of side (it is not incorrect to see LAF as being intrinsically one-sided), we shall embed a *side information* in the notions of atoms and molecules:

DEFINITION 9 (Literals, formulae, patterns, decomposition)

Let $\mathbb{L}^+$ (resp. $\mathbb{L}^-$) be a set of elements called positive (resp. negative) literals, and ranged over by $l^+, l_1^+, l_2^+, \ldots$ (resp. $l^-, l_1^-, l_2^-, \ldots$). Formulae are defined by the following grammar:

$$\begin{array}{lll} \text{Positive formulae} & P,\ldots & ::= l^+ \,|\, \top^+ \,|\, \bot^+ \,|\, A \wedge^+ B \,|\, A \vee B \\ \text{Negative formulae} & N,\ldots & ::= l^- \,|\, \top^- \,|\, \bot^- \,|\, A \wedge^- B \,|\, A \Rightarrow B \,|\, \neg A \\ \text{Unspecified formulae} & A & ::= P \,|\, N \end{array}$$

We *position* a literal or a formula on the left-hand side or the right-hand side of a sequent by combining it with an element, called *side information*, of the set $\{\mathsf{l}, \mathsf{r}\}$: we define

$$\begin{array}{lll} \mathbb{A} & := & \{(l^+, \mathsf{r}) \,|\, l^+ \text{ positive literal}\} \cup \{(l^-, \mathsf{l}) \,|\, l^- \text{ negative literal}\} \cup \{(\bot^-, \mathsf{l})\} \\ \mathbb{M} & := & \{(P, \mathsf{r}) \,|\, P \text{ positive formula}\} \cup \{(N, \mathsf{l}) \,|\, N \text{ negative formula}\} \end{array}$$

In the rest of this sub-section v stands for either a negative literal l^- or $\bot^-$.

The set Pat of *pattern* is defined by the following grammar:

$$\begin{array}{lll} p, p_1, p_2, \ldots & ::= & {}_{-\mathsf{r}}^{+} \,|\, {}_{-\mathsf{r}}^{-} \,|\, \bullet_{\mathsf{r}} \,|\, (p_1, p_2) \,|\, \mathsf{inj}_i(p) \\ & | & {}_{-\mathsf{l}}^{+} \,|\, {}_{-\mathsf{l}}^{-} \,|\, \bullet_{\mathsf{l}} \,|\, p_1 :: p_2 \,|\, \pi_i(p) \,|\, \curvearrowright(p) \end{array}$$

The decomposition relation $(\_ \Vdash \_ : \_) : (\mathbb{D} \times \mathsf{Pat} \times \mathbb{M})$ is the restriction to molecules of the relation defined inductively for all positioned formulae by the inference system of Fig. 3.

$$\frac{}{\sim(N,\mathsf{l}) \Vdash {}_{-\mathsf{r}}^{-} : (N,\mathsf{r})} \qquad \frac{}{(l^+,\mathsf{r}) \Vdash {}_{-\mathsf{r}}^{+} : (l^+,\mathsf{r})}$$

$$\frac{}{\bullet \Vdash \bullet_{\mathsf{r}} : (\top^+,\mathsf{r})}$$

$$\frac{\Delta_1 \Vdash p_1 : (A_1,\mathsf{r}) \quad \Delta_2 \Vdash p_2 : (A_2,\mathsf{r})}{\Delta_1,\Delta_2 \Vdash (p_1,p_2) : (A_1 \wedge^+ A_2,\mathsf{r})} \qquad \frac{\Delta \Vdash p : (A_i,\mathsf{r})}{\Delta \Vdash \mathsf{inj}_i(p) : (A_1 \vee A_2,\mathsf{r})}$$

$$\frac{}{\sim(P,\mathsf{r}) \Vdash {}_{-\mathsf{l}}^{-} : (P,\mathsf{l})} \qquad \frac{}{(l^-,\mathsf{l}) \Vdash {}_{-\mathsf{l}}^{+} : (l^-,\mathsf{l})}$$

$$\frac{}{(\bot^-,\mathsf{l}) \Vdash \bullet_{\mathsf{l}} : (\bot^-,\mathsf{l})} \qquad \frac{\Delta \Vdash p : (A,\mathsf{r})}{\Delta,(\bot^-,\mathsf{l}) \Vdash \curvearrowright(p) : (\neg A,\mathsf{l})}$$

$$\frac{\Delta_1 \Vdash p_1 : (A_1,\mathsf{r}) \quad \Delta_2 \Vdash p_2 : (A_2,\mathsf{l})}{\Delta_1,\Delta_2 \Vdash p_1 :: p_2 : (A_1 \Rightarrow A_2,\mathsf{l})} \qquad \frac{\Delta \Vdash p : (A_i,\mathsf{l})}{\Delta \Vdash \pi_i(p) : (A_1 \wedge^- A_2,\mathsf{l})}$$

Figure 3: Decomposition relation for LAF_J

Again, we can already see in Fig. 3 the traditional right-introduction rules of positive connectives and left-introduction rules of negative connectives, and again, it is clear from these rules that LAF_J is well-founded.

We now interpret LAF_J sequents as intuitionistic sequents (from e.g. LJF [LM09]):

DEFINITION 10 (LAF_J sequents as two-sided LJF sequents)

First, when $\pm$ is either $+$ or $-$, we define

$$\begin{aligned}\mathsf{Im}^{\pm\mathsf{r}}(\Gamma) &:= \{A \mid (A,\mathsf{r}) \in \mathsf{Im}^{\pm}(\Gamma)\}\\ \mathsf{Im}^{+\mathsf{l}}(\Gamma) &:= \{l^- \mid (l^-,\mathsf{l}) \in \mathsf{Im}^{+}(\Gamma)\}\\ \mathsf{Im}^{-\mathsf{l}}(\Gamma) &:= \{N \mid (N,\mathsf{l}) \in \mathsf{Im}^{-}(\Gamma)\}\end{aligned}$$

Then we define the encoding:

$$\begin{aligned}\phi(\Gamma \vdash c) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] \longrightarrow [\mathsf{Im}^{+\mathsf{l}}(\Gamma),\mathsf{Im}^{-\mathsf{r}}(\Gamma)]\\ \phi(\Gamma \vdash x^+ : (l^-,\mathsf{l})) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] \xrightarrow{l^-} [\mathsf{Im}^{+\mathsf{l}}(\Gamma),\mathsf{Im}^{-\mathsf{r}}(\Gamma)]\\ \phi(\Gamma \vdash f : \sim(P,\mathsf{r})) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] \xrightarrow{P} [\mathsf{Im}^{+\mathsf{l}}(\Gamma),\mathsf{Im}^{-\mathsf{r}}(\Gamma)]\\ \phi(\Gamma \vdash [t^+ : (N,\mathsf{l})]) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] \xrightarrow{N} [\mathsf{Im}^{+\mathsf{l}}(\Gamma),\mathsf{Im}^{-\mathsf{r}}(\Gamma)]\\ \phi(\Gamma \vdash x^+ : (l^+,\mathsf{r})) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] -_{l^+}\!\!\rightarrow\\ \phi(\Gamma \vdash f : \sim(N,\mathsf{l})) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] -_{N}\!\!\rightarrow\\ \phi(\Gamma \vdash [t^+ : (P,\mathsf{r})]) &:= [\mathsf{Im}^{+\mathsf{r}}(\Gamma),\mathsf{Im}^{-\mathsf{l}}(\Gamma)] -_{P}\!\!\rightarrow\end{aligned}$$

In the first four cases, we require $\mathsf{Im}^{+\mathsf{l}}(\Gamma),\mathsf{Im}^{-\mathsf{r}}(\Gamma)$ to be a singleton (or be empty).

DEFINITION 11 (Typing contexts) We assume that we always have $(\bot^-,\mathsf{l}) \in \mathsf{Im}^+(\Gamma)$ and that

$$\begin{aligned}\mathsf{Im}^+(\Gamma;(l^+,\mathsf{r})) &= \mathsf{Im}^+(\Gamma) \cup \{(l^+,\mathsf{r})\} & \mathsf{Im}^-(\Gamma;a) &= \mathsf{Im}^-(\Gamma)\\ \mathsf{Im}^+(\Gamma;\sim M) &= \mathsf{Im}^+(\Gamma) & \mathsf{Im}^-(\Gamma;\sim(N,\mathsf{l})) &= \mathsf{Im}^-(\Gamma) \cup \{(N,\mathsf{l})\}\\ \mathsf{Im}^\pm(\Gamma;\bullet) &= \mathsf{Im}^\pm(\Gamma) & \mathsf{Im}^\pm(\Gamma;(\Delta_1,\Delta_2)) &= \mathsf{Im}^\pm(\Gamma;\Delta_1;\Delta_2)\\ \mathsf{Im}^+(\Gamma;(v,\mathsf{l})) &= \{(l^+,\mathsf{r}) \mid (l^+,\mathsf{r}) \in \mathsf{Im}^+(\Gamma)\} \cup \{(v,\mathsf{l}),(\bot^-,\mathsf{l})\}\\ \mathsf{Im}^-(\Gamma;\sim(P,\mathsf{r})) &= \{(N,\mathsf{l}) \mid (N,\mathsf{l}) \in \mathsf{Im}^-(\Gamma)\} \cup \{(P,\mathsf{r})\}\end{aligned}$$

where again $\pm$ stands for either $+$ or $-$ and v stands for either a negative literal l^- or $\bot^-$.

The first three lines are the same as those assumed for $K1$, except they are restricted to those cases where we do not try to add to Γ an atom or a molecule that is interpreted as going to the right-hand side of a sequent. When we want to do that, this atom or molecule should overwrite the previous atom(s) or molecule(s) that was (were) interpreted as being on the right-hand side; this is done in the last two lines, where $\mathsf{Im}^{+\mathsf{l}}(\Gamma),\mathsf{Im}^{-\mathsf{r}}(\Gamma)$ is completely erased.

THEOREM 2 *ϕ satisfies structural adequacy between $\mathsf{LAF}_J^{\mathsf{cf}}$ and LJF.*

The details are similar to those of Theorem 1, relying on the LJF properties expressed in [LM09].

4 Realisability semantics of LAF

We now look at the semantics of LAF systems, setting as an objective the definition of models and the proof of their correctness at the same generic level as that of LAF.

In this section, $\mathbb{P}(\mathscr{A})$ stands for the power set of a given set $\mathscr{A}$.

Given a LAF instance, we define the following notion of realisability algebra:

DEFINITION 12 (Realisability algebra) A *realisability algebra* is an algebra of the form

$$\left(\mathscr{L},\mathscr{P},\mathscr{N},\ \bot,\tilde{\mathsf{Co}},\begin{pmatrix}\mathsf{Pat}\to(\mathbb{D}_{\mathscr{L},\mathscr{N}}\to\mathscr{P})\\ p\ \mapsto\tilde{p}\end{pmatrix},\begin{pmatrix}(\mathsf{Pat}\rightharpoonup\mathsf{Terms})\times\tilde{\mathsf{Co}}\to\mathscr{N}\\ (f,\rho)\qquad\quad\mapsto[\![f]\!]_\rho\end{pmatrix},\begin{pmatrix}\mathbb{A}\to\mathbb{P}(\mathscr{L})\\ a\mapsto[\![a]\!]\end{pmatrix}\right)$$

where
- $\mathscr{L}, \mathscr{P}, \mathscr{N}$ are three arbitrary sets of elements called *label denotations*, *positive denotations*, *negative denotations*, respectively;
- $\perp$ is a relation between negative and positive denotations ($\perp \subseteq \mathscr{N} \times \mathscr{P}$), called the *orthogonality relation*;
- $\tilde{\mathsf{Co}}$ is a $(\mathscr{L}, \mathscr{N})$-context algebra, whose elements, denoted $\rho, \rho', \ldots$, are called *semantic contexts*.

The $(\mathscr{L}, \mathscr{N})$-decomposition algebra $\mathbb{D}_{\mathscr{L},\mathscr{N}}$ is abbreviated $\tilde{\mathbb{D}}$; its elements, denoted $\mathfrak{d}$, $\mathfrak{d}' \ldots$, are called *semantic decompositions*.

Given a model structure, we can define the interpretation of proof-terms. The model structure already gives an interpretation for the partial functions f from patterns to commands. We extend it to all proof-terms as follows:

DEFINITION 13 (Interpretation of proof-terms)

Positive terms (in Terms^+) are interpreted as positive denotations (in $\mathscr{P}$), decomposition terms (in $\mathsf{Terms}^{\mathsf{d}}$) are interpreted as semantic decompositions (in $\tilde{\mathbb{D}}$), and commands (in Terms) are interpreted as pairs in $\mathscr{N} \times \mathscr{P}$ (that may or may not be orthogonal), as follows:

$$[\![pd]\!]_\rho := \tilde{p}([\![d]\!]_\rho) \qquad \begin{array}{lll} [\![\bullet]\!]_\rho & := \bullet & [\![\langle x^- \mid t^+ \rangle]\!]_\rho := (\rho\,[x^-], [\![t^+]\!]_\rho) \\ [\![d_1, d_2]\!]_\rho & := [\![d_1]\!]_\rho, [\![d_2]\!]_\rho & [\![\langle f \mid t^+ \rangle]\!]_\rho := ([\![f]\!]_\rho, [\![t^+]\!]_\rho) \\ [\![x^+]\!]_\rho & := \rho\,[x^+] & \\ [\![f]\!]_\rho & := [\![f]\!]_\rho \text{ as given by the realisability algebra} & \end{array}$$

Our objective is now the Adequacy Lemma whereby, if t is of type A then the interpretation of t is in the interpretation of A. We have already defined the interpretation of proof-terms in a model structure. We now proceed to define the interpretation of types.

In system LAF, there are four concepts of "type inhabitation":
1. "proving" an atom by finding a suitable positive label in the typing context;
2. "proving" a molecule by choosing a way to decompose it into a typing decomposition;
3. "refuting" a molecule by case analysing all the possible ways of decomposing it into a typing decomposition;
4. "proving" a typing decomposition by inhabiting it with a decomposition term.

Correspondingly, we have the four following interpretations, with the interpretations of atoms (1.) in $\mathbb{P}(\mathscr{L})$ being arbitrary and provided as a parameter of a realisability algebra:

DEFINITION 14 (Interpretation of types and typing contexts) Assume the instance of LAF is well-founded. We define (by induction on the well-founded ordering between molecules):
2. the positive interpretation of a molecule in $\mathbb{P}(\mathscr{P})$;
3. the negative interpretation of a molecule in $\mathbb{P}(\mathscr{N})$;
4. the interpretation of a typing decomposition in $\mathbb{P}(\mathbb{D}_{\mathscr{L},\mathscr{N}})$:

$$
\begin{array}{lll}
[\![M]\!]^+ & := \{\tilde{p}(\mathfrak{d}) \in \mathscr{P} & \mid \mathfrak{d} \in [\![\Delta]\!], \text{ and } \Delta \Vdash p : M\} \\
[\![M]\!]^- & := \{\mathfrak{n} \in \mathscr{N} & \mid \forall \mathfrak{p} \in [\![M]\!]^+, \mathfrak{n} \perp \mathfrak{p}\} \\
[\![\bullet]\!] & := \{\bullet\} & \\
[\![\Delta_1, \Delta_2]\!] & := \{\mathfrak{d}_1, \mathfrak{d}_2 & \mid \mathfrak{d}_1 \in [\![\Delta_1]\!] \text{ and } \mathfrak{d}_2 \in [\![\Delta_2]\!]\} \\
[\![a]\!] & := [\![a]\!] & \text{as given by the realisability algebra} \\
[\![\sim M]\!] & := \{\sim \mathfrak{n} & \mid \mathfrak{n} \in [\![M]\!]^-\}
\end{array}
$$

We then define the interpretation of a typing context:

$$
\begin{array}{lll}
[\![\Gamma]\!] := \{\rho \in \tilde{\mathsf{Co}} \mid & \forall x^+ \in \mathsf{dom}^+(\rho),\ \rho\,[x^+] \in [\![\Gamma\,[x^+]]\!] \\
& \forall x^- \in \mathsf{dom}^-(\rho),\ \rho\,[x^-] \in [\![\Gamma\,[x^-]]\!]^- \quad \}
\end{array}
$$

Now that we have defined the interpretation of terms and the interpretation of types, we get to the Adequacy Lemma.

LEMMA 3 (Adequacy for LAF) We assume the following hypotheses:

Well-foundedness: The LAF instance is well-founded.

Typing correlation: If $\rho \in [\![\Gamma]\!]$ and $\mathfrak{d} \in [\![\Delta]\!]$ then $(\rho; \mathfrak{d}) \in [\![\Gamma; \Delta]\!]$.

Stability: If $\mathfrak{d} \in [\![\Delta]\!]$ for some Δ and $[\![f(p)]\!]_{\rho;\mathfrak{d}} \in \perp$, then $[\![f]\!]_\rho \perp \tilde{p}(\mathfrak{d})$.

We conclude that, for all $\rho \in [\![\Gamma]\!]$,

1. if $\Gamma \vdash [t^+ : M]$ then $[\![t^+]\!]_\rho \in [\![M]\!]^+$;
2. if $\Gamma \vdash d : \Delta$ then $[\![d]\!]_\rho \in [\![\Delta]\!]$;
3. if $\Gamma \vdash t$ then $[\![t]\!]_\rho \in \perp$.

Proof: See the proof in Coq [GL14b]. □

Looking at the Adequacy Lemma, the stability condition is traditional: it is the generalisation, to that level of abstraction, of the usual condition on the orthogonality relation in orthogonality models (those realisability models that are defined in terms of orthogonality, usually to model classical proofs [Gir87, DK00, Kri01, MM09]): orthogonality is "closed under anti-reduction". Here, we have not defined a notion of reduction on LAF proof-terms, but intuitively, we would expect to rewrite $\langle f \mid pd \rangle$ to $f(p)$ "substituted by d".

On the other hand, the typing correlation property is new, and is due to the level of abstraction we operate at: there is no reason why our data structure for typing contexts would relate to our data structure for semantic contexts, and the extension operation, in both of them, has so far been completely unspecified. Hence, we clearly need such an assumption to relate the two.

However, one may wonder when and why the typing correlation property should be satisfied. One may anticipate how typing correlation could hold for the instance LAF_{K1} of LAF. Definition 7 suggests that, in the definition of a typing context algebra, the extension operation does not depend on the nature of the atom a or molecule M that is being added to the context. So we could *parametrically* define $(\mathscr{A}, \mathscr{B})$-contexts for any sets $\mathscr{A}$ and $\mathscr{B}$ (in the sense of relational parametricity [Rey83]). The typing context algebra would be the instance where $\mathscr{A} = \mathbb{A}$ and $\mathscr{B} = \mathbb{M}$ and the semantic context algebra would be the instance where $\mathscr{A} = \mathscr{L}$ and $\mathscr{B} = \mathscr{N}$. Parametricity of context extension would then provide the typing correlation property.

5 Example: boolean models to prove Consistency

We now exhibit models to prove the consistency of LAF systems.

We call *boolean realisability algebra* a realisability algebra where $\bot = \emptyset$. The terminology comes from the fact that in such a realisability algebra, $[\![M]\!]^-$ can only take one of two values: $\emptyset$ or $\mathscr{N}$, depending on whether $[\![M]\!]^+$ is empty. A boolean realisability algebra satisfies Stability.

THEOREM 4 (Consistency of LAF instances) Assume we have a well-founded LAF instance, and a boolean realisability algebra for it, where typing correlation holds and there is an empty semantic context $\rho_\emptyset$. There is no empty typing context $\Gamma_\emptyset$ and command t such that $\Gamma_\emptyset \vdash t$.

Proof: The previous remark provides Stability. If there was such a $\Gamma_\emptyset$ and t, then we would have $\rho_\emptyset \in [\![\Gamma_\emptyset]\!]$, and the Adequacy Lemma (Lemma 3) would conclude $[\![t]\!]_{\rho_\emptyset} \in \emptyset$. □

We provide such a realisability model that works with all "parametric" LAF instances:

DEFINITION 15 (Trivial model for parametric LAF instances)

Assume we have a parametric LAF instance, i.e. an instance where the typing context algebra Co is the instance $\mathscr{G}_{\mathbb{A},\mathbb{M}}$ of a family of context algebras $(\mathscr{G}_{\mathscr{A},\mathscr{B}})_{\mathscr{A},\mathscr{B}}$ whose notion of extension is defined parametrically in $\mathscr{A},\mathscr{B}$. The *trivial boolean model* for it is:

$$\begin{array}{rrlll} \mathscr{L} := & \mathscr{P} := & \mathscr{N} := \mathtt{unit} & \tilde{\mathsf{Co}} := \mathscr{G}_{\mathtt{unit},\mathtt{unit}} & \\ & & \bot := \emptyset & \text{and therefore} & \\ & \forall \mathfrak{d} \in \tilde{\mathbb{D}}, & \tilde{p}(\mathfrak{d}) := () & \forall \rho \in \tilde{\mathsf{Co}}, \forall x^+ \in \mathsf{dom}^+(\rho), & \rho\,[x^+] := () \\ \forall f : \mathsf{Pat} \rightharpoonup \mathsf{Terms}, \forall \rho \in \tilde{\mathsf{Co}}, & & [\![f]\!]_\rho := () & \forall \rho \in \tilde{\mathsf{Co}}, \forall x^- \in \mathsf{dom}^-(\rho), & \rho\,[x^-] := () \\ & \forall a \in \mathbb{A}, & [\![a]\!] := \mathtt{unit} & & \end{array}$$

Note that, not only can $[\![M]\!]^-$ only take one of the two values $\emptyset$ or unit, but $[\![M]\!]^+$ can also only take one of the two values $\emptyset$ or unit.

We can now use such a structure to derive consistency of parametric LAF instances:

COROLLARY 5 (Consistency for parametric LAF instances) Assume we have a parametric LAF instance that is well-founded and assume there is an empty $(\mathtt{unit},\mathtt{unit})$-context in $\mathscr{G}_{\mathtt{unit},\mathtt{unit}}$. Then there is no empty typing context $\Gamma_\emptyset$ and command t such that $\Gamma_\emptyset \vdash t$.

In particular, this is the case for LAF_{K1}.

The system LAF_J does not fall in the above category since the operation of context extension is not parametric enough: when computing $\Gamma;a$ (resp. $\Gamma;{\sim}M$), we have to make a case analysis on whether a is of the form (l^+,r) or (v,l) (resp. whether M is of the form (N,l) or (P,r)).

But we can easily adapt the above trivial model into a not-as-trivial-but-almost model for LAF_J, as is shown in [GL14a], Ch. 6.

6 Conclusion and Further Work

6.1 Contributions

In this paper we have used, and slightly tweaked, a system with proof-terms proposed by Zeilberger for *big-step focussing* [Zei08a], which abstracts away from the idiosyncrasies of logical connectives and

(to some extent) logical systems: In particular we have shown how two focussed sequent calculi of the literature, namely LKF and LJF [LM09], are captured as instances of this abstract focussing system LAF.

Building on Munch-Maccagnoni's description [MM09] of classical realisability in the context of polarised classical logic and focussing, we have then presented the realisability models for LAF, thus providing a generic approach to the denotational semantics of focussed proofs. Central to this is the Adequacy Lemma 3, which connects typing and realisability, and our approach is generic in that the Adequacy Lemma is proved once and for all, holding for all focussed systems that can be captured as LAF instances.

Incidently, a by-product of this paper is that we provided proof-term assigments for LKF and LJF, and provided realisability semantics for their proofs. We believe this to be new.

But showing the Adequacy Lemma at this level of abstraction was also motivated by the will to exhibit how close typing and realisability are while differing in an essential way:

6.2 Typing vs. realisability

Concerning the *positives*, typing and realisability simply mimic each other: Ignoring contexts,

- in typing, $\vdash [t^+ : M]$ means t^+ is of the form pd with $\Delta \Vdash p : M$ and $\vdash d : \Delta$ for some Δ;
- in realisability, $[\![t^+]\!] \in [\![M]\!]^+$ means $[\![t^+]\!]$ is of the form $\tilde{p}(\mathfrak{d})$ with $\Delta \Vdash p : M$ and $\mathfrak{d} \in \Delta$ for some Δ.

Concerning the *negatives*, it is appealing to relate the quantification in rule async with the quantification in the orthogonality construction:

- in typing, $\vdash f : {\sim}M$ means that for all p and Δ such that $\Delta \Vdash p : M$, we have $; \Delta \vdash f(p)$ (Δ extending the empty typing context);
- in realisability, $[\![f]\!] \in [\![M]\!]^-$ means that for all p and Δ such that $\Delta \Vdash p : M$, for all $\mathfrak{d} \in [\![\Delta]\!]$ we have $[\![f]\!] \perp \tilde{p}(\mathfrak{d})$, usually obtained from $[\![f(p)]\!]_{;\mathfrak{d}} \in \perp$ ($\mathfrak{d}$ extending the empty semantic context).

In both cases, a "contradiction" needs to be reached for all p and Δ decomposing M. But in typing, the proof-term $f(p)$ witnessing the contradiction can only depend on the pattern p and must treat its holes (whose types are agregated in Δ) parametrically, while in realisability, the reason why $[\![f]\!] \perp \tilde{p}(\mathfrak{d})$ holds, though it may rely on the computation of $f(p)$, can differ for every $\mathfrak{d} \in [\![\Delta]\!]$. It is the same difference that lies between the usual rule for $\forall$ in a Natural Deduction system for arithmetic, and the ω-rule [Hil31, Sch50]:

$$\frac{A}{\forall nA}\ \forall\text{-intro} \qquad \frac{\left\{{}^{0}\!/_{n}\right\}A \quad \left\{{}^{1}\!/_{n}\right\}A \quad \left\{{}^{2}\!/_{n}\right\}A \quad \dots}{\forall nA}\ \omega$$

The difference explains why typing is (usually) decidable, while realisability is not, and contributes to the idea that "typing is but a decidable approximation of realisability".

We believe that we have brought typing and realisability as close as they could get, emphasising where they coincide and where they differ, in a framework stripped from the idiosyncrasies of logics, of their connectives, and of the implementation of those functions we use for witnessing refutations, i.e. inhabiting negatives (e.g. the λ of λ-calculus).

6.3 Coq formalisation and additional results

In this paper we have also exhibited simple realisability models to prove the consistency of LAF instances.

The parameterised LAF system has been formalised in Coq [GL14b], together with realisability algebras. The Adequacy Lemma 3 and the Consistency result (Corollary 5) are proved there as well. Because of the abstract level of the formalism, very few concrete structures are used, only one for $(\mathscr{A},\mathscr{B})$-decompositions and one for proof-terms; rather, Coq's *records* are used to formalise the algebras used throughout the paper, declaring type fields for the algebras' support sets, term fields for operations, and proof fields for specifications. Coercions between records (and a few structures declared to be canonical) are used to lighten the proof scripts.

Besides this, the Coq formalisation presents no major surprises. It contributes to the corpus and promotion of machine-checked theories and proofs. However, formalising this in Coq was a particularly enlightening process, directly driving the design and definitions of the concepts. In getting to the essence of focussing and stripping the systems from the idiosyncrasies of logics and of their connectives, Coq was particularly useful: Developing the proofs led to identifying the concepts (e.g. what a typing context is), with their basic operations and their minimal specifications. Definitions and hypotheses (e.g. the three hypotheses of the Adequacy Lemma) were systematically weakened to the minimal form that would let proofs go through. Lemma statements were identified so as to exactly fill-in the gaps of inductive proofs, and no more.

The formalisation was actually done for a *first-order* version of LAF, that is fully described in [GL14a]. That in itself forms a proper extension of Zeilberger's systems [Zei08a]. In this paper though we chose to stick to the propositional fragment to simplify the presentation.

Regarding realisability models, more interesting examples than those given here to prove consistency can obviously be built to illustrate this kind of semantics. In particular, realisability models built from the term syntax can be used to prove normalisation properties of LAF, as shown in [GL14a]. Indeed, one of the appeals of big-step focussing systems is an elegant notion of cut-reduction, based on rewrite rules on proof-terms and with a functional programming interpretation in terms of *pattern-matching*. A cut-reduction system at the abstraction level of LAF is given in [GL14a], in terms of an abstract machine (performing head-reduction). A command t is evaluated in an evaluation context ρ; denoting such a pair as $\langle\!\langle\, t \mid \rho \,\rangle\!\rangle$, we have the main reduction rule (where d' stands for the evaluation of d in the evaluation context ρ):

$$\langle\!\langle\, \langle f \mid pd \rangle \mid \rho \,\rangle\!\rangle \longrightarrow \langle\!\langle\, f(p) \mid \rho; d' \,\rangle\!\rangle$$

Normalisation of this system (for a well-founded LAF instance), is proved by building a syntactic realisability model, in which orthogonality holds when the interaction between a negative denotation and a positive one is normalising. This model, together with the head normalisation result, are also formalised in Coq [GL14b]. It forms a formal connection, via the orthogonality techniques, between proofs of normalisation *à la* Tait-Girard and realisability. From this termination result, an informal argument is proposed [GL14a] to infer cut-elimination, but the argument still needs to be formalised in Coq. This is tricky since, cut-elimination needing to be performed arbitrarily deeply in a proof-tree ("under lambdas"), we need to formalise a notion of reduction on those functions we use for witnessing refutations, for which we have no syntax.

Finally, more work needs to be done on formalising the connections between LAF and other related systems: firstly, LAF is very strongly related to *ludics* [Gir01], a system for big-step focussing for linear logic, and which is also related to game semantics. LAF can be seen as a non-linear variant of ludics, our proof-term-syntax more-or-less corresponding to ludics' *designs*. But in order to get linearity, LAF would need to force it in the typing rules for the decomposition terms x^+, $\bullet$, and d_1, d_2. It would also be interesting to investigate whether or how LAF could be adapted to modal logics.

References

[And92] J. M. Andreoli. Logic programming with focusing proofs in linear logic. *J. Logic Comput.*, 2(3):297–347, 1992. DOI:10.1093/logcom/2.3.297

[DJS95] V. Danos, J.-B. Joinet, and H. Schellinx. LKQ and LKT: sequent calculi for second order logic based upon dual linear decompositions of classical implication. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Proc. of the Work. on Advances in Linear Logic*, volume 222 of *London Math. Soc. Lecture Note Ser.*, pages 211–224. Cambridge University Press, 1995.

[DK00] V. Danos and J.-L. Krivine. Disjunctive tautologies as synchronisation schemes. In P. Clote and H. Schwichtenberg, editors, *Proc. of the 9th Annual Conf. of the European Association for Computer Science Logic (CSL'00)*, volume 1862 of *LNCS*, pages 292–301. Springer-Verlag, 2000. DOI:10.1007/3-540-44622-2_19

[DL07] R. Dyckhoff and S. Lengrand. Call-by-value λ-calculus and LJQ. *J. Logic Comput.*, 17:1109–1134, 2007. DOI:10.1093/logcom/exm037

[Gir87] J.-Y. Girard. Linear logic. *Theoret. Comput. Sci.*, 50(1):1–101, 1987. DOI:10.1016/0304-3975(87)90045-4

[Gir91] J.-Y. Girard. A new constructive logic: Classical logic. *Math. Structures in Comput. Sci.*, 1(3):255–296, 1991. DOI:10.1017/S0960129500001328

[Gir01] J. Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001. DOI:10.1017/S096012950100336X

[GL14a] S. Graham-Lengrand. *Polarities & Focussing: a journey from Realisability to Automated Reasoning*. Habilitation thesis, Université Paris-Sud, 2014. Available at http://hal.archives-ouvertes.fr/tel-01094980

[GL14b] S. Graham-Lengrand. Polarities & focussing: a journey from realisability to automated reasoning – Coq proofs of Part II, 2014. http://www.lix.polytechnique.fr/~lengrand/Work/HDR/

[Hil31] D. Hilbert. Die Grundlegung der elementaren Zahlenlehre. *Mathematische Annalen*, 104:485–494, 1931.

[Kle45] S. Kleene. On the interpretation of intuitionistic number theory. *J. of Symbolic Logic*, 10:109–124, 1945. DOI:10.2307/2269016

[Kri01] J.-L. Krivine. Typed lambda-calculus in classical Zermelo-Frænkel set theory. *Arch. Math. Log.*, 40(3):189–205, 2001. DOI:10.1007/s001530000057

[LM09] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoret. Comput. Sci.*, 410(46):4747–4768, 2009. DOI:10.1016/j.tcs.2009.07.041

[MM09] G. Munch-Maccagnoni. Focalisation and classical realisability. In E. Grädel and R. Kahle, editors, *Proc. of the 18th Annual Conf. of the European Association for Computer Science Logic (CSL'09)*, volume 5771 of *LNCS*, pages 409–423. Springer-Verlag, 2009. DOI:10.1007/978-3-642-04027-6_30

[Rey83] J. C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Proc. of the IFIP 9th World Computer Congress - Information Processing*, pages 513–523. North-Holland, 1983.

[Sch50] K. Schütte. Beweistheoretische Erfassung der unendlichen Induktion in der Zahlentheorie. *Mathematische Annalen*, 122:369–389, 1950.

[Zei08a] N. Zeilberger. Focusing and higher-order abstract syntax. In G. C. Necula and P. Wadler, editors, *Proc. of the 35th Annual ACM Symp. on Principles of Programming Languages (POPL'08)*, pages 359–369. ACM Press, 2008. DOI:10.1145/1328438.1328482

[Zei08b] N. Zeilberger. On the unity of duality. *Ann. Pure Appl. Logic*, 153(1-3):66–96, 2008. DOI:10.1016/j.apal.2008.01.001

Multiplicative-Additive Focusing for Parsing as Deduction

Glyn Morrill Oriol Valentín
Department of Computer Science
Universitat Politècnica de Catalunya
Barcelona
morrill@cs.upc.edu
oriol.valentin@gmail.com

Spurious ambiguity is the phenomenon whereby distinct derivations in grammar may assign the same structural reading, resulting in redundancy in the parse search space and inefficiency in parsing. Understanding the problem depends on identifying the essential mathematical structure of derivations. This is trivial in the case of context free grammar, where the parse structures are ordered trees; in the case of type logical categorial grammar, the parse structures are proof nets. However, with respect to multiplicatives intrinsic proof nets have not yet been given for displacement calculus, and proof nets for additives, which have applications to polymorphism, are not easy to characterise. Here we approach multiplicative-additive spurious ambiguity by means of the proof-theoretic technique of focalisation.

1 Introduction

In context free grammar (CFG) sequential rewriting derivations exhibit spurious ambiguity: distinct rewriting derivations may correspond to the same parse structure (tree) and the same structural reading.[1] In this case it is transparent to develop parsing algorithms avoiding spurious ambiguity by reference to parse trees. In categorial grammar (CG) the problem is more subtle. The Cut-free Lambek sequent proof search space is finite, but involves a combinatorial explosion of spuriously ambiguous sequential proofs. This can be understood, analogously to CFG, as inessential rule reorderings, which we parallelise in underlying geometric parse structures which are (planar) proof nets.

The planarity of Lambek proof nets reflects that the formalism is continuous or concatenative. But the challenge of natural grammar is discontinuity or apparent displacement, whereby there is syntactic/semantic mismatch, or elements appearing out of place. Hence the subsumption of Lambek calculus by displacement calculus **D** including intercalation as well as concatenation [17].

Proof nets for **D** must be partially non-planar; steps towards intrinsic correctness criteria for displacement proof nets are made in [5] and [13]. Additive proof nets are considered in [7] and [1]. However, even in the case of Lambek calculus, parsing by reference to intrinsic criteria [14], [18], appendix B, is not more efficient than parsing by reference to extrinsic criteria of normalised sequent calculus [6]. In its turn, on the other hand, normalisation does not extend to product left rules and product unit left rules nor to additives. The focalisation of [2] is a methodology midway between proof nets and normalisation. Here we apply the focusing discipline to the parsing as deduction of **D** with additives.

In [4] multifocusing is defined for unit-free MALL,[2] providing canonical sequent proofs; an eventual goal would be to formulate multifocusing for multiplicative-additive categorial logic and for categorial

[1]Research partially supported by SGR2014-890 (MACDA) of the Generalitat de Catalunya and MINECO project APCOM (TIN2014-57226-P), and by an ICREA Acadèmia 2012 to GM. Thanks to three anonymous WoF reviewers for comments and suggestions, and to Iliano Cervesato for editorial attention. All errors are our own.

[2]Here we include units, which are linguistically relevant.

I. Cervesato and C. Schürmann (Eds.)
First Workshop on Focusing (WoF'15)
EPTCS 197, 2015, pp. 29–54, doi:10.4204/EPTCS.197.4

logic generally. In this respect the present paper represents an intermediate step. Note that [19] develops focusing for Lambek calculus with additives, but not for displacement logic, for which we show completeness of focusing here.

1.1 Spurious ambiguity in CFG

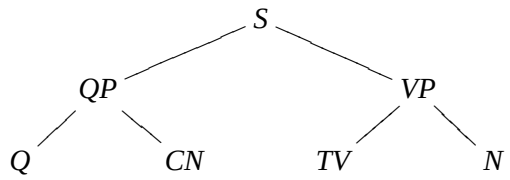
Consider the following production rules:

$$S \to QP\ VP$$
$$QP \to Q\ CN$$
$$VP \to TV\ N$$

These generate the following sequential rewriting derivations:

$$S \to QP\ VP \to Q\ CN\ VP \to Q\ CN\ TV\ N$$
$$S \to QP\ VP \to QP\ TV\ N \to Q\ CN\ TV\ N$$

These sequential rewriting derivations correspond to the same parellelised parse structure:



And they correspond to the same structural reading; sequential rewriting has *spurious ambiguity*.

1.2 Spurious ambiguity in CG

Lambek calculus is a logic of strings with the operation $+$ of concatenation. Recall the definitions of types, configurations and sequents in the Lambek calculus **L** [11], in terms of a set $\mathscr{P}$ of primitive types (the original Lambek calculus did not include the product unit):

(1)

Types	$\mathscr{F}$	::=	$\mathscr{P} \mid \mathscr{F}/\mathscr{F} \mid \mathscr{F}\backslash\mathscr{F} \mid \mathscr{F}\bullet\mathscr{F}$
Configurations	$\mathscr{O}$	::=	$\Lambda \mid \mathscr{F},\mathscr{O}$
Sequents	Σ	::=	$\mathscr{O} \Rightarrow \mathscr{F}$

Lambek calculus types have the following interpretation:

$$\begin{array}{lcl} [[C/B]] & = & \{s_1 \mid \forall s_2 \in [[B]], s_1{+}s_2 \in [[C]]\} \\ [[A\backslash C]] & = & \{s_2 \mid \forall s_1 \in [[A]], s_1{+}s_2 \in [[C]]\} \\ [[A\bullet B]] & = & \{s_1{+}s_2 \mid s_1 \in [[A]] \ \&\ s_2 \in [[B]]\} \end{array}$$

The logical rules of **L** are as follows:

$$\frac{\Gamma \Rightarrow A \qquad \Delta(C) \Rightarrow D}{\Delta(\Gamma, A\backslash C) \Rightarrow D}\backslash L \qquad \frac{A,\Gamma \Rightarrow C}{\Gamma \Rightarrow A\backslash C}\backslash R$$

$$\frac{\Gamma \Rightarrow B \qquad \Delta(C) \Rightarrow D}{\Delta(C/B, \Gamma) \Rightarrow D}/L \qquad \frac{\Gamma, B \Rightarrow C}{\Gamma \Rightarrow C/B}/R$$

$$\frac{\Delta(A,B) \Rightarrow D}{\Delta(A\bullet B) \Rightarrow D}\bullet L \qquad \frac{\Gamma_1 \Rightarrow A \qquad \Gamma_2 \Rightarrow B}{\Gamma_1, \Gamma_2 \Rightarrow A\bullet B}\bullet R$$

$$\dfrac{CN \Rightarrow CN \qquad \dfrac{N \Rightarrow N \qquad \dfrac{S \Rightarrow S \qquad N, N\backslash S \Rightarrow S}{N \Rightarrow N \qquad N, N\backslash S \Rightarrow S}\backslash L \qquad \dfrac{N, (N\backslash S)/N, N \Rightarrow S}{(N\backslash S)/N, N \Rightarrow N\backslash S}/L \qquad S \Rightarrow S}{S/(N\backslash S), (N\backslash S)/N, N \Rightarrow S}\backslash R}{(S/(N\backslash S))/CN, CN, (N\backslash S)/N, N \Rightarrow S}/L$$

$$\dfrac{N \Rightarrow N \qquad \dfrac{CN \Rightarrow CN \qquad \dfrac{\dfrac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\backslash L \qquad \dfrac{N\backslash S \Rightarrow N\backslash S \qquad S \Rightarrow S}{S/(N\backslash S), N\backslash S \Rightarrow S}/L}{N, N\backslash S \Rightarrow S}\backslash R}{(S/(N\backslash S))/CN, CN, N\backslash S \Rightarrow S}/L}{(S/(N\backslash S))/CN, CN, (N\backslash S)/N, N \Rightarrow S}/L$$

Figure 1: Spurious ambiguity

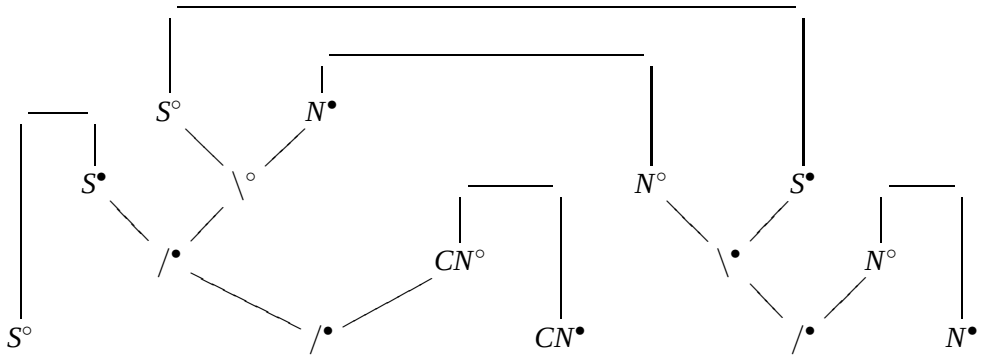


Figure 2: Proof net

Even amongst Cut-free proofs there is spurious ambiguity; consider for example the sequential derivations of Figure 1. These have the same parallelised parse structure (proof net) of Figure 2.

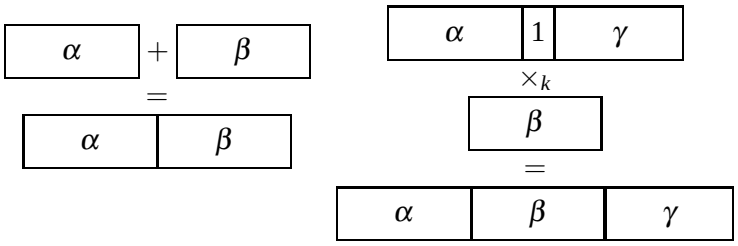
Lambek proof structures are planar graphs which must satisfy certain global and local properties to be correct as proofs (proof nets). Proof nets provide a geometric perspective on derivational equivalence. Alternatively we may identify the same algebraic parse structure (Curry-Howard term):

$$((x_Q\, x_{CN})\, \lambda x((x_{TV}\, x_N)\, x))$$

But Lambek calculus is continuous (planarity). A major issue issue in grammar is discontinuity, hence the displacement calculus.

2 D with additives, DA

In this section we present displacement calculus **D**, and a displacement logic **DA** comprising **D** with additives. Although **D** is indeed a conservative extension of **L**, we think of it not just as an *extension* of Lambek calculus but as a *generalisation*, because it involves a whole new machinery of sequent calculus to deal with discontinuity. Displacement calculus is a logic of discontinuous strings — strings punctuated by a *separator* 1 and subject to operations of append and plug; see Figure 3. Recall the definition of types and their sorts, configurations and their sorts, and sequents, for the displacement calculus with additives:



append $+ : L_i, L_j \to L_{i+j}$ plug $\times_k : L_{i+1}, L_j \to L_{i+j}$

Figure 3: Append and plug

(2) Types
$$\begin{array}{rcll}
\mathscr{F}_i & ::= & \mathscr{F}_{i+j}/\mathscr{F}_j & \\
\mathscr{F}_j & ::= & \mathscr{F}_i\backslash\mathscr{F}_{i+j} & \\
\mathscr{F}_{i+j} & ::= & \mathscr{F}_i \bullet \mathscr{F}_j & \\
\mathscr{F}_0 & ::= & I & \\
\mathscr{F}_{i+1} & ::= & \mathscr{F}_{i+j}\uparrow_k \mathscr{F}_j & 1 \le k \le i+1 \\
\mathscr{F}_j & ::= & \mathscr{F}_{i+1}\downarrow_k \mathscr{F}_{i+j} & 1 \le k \le i+1 \\
\mathscr{F}_{i+j} & ::= & \mathscr{F}_{i+1} \odot_k \mathscr{F}_j & 1 \le k \le i+1 \\
\mathscr{F}_1 & ::= & J & \\
\mathscr{F}_i & ::= & \mathscr{F}_i \& \mathscr{F}_i & \\
\mathscr{F}_i & ::= & \mathscr{F}_i \oplus \mathscr{F}_i &
\end{array}$$

Sort $sA =$ the i s.t. $A \in \mathscr{F}_i$
For example, $s(S\uparrow_1 N)\uparrow_2 N = s(S\uparrow_1 N)\uparrow_1 N = 2$ where $sN = sS = 0$

Configurations
$$\begin{array}{rcl}
\mathscr{O} & ::= & \Lambda \mid \mathscr{T}, \mathscr{O} \\
\mathscr{T} & ::= & 1 \mid \mathscr{F}_0 \mid \mathscr{F}_{i>0}\{\underbrace{\mathscr{O} : \ldots : \mathscr{O}}_{i\,\mathscr{O}'\mathrm{s}}\}
\end{array}$$
For example, there is the configuration $(S\uparrow_1 N)\uparrow_2 N\{N, 1 : S\uparrow_1 N, S\}, 1, N, 1$

Sort $s\mathscr{O} = |\mathscr{O}|_1$
For example $s(S\uparrow_1 N)\uparrow_2 N\{N, 1 : S\uparrow_1 N, S\}, 1, N, 1 = 3$

Sequents $\Sigma ::= \mathscr{O} \Rightarrow A$ s.t. $s\mathscr{O} = sA$

The figure $\overrightarrow{A}$ of a type A is defined by:

$$\overrightarrow{A} = \begin{cases} A & \text{if } sA = 0 \\ A\{\underbrace{1 : \ldots : 1}_{sA\ 1's}\} & \text{if } sA > 0 \end{cases}$$

Where Γ is a configuration of sort i and $\Delta_1, \ldots, \Delta_i$ are configurations, the *fold* $\Gamma \otimes \langle \Delta_1 : \ldots : \Delta_i \rangle$ is the result of replacing the successive 1's in Γ by $\Delta_1, \ldots, \Delta_i$ respectively.

Where Δ is a configuration of sort $i > 0$ and Γ is a configuration, the kth metalinguistic wrap $\Delta|_k \Gamma$, $1 \le k \le i$, is given by

(3) $\Delta|_k\Gamma =_{df} \Delta \otimes \langle \underbrace{1:\ldots:1}_{k-1\ 1\text{'s}} : \Gamma : \underbrace{1:\ldots:1}_{i-k\ 1\text{'s}} \rangle$

i.e. $\Delta|_k\Gamma$ is the configuration resulting from replacing by Γ the kth separator in Δ.

In broad terms, syntactical interpretation of displacement calculus is as follows:

$$\begin{array}{rcl}
[[C/B]] & = & \{s_1 |\ \forall s_2 \in [[B]], s_1{+}s_2 \in [[C]]\} \\
[[A\backslash C]] & = & \{s_2 |\ \forall s_1 \in [[A]], s_1{+}s_2 \in [[C]]\} \\
[[A{\bullet}B]] & = & \{s_1{+}s_2 |\ s_1 \in [[A]]\ \&\ s_2 \in [[B]]\} \\
[[I]] & = & \{0\}
\end{array}$$

$$\begin{array}{rcl}
[[C{\uparrow_k}B]] & = & \{s_1 |\ \forall s_2 \in [[B]], s_1 \times_k s_2 \in [[C]]\} \\
[[A{\downarrow_k}C]] & = & \{s_2 |\ \forall s_1 \in [[A]], s_1 \times_k s_2 \in [[C]]\} \\
[[A{\odot_k}B]] & = & \{s_1 \times_k s_2 |\ s_1 \in [[A]]\ \&\ s_2 \in [[B]]\} \\
[[J]] & = & \{1\}
\end{array}$$

The logical rules of the displacement calculus with additives are as follows, where $\Delta\langle\Gamma\rangle$ abbreviates $\Delta_0(\Gamma \otimes \langle\Delta_1 : \ldots : \Delta_i\rangle)$:

$$\frac{\Gamma \Rightarrow B \qquad \Delta\langle \overrightarrow{C} \rangle \Rightarrow D}{\Delta\langle \overrightarrow{C/B}, \Gamma \rangle \Rightarrow D}\, /L \qquad \frac{\Gamma, \overrightarrow{B} \Rightarrow C}{\Gamma \Rightarrow C/B}\, /R$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta\langle \overrightarrow{C} \rangle \Rightarrow D}{\Delta\langle \Gamma, \overrightarrow{A\backslash C} \rangle \Rightarrow D}\, \backslash L \qquad \frac{\overrightarrow{A}, \Gamma \Rightarrow C}{\Gamma \Rightarrow A\backslash C}\, \backslash R$$

$$\frac{\Delta\langle \overrightarrow{A}, \overrightarrow{B} \rangle \Rightarrow D}{\Delta\langle \overrightarrow{A{\bullet}B} \rangle \Rightarrow D}\, {\bullet}L \qquad \frac{\Gamma_1 \Rightarrow A \qquad \Gamma_2 \Rightarrow B}{\Gamma_1, \Gamma_2 \Rightarrow A{\bullet}B}\, {\bullet}R$$

$$\frac{\Delta\langle \Lambda \rangle \Rightarrow A}{\Delta\langle \overrightarrow{I} \rangle \Rightarrow A}\, IL \qquad \frac{}{\Lambda \Rightarrow I}\, IR$$

$$\frac{\Gamma \Rightarrow B \qquad \Delta\langle \overrightarrow{C} \rangle \Rightarrow D}{\Delta\langle \overrightarrow{C{\uparrow_k}B}|_k \Gamma \rangle \Rightarrow D}\, {\uparrow_k}L \qquad \frac{\Gamma|_k \overrightarrow{B} \Rightarrow C}{\Gamma \Rightarrow C{\uparrow_k}B}\, {\uparrow_k}R$$

$$\frac{\Delta\langle \overrightarrow{A}|_k \overrightarrow{B} \rangle \Rightarrow D}{\Delta\langle \overrightarrow{A{\odot_k}B} \rangle \Rightarrow D}\, {\odot_k}L \qquad \frac{\Gamma_1 \Rightarrow A \qquad \Gamma_2 \Rightarrow B}{\Gamma_1|_k\Gamma_2 \Rightarrow A{\odot_k}B}\, {\odot_k}R$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta\langle \overrightarrow{C} \rangle \Rightarrow D}{\Delta\langle \Gamma|_k \overrightarrow{A{\downarrow_k}C} \rangle \Rightarrow D}\, {\downarrow_k}L \qquad \frac{\overrightarrow{A}|_k\Gamma \Rightarrow C}{\Gamma \Rightarrow A{\downarrow_k}C}\, {\downarrow_k}R$$

$$\frac{\Delta\langle 1 \rangle \Rightarrow A}{\Delta\langle \overrightarrow{J} \rangle \Rightarrow A}\, JL \qquad \frac{}{1 \Rightarrow J}\, JR$$

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\&B}\rangle \Rightarrow C}\,\&L_1 \qquad \frac{\Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\&B}\rangle \Rightarrow C}\,\&L_2$$

$$\frac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A\&B}\,\&R$$

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C \qquad \Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\oplus B}\rangle \Rightarrow C}\,\oplus L$$

$$\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A\oplus B}\,\oplus R_1 \qquad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A\oplus B}\,\oplus R_2$$

The continuous multiplicatives $\{/,\ \backslash,\ \bullet,\ I\}$ of Lambek (1958[11]; 1988[10]), are the basic means of categorial (sub)categorization. The directional divisions over, $/$, and under, $\backslash$, are exemplified by assignments such as *the*$:N/CN$ for *the man*$:N$ and *sings*$:N\backslash S$ for *John sings*$:S$, and *loves*$:(N\backslash S)/N$ for *John loves Mary*$:S$. Hence, for *the man*:

$$\frac{CN \Rightarrow CN \qquad N \Rightarrow N}{N/CN, N \Rightarrow N}\,/L$$

And for *John sings* and *John loves Mary*:

$$\frac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\,\backslash L \qquad \frac{N \Rightarrow N \qquad \dfrac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\,\backslash L}{N, (N\backslash S)/N, N \Rightarrow S}\,/L$$

The continuous product $\bullet$ is exemplified by a 'small clause' assignment such as *considers*$:(N\backslash S)/(N\bullet(CN/CN))$ for *John considers Mary socialist*$:S$.

$$\frac{N \Rightarrow N \qquad \dfrac{\dfrac{CN \Rightarrow CN \qquad CN \Rightarrow CN}{CN/CN, CN \Rightarrow CN}\,/L}{CN/CN \Rightarrow CN/CN}\,/R}{N, CN/CN \Rightarrow N\bullet(CN/CN)}\,\bullet R \qquad \frac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\,\backslash L}{N, (N\backslash S)/(N\bullet(CN/CN)), N, CN/CN \Rightarrow S}\,/L$$

Of course this use of product is not essential: we could just as well have used $((N\backslash S)/(CN/CN))/N$ since in general we have both $A/(C\bullet B) \Rightarrow (A/B)/C$ (currying) and $(A/B)/C \Rightarrow A/(C\bullet B)$ (uncurrying).

The discontinuous multiplicatives $\{\uparrow, \downarrow, \odot, J\}$, the displacement connectives, of Morrill and Valentín (2010[16]), Morrill et al. (2011[17]), are defined in relation to intercalation. When the value of the k subscript is one it may be omitted, i.e. it defaults to one. Circumfixation, or extraction, $\uparrow$, is exemplified by a discontinuous idiom assignment *gives*$+1+$*the*$+$*cold*$+$*shoulder*: $(N\backslash S)\uparrow N$ for *Mary gives the man the cold shoulder*$:S$:

$$\frac{\dfrac{CN \Rightarrow CN \qquad N \Rightarrow N}{N/CN, CN \Rightarrow N}\,/L \qquad \dfrac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\,\backslash L}{N, (N\backslash S)\uparrow N\{N/CN, CN\} \Rightarrow S}\,\uparrow L$$

Infixation, $\downarrow$, and extraction together are exemplified by a quantifier assignment *everyone*: $(S{\uparrow}N){\downarrow}S$ simulating Montague's S14 quantifying in:

$$\dfrac{\dfrac{\ldots,N,\ldots \Rightarrow S}{\ldots,1,\ldots \Rightarrow S{\uparrow}N}\uparrow R \quad \dfrac{}{S \Rightarrow S}\,id}{\ldots,(S{\uparrow}N){\downarrow}S,\ldots \Rightarrow S}\downarrow L$$

Circumfixation and discontinuous product, $\odot$, are illustrated in an assignment to a relative pronoun *that*: $(CN\backslash CN)/((S{\uparrow}N){\odot}I)$ allowing both peripheral and medial extraction, *that John likes*: $CN\backslash CN$ and *that John saw today*: $CN\backslash CN$:

$$\dfrac{\dfrac{\dfrac{N,(N\backslash S)/N,N \Rightarrow S}{N,(N\backslash S)/N,1 \Rightarrow S{\uparrow}N}\uparrow R \quad \dfrac{}{\Rightarrow I}\,IL}{N,(N\backslash S)/N \Rightarrow (S{\uparrow}N){\odot}I}\odot R \qquad CN\backslash CN \Rightarrow CN\backslash CN}{(CN\backslash CN)/((S{\uparrow}N){\odot}I),N,(N\backslash S)/N \Rightarrow CN\backslash CN}/L$$

$$\dfrac{\dfrac{\dfrac{N,(N\backslash S)/N,N,S\backslash S \Rightarrow S}{N,(N\backslash S)/N,1,S\backslash S \Rightarrow S{\uparrow}N}\uparrow R \quad \dfrac{}{\Rightarrow I}\,IL}{N,(N\backslash S)/N,S\backslash S \Rightarrow (S{\uparrow}N){\odot}I}\odot R \qquad CN\backslash CN \Rightarrow CN\backslash CN}{(CN\backslash CN)/((S{\uparrow}N){\odot}I),N,(N\backslash S)/N,S\backslash S \Rightarrow CN\backslash CN}/L$$

The additive conjunction and disjunction $\{\&, \oplus\}$ of Lambek (1961[9]), Morrill (1990[15]), and Kanazawa (1992[8]), capture polymorphism. For example the additive conjunction & can be used for *rice*: $N\&CN$ as in *rice grows*: S and *the rice grows*: S:

$$\dfrac{\dfrac{\dfrac{N \Rightarrow N}{N\&CN \Rightarrow N}\&L_1 \qquad S \Rightarrow S}{N\&CN,N\backslash S \Rightarrow S}\backslash L \qquad \dfrac{N/CN,CN,N\backslash S \Rightarrow S}{N/CN,N\&CN,N\backslash S \Rightarrow S}\&L_2$$

The additive disjunction $\oplus$ can be used for *is*: $(N\backslash S)/(N{\oplus}(CN/CN))$ as in *Tully is Cicero*: S and *Tully is humanist*: S:

$$\dfrac{\dfrac{\dfrac{N \Rightarrow N}{N \Rightarrow N{\oplus}(CN/CN)}\oplus R_1 \qquad N\backslash S \Rightarrow N\backslash S}{(N\backslash S)/(N{\oplus}(CN/CN)),N \Rightarrow N\backslash S}/L \qquad \dfrac{\dfrac{CN/CN \Rightarrow CN/CN}{CN/CN \Rightarrow N{\oplus}(CN/CN)}\oplus R_2 \qquad N\backslash S \Rightarrow N\backslash S}{(N\backslash S)/(N{\oplus}(CN/CN)),CN/CN \Rightarrow N\backslash S}/L$$

3 Focalisation for DA

In focalisation situated (antecedent, input, $^\bullet$/ succedent, output, $^\circ$) non-atomic types are classified as of negative (asynchronous) or positive (synchronous) *polarity* according as their rule is reversible or not; situated atoms are positive or negative according to their bias. The table below summarizes the notational convention on formulas P, Q, M and N:

	input	output
sync.	**Q**	**P**
async.	**M**	**N**

The grammar of these types polarised with respect to input and output occurrences is as follows; Q and P denote synchronous formulas in input and output position respectively, whereas M and N denote asynchronous formulas in input and output position respectively (in the nonatomic case we will abbreviate thus: *left sync., right synch., left async., and right async.*):

(4)
$$\begin{array}{lrcl}
\text{Positive output} & P & ::= & \mathrm{At}^+ \mid A \bullet B^\circ \mid I^\circ \mid A \odot_k B^\circ \mid J^\circ \mid A \oplus B^\circ \\
\text{Positive input} & Q & ::= & \mathrm{At}^- \mid C/B^\bullet \mid A \backslash C^\bullet \mid C \uparrow_k B^\bullet \mid A \downarrow_k C^\bullet \mid A \& B^\bullet \\
\text{Negative output} & N & ::= & \mathrm{At}^- \mid C/B^\circ \mid A \backslash C^\circ \mid C \uparrow_k B^\circ \mid A \downarrow_k C^\circ \mid A \& B^\circ \\
\text{Negative input} & M & ::= & \mathrm{At}^+ \mid A \bullet B^\bullet \mid I^\bullet \mid A \odot_k B^\bullet \mid J^\bullet \mid A \oplus B^\bullet
\end{array}$$

Notice that if P occurs in the antecedent then this occurrence of P is negative, and so forth.

There are alternating phases of don't-care nondeterministic negative rule application, and positive rule application locking on to *focalised* formulas.

Given a sequent with no occurrences of negative formulas, one chooses a positive formula as principal formula (which is boxed; we say it is focalised) and applies proof search to its subformulas while these remain positive. When one finds a negative formula or a literal, invertible rules are applied in a don't care nondeterminitic fashion until no longer possible, when another positive formula is chosen, and so on.

A sequent is either unfocused and as before, or else focused and has exactly one type boxed. The focalised logical rules are given in Figures 4-11 including Curry-Howard categorial semantic labelling. Occurrences of P, Q, M and N are supposed not to be focalised, which means that their focalised occurrence *must* be signalled with a box. By contrast, occurrences of A, B, C may be focalised or not.

4 Completeness of focalisation for DA

We shall be dealing with three systems: the displacement calculus **DA** with sequents notated $\Delta \Rightarrow A$, the *weakly focalised* displacement calculus with additives $\mathbf{DA_{foc}}$ with sequents notated $\Delta \Longrightarrow_w A$, and the *strongly focalised* displacement calculus with additives $\mathbf{DA_{Foc}}$ with sequents notated $\Delta \Longrightarrow A$. Sequents of both $\mathbf{DA_{foc}}$ and $\mathbf{DA_{Foc}}$ may contain at most one focalised formula, possibly A. When a $\mathbf{DA_{foc}}$ sequent is notated $\Delta \Longrightarrow_w A \diamond$ foc, it means that the sequent possibly contains a (unique) focalised formula. Otherwise, $\Delta \Longrightarrow_w A$ means that the sequent does not contain a focus.

In this section we prove the strong focalisation property for the displacement calculus with additives **DA**.

The focalisation property for Linear Logic was discovered by [2]. In this paper we follow the proof idea from [12], which we adapt to the intuitionistic non-commutative case **DA** with twin multiplicative modes of combination, the continuous (concatenation) and the discontinuous (intercalation) products. The proof relies heavily on the Cut-elimination property for weakly focalised **DA** which is proved in

$$\frac{\overrightarrow{A}:x,\Gamma\Rightarrow C:\chi}{\Gamma\Rightarrow A\backslash C:\lambda x\chi}\backslash R \qquad \frac{\Gamma,\overrightarrow{B}:y\Rightarrow C:\chi}{\Gamma\Rightarrow C/B:\lambda y\chi}/R$$

$$\frac{\Delta\langle\overrightarrow{A}:x,\overrightarrow{B}:y\rangle\Rightarrow D:\omega}{\Delta\langle\overrightarrow{A\bullet B}:z\rangle\Rightarrow D:\omega\{\pi_1 z/x,\pi_2 z/y\}}\bullet L$$

$$\frac{\Delta\langle\Lambda\rangle\Rightarrow A:\phi}{\Delta\langle\overrightarrow{I}:x\rangle\Rightarrow A:\phi}IL$$

$$\frac{\overrightarrow{A}:x|_k\Gamma\Rightarrow C:\chi}{\Gamma\Rightarrow A|_k C:\lambda x\chi}\downarrow_k R \qquad \frac{\Gamma|_k\overrightarrow{B}:y\Rightarrow C:\chi}{\Gamma\Rightarrow C\uparrow_k B:\lambda y\chi}\uparrow_k R$$

$$\frac{\Delta\langle\overrightarrow{A}:x|_k\overrightarrow{B}:y\rangle\Rightarrow D:\omega}{\Delta\langle\overrightarrow{A\odot_k B}:z\rangle\Rightarrow D:\omega\{\pi_1 z/x,\pi_2 z/y\}}\odot_k L$$

$$\frac{\Delta\langle 1\rangle\Rightarrow A:\phi}{\Delta\langle\overrightarrow{J}:x\rangle\Rightarrow A:\phi}JL$$

Figure 4: Asynchronous multiplicative rules

$$\frac{\Gamma\Rightarrow A:\phi \qquad \Gamma\Rightarrow B:\psi}{\Gamma\Rightarrow A\&B:(\phi,\psi)}\&R$$

$$\frac{\Gamma\langle\overrightarrow{A}:x\rangle\Rightarrow C:\chi_1 \qquad \Gamma\langle\overrightarrow{B}:y\rangle\Rightarrow C:\chi_2}{\Gamma\langle\overrightarrow{A\oplus B}:z\rangle\Rightarrow C:z\rightarrow x.\chi_1;y.\chi_2}\oplus L$$

Figure 5: Asynchronous additive rules

$$\frac{\Gamma \Rightarrow \boxed{P}:\phi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma, \overrightarrow{\boxed{P\backslash Q}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\backslash L \qquad \frac{\Gamma \Rightarrow \boxed{P}:\phi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma, \overrightarrow{\boxed{P\backslash M}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\backslash L$$

$$\frac{\Gamma \Rightarrow N:\phi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma, \overrightarrow{\boxed{N\backslash Q}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\backslash L \qquad \frac{\Gamma \Rightarrow N:\phi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma, \overrightarrow{\boxed{N\backslash M}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\backslash L$$

$$\frac{\Gamma \Rightarrow \boxed{P}:\psi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{Q/P}}:x,\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}/L \qquad \frac{\Gamma \Rightarrow N:\psi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{Q/N}}:x,\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}/L$$

$$\frac{\Gamma \Rightarrow \boxed{P}:\psi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{M/P}}:x,\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}/L \qquad \frac{\Gamma \Rightarrow M:\psi \qquad \Delta\langle \overrightarrow{N}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{N/M}}:x,\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}/L$$

Figure 6: Left synchronous continuous multiplicative rules

$$\frac{\Gamma \Rightarrow \boxed{P}:\phi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma|_k \overrightarrow{\boxed{P\downarrow_k Q}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\downarrow_k L \qquad \frac{\Gamma \Rightarrow \boxed{P}:\phi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma|_k \overrightarrow{\boxed{P\downarrow_k M}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\downarrow_k L$$

$$\frac{\Gamma \Rightarrow N:\phi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma|_k \overrightarrow{\boxed{N\downarrow_k Q}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\downarrow_k L \qquad \frac{\Gamma \Rightarrow N:\phi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \Gamma|_k \overrightarrow{\boxed{N\downarrow_k M}}:y\rangle \Rightarrow D:\omega\{(y\ \phi)/z\}}\downarrow_k L$$

$$\frac{\Gamma \Rightarrow \boxed{P}:\psi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{Q\uparrow_k P}}:x|_k\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}\uparrow_k L \qquad \frac{\Gamma \Rightarrow N:\psi \qquad \Delta\langle \overrightarrow{\boxed{Q}}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{Q\uparrow_k N}}:x|_k\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}\uparrow_k L$$

$$\frac{\Gamma \Rightarrow \boxed{P}:\psi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{M\uparrow_k P}}:x|_k\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}\uparrow_k L \qquad \frac{\Gamma \Rightarrow N:\psi \qquad \Delta\langle \overrightarrow{M}:z\rangle \Rightarrow D:\omega}{\Delta\langle \overrightarrow{\boxed{M\uparrow_k N}}:x|_k\Gamma\rangle \Rightarrow D:\omega\{(x\ \psi)/z\}}\uparrow_k L$$

Figure 7: Left synchronous discontinuous multiplicative rules

$$\frac{\Gamma\langle\overrightarrow{\boxed{Q}}:x\rangle \Rightarrow C:\chi}{\Gamma\langle\overrightarrow{\boxed{Q\&B}}:z\rangle \Rightarrow C:\chi\{\pi_1 z/x\}}\,\&L_1 \qquad \frac{\Gamma\langle\overrightarrow{M}:x\rangle \Rightarrow C:\chi}{\Gamma\langle\overrightarrow{\boxed{M\&B}}:z\rangle \Rightarrow C:\chi\{\pi_1 z/x\}}\,\&L_1$$

$$\frac{\Gamma\langle\overrightarrow{\boxed{Q}}:y\rangle \Rightarrow C:\chi}{\Gamma\langle\overrightarrow{\boxed{A\&Q}}:z\rangle \Rightarrow C:\chi\{\pi_2 z/y\}}\,\&L_2 \qquad \frac{\Gamma\langle\overrightarrow{M}:y\rangle \Rightarrow C:\chi}{\Gamma\langle\overrightarrow{\boxed{A\&M}}:z\rangle \Rightarrow C:\chi\{\pi_2 z/y\}}\,\&L_2$$

Figure 8: Left synchronous additive rules

$$\frac{\Gamma_1 \Rightarrow \boxed{P_1}:\phi \qquad \Gamma_2 \Rightarrow \boxed{P_2}:\psi}{\Gamma_1,\Gamma_2 \Rightarrow \boxed{P_1 \bullet P_2}:(\phi,\psi)}\,\bullet R \qquad \frac{\Gamma_1 \Rightarrow \boxed{P}:\phi \qquad \Gamma_2 \Rightarrow N:\psi}{\Gamma_1,\Gamma_2 \Rightarrow \boxed{P \bullet N}:(\phi,\psi)}\,\bullet R$$

$$\frac{\Gamma_1 \Rightarrow N:\phi \qquad \Gamma_2 \Rightarrow \boxed{P}:\psi}{\Gamma_1,\Gamma_2 \Rightarrow \boxed{N \bullet P}:(\phi,\psi)}\,\bullet R \qquad \frac{\Gamma_1 \Rightarrow N_1:\phi \qquad \Gamma_2 \Rightarrow N_2:\psi}{\Gamma_1,\Gamma_2 \Rightarrow \boxed{N_1 \bullet N_2}:(\phi,\psi)}\,\bullet R$$

$$\frac{}{\Lambda \Rightarrow \boxed{I}:0}\,IR$$

Figure 9: Right synchronous continuous multiplicative rules

$$\frac{\Gamma_1 \Rightarrow \boxed{P_1}:\phi \qquad \Gamma_2 \Rightarrow \boxed{P_2}:\psi}{\Gamma_1 \mid_k \Gamma_2 \Rightarrow \boxed{P_1 \odot_k P_2}:(\phi,\psi)}\,\odot_k R \qquad \frac{\Gamma_1 \Rightarrow \boxed{P}:\phi \qquad \Gamma_2 \Rightarrow N:\psi}{\Gamma_1 \mid_k \Gamma_2 \Rightarrow \boxed{P \odot_k N}:(\phi,\psi)}\,\odot_k R$$

$$\frac{\Gamma_1 \Rightarrow N:\phi \qquad \Gamma_2 \Rightarrow \boxed{P}:\psi}{\Gamma_1 \mid_k \Gamma_2 \Rightarrow \boxed{N \odot_k P}:(\phi,\psi)}\,\odot_k R \qquad \frac{\Gamma_1 \Rightarrow N_1:\phi \qquad \Gamma_2 \Rightarrow N_2:\psi}{\Gamma_1 \mid_k \Gamma_2 \Rightarrow \boxed{N_1 \odot_k N_2}:(\phi,\psi)}\,\odot_k R$$

$$\frac{}{1 \Rightarrow \boxed{J}:0}\,JR$$

Figure 10: Right synchronous discontinuous multiplicative rules

$$\frac{\Gamma \Rightarrow \boxed{P}:\phi}{\Gamma \Rightarrow \boxed{P \oplus B}:\iota_1\phi}\,\oplus R_1 \qquad \frac{\Gamma \Rightarrow N:\phi}{\Gamma \Rightarrow \boxed{N \oplus B}:\iota_1\phi}\,\oplus R_1$$

$$\frac{\Gamma \Rightarrow \boxed{P}:\psi}{\Gamma \Rightarrow \boxed{A \oplus P}:\iota_2\psi}\,\oplus R_2 \qquad \frac{\Gamma \Rightarrow N:\psi}{\Gamma \Rightarrow \boxed{A \oplus N}:\iota_2\psi}\,\oplus R_2$$

Figure 11: Right synchronous additive rules

the appendix. In our presentation of focalisation we have avoided the *react* rules of [2] and [3], and use instead a simpler, box, notation suitable for non-commutativity.

$\mathbf{DA_{Foc}}$ is a subsystem of $\mathbf{DA_{foc}}$. $\mathbf{DA_{foc}}$ has the focusing rules *foc* and Cut rules *p-Cut*$_1$, *p-Cut*$_2$, *n-Cut*$_1$ and *n-Cut*$_2$[3] shown in (5), and the synchronous and asynchronous rules displayed before, which are read as allowing in synchronous rules the occurrence of asynchronous formulas, and in asynchronous rules as allowing arbitrary sequents with possibly one focalised formula. $\mathbf{DA_{Foc}}$ has the focusing rules but not the Cut rules, and the synchronous and asynchronous rules displayed before, which are such that focalised sequents cannot contain any complex asynchronous formulas, whereas sequents with at least one complex asynchronous formula cannot contain a focalised formula. Hence, strongly focalised proof search operates in alternating asynchronous and synchronous phases. The weakly focalised calculus $\mathbf{DA_{foc}}$ is an intermediate logic which we use to prove the completeness of $\mathbf{DA_{Foc}}$ for **DA**.

(5) $$\frac{\Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w A}{\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w A}\,foc \qquad \frac{\Delta\Longrightarrow_w\boxed{P}}{\Delta\Longrightarrow_w P}\,foc$$

$$\frac{\Gamma\Longrightarrow_w\boxed{P} \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1 \qquad \frac{\Gamma\Longrightarrow_w N\diamond \text{foc} \qquad \Delta\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2$$

$$\frac{\Gamma\Longrightarrow_w P\diamond \text{foc} \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\rangle\Longrightarrow_w C\diamond \text{foc}}\,n\text{-}Cut_1 \qquad \frac{\Gamma\Longrightarrow_w N \qquad \Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\rangle\Longrightarrow_w C\diamond \text{foc}}\,n\text{-}Cut_2$$

4.1 Embedding of DA into $\mathbf{DA_{foc}}$

The identity axiom we consider for **DA** and for both $\mathbf{DA_{foc}}$ and $\mathbf{DA_{Foc}}$ is restricted to atomic types; recalling that atomic types are classified into positive bias At^+ and negative bias At^-:

(6) If $P\in At^+$, $\text{P}\Longrightarrow_w\boxed{P}$ and $\text{P}\Longrightarrow\boxed{P}$
If $Q\in At^-$, $\boxed{Q}\Longrightarrow_w\text{Q}$ and $\boxed{Q}\Longrightarrow\text{Q}$

In fact, the Identity rule holds of any type A. It has the following formulation in the sequent calculi considered here:

(7) $$\begin{cases} \overrightarrow{A}\Rightarrow A & & \text{in } \mathbf{DA} \\ \overrightarrow{P}\Longrightarrow_w\boxed{P} & \overrightarrow{\boxed{N}}\Longrightarrow_w N & \text{in } \mathbf{DA_{foc}} \\ \overrightarrow{P}\Longrightarrow P & \overrightarrow{N}\Longrightarrow N & \text{in } \mathbf{DA_{Foc}} \end{cases}$$

The Identity axiom for arbitrary types is also known as *Eta-expansion*. Eta-expansion is easy to prove in both **DA** and $\mathbf{DA_{foc}}$, but the same is not the case for $\mathbf{DA_{Foc}}$. This is the reason to consider what we have called weak focalisation, which helps us to prove smoothly this crucial property for the proof of strong focalisation.

Theorem 4.1 (Embedding of **DA** into $\mathbf{DA_{foc}}$) *For any configuration* Δ *and type A, we have that if* $\Delta\Rightarrow A$ *then* $\Delta\Longrightarrow_w A$.

Proof. We proceed by induction on the length of the derivation of **DA** proofs. In the following lines, we apply the induction hypothesis (i.h.) for each premise of **DA** rules (with the exception of the Identity rule and the right rules of units):

- Identity axiom:

[3] If it is convenient, we may drop the subscripts.

$$(8)\ \frac{\overrightarrow{P}\Longrightarrow_w \boxed{\mathrm{P}}}{\overrightarrow{P}\Longrightarrow_w P}\,foc \qquad \frac{\overrightarrow{\boxed{\mathrm{N}}}\Longrightarrow_w N}{\overrightarrow{N}\Longrightarrow_w N}\,foc$$

- Cut rule: just apply *n-Cut*.
- Units

$$(9)\ \frac{}{\Lambda\Rightarrow I}\,IR \quad \rightsquigarrow \quad \frac{\dfrac{}{\Lambda\Longrightarrow_w \boxed{I}}\,IR}{\Lambda\Longrightarrow_w I}\,foc$$

$$(10)\ \frac{}{1\Rightarrow J}\,JR \quad \rightsquigarrow \quad \frac{\dfrac{}{1\Longrightarrow_w \boxed{J}}\,JR}{1\Longrightarrow_w J}\,foc$$

Left unit rules apply as in the case of **DA**.

- Left discontinuous product: directly translates.
- Right discontinuous product. There are cases $P_1\odot_k P_2$, $N_1\odot_k N_2$, $N\odot_k P$ and $P\odot_k N$. We show one representative example:

$$\frac{\Delta\Rightarrow P \qquad \Gamma\Rightarrow N}{\Delta|_k\Gamma\Rightarrow P\odot_k N}\,\odot_k R \quad \rightsquigarrow \quad \dfrac{\Gamma\Longrightarrow_w N \qquad \dfrac{\Delta\Longrightarrow_w P \qquad \dfrac{\overrightarrow{P}\Longrightarrow_w \boxed{P} \qquad \dfrac{\overrightarrow{\boxed{N}}\Longrightarrow_w N}{\overrightarrow{N}\Longrightarrow_w N}\,foc}{\overrightarrow{P}|_k\overrightarrow{N}\Rightarrow \boxed{P\odot_k N}}\,\odot_k R}{\Delta|_k\overrightarrow{N}\Longrightarrow_w \boxed{P\odot_k N}}\,n\text{-}Cut_2}{\dfrac{\Delta|_k\Gamma\Longrightarrow_w \boxed{P\odot_k N}}{\Delta|_k\Gamma\Longrightarrow_w P\odot_k N}\,foc}\,n\text{-}Cut_2$$

- Left discontinuous $\uparrow_k$ rule (the left rule for $\downarrow_k$ is entirely similar). Like in the case for the right discontinuous product $\odot_k$ rule, we only show one representative example:

$$\frac{\Gamma\Rightarrow P \qquad \Delta\langle\overrightarrow{N}\rangle\Rightarrow A}{\Delta\langle\overrightarrow{N\uparrow_k P}|_k\Gamma\rangle\Rightarrow A}\,\uparrow_k L \quad \rightsquigarrow \quad \dfrac{\Gamma\Longrightarrow_w P \qquad \dfrac{\dfrac{\overrightarrow{P}\Longrightarrow_w \boxed{\mathrm{P}} \qquad \overrightarrow{\boxed{N}}\Longrightarrow_w N}{\overrightarrow{\boxed{N\uparrow_k P}}|_k\overrightarrow{P}\Longrightarrow_w N}\,\uparrow_k L \qquad \Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w A}{\Delta\langle\overrightarrow{\boxed{N\uparrow_k P}}|_k\overrightarrow{P}\rangle\Longrightarrow_w A}\,n\text{-}Cut_1}{\dfrac{\Delta\langle\overrightarrow{\boxed{N\uparrow_k P}}|_k\Gamma\rangle\Longrightarrow_w A}{\Delta\langle\overrightarrow{N\uparrow_k P}|_k\Gamma\rangle\Longrightarrow_w A}\,foc}\,n\text{-}Cut_2$$

- Right discontinuous $\uparrow_k$ rule (the right discontinuous rule for $\downarrow_k$ is entirely similar):

$$(11)\ \frac{\Delta|_k\overrightarrow{A}\Rightarrow B}{\Delta\Rightarrow B\uparrow_k A}\,\uparrow_k R \quad \rightsquigarrow \quad \frac{\Delta|_k\overrightarrow{A}\Longrightarrow_w B}{\Delta\Longrightarrow_w B\uparrow_k A}\,\uparrow_k R$$

- Product and implicative continuous rules. These follow the same pattern as the discontinuous case. We interchange the metalinguistic k-th intercalation $|_k$ with the metalinguistic concatenation ',', and we interchange $\odot_k$, $\uparrow_k$ and $\downarrow_k$ with $\bullet$, $/$, and $\backslash$ respectively.

Concerning additives, conjunction Right translates directly and we consider then conjunction Left (disjunction is symmetric):

$$(12)\quad \frac{\Delta\langle \overrightarrow{P}\rangle \Rightarrow C}{\Delta\langle \overrightarrow{P\&M}\rangle \Rightarrow C}\,\&L \quad \rightsquigarrow \quad \frac{\overrightarrow{P\&M}\Longrightarrow_w P \qquad \Delta\langle \overrightarrow{P}\rangle \Longrightarrow_w C}{\Delta\langle \overrightarrow{P\&M}\rangle \Longrightarrow_w C}\,n\text{-}Cut_1$$

where by Eta expansion and application of the *foc* rule, we have $\overrightarrow{P\&M}\Longrightarrow_w P$. □

4.2 Embedding of $\mathbf{DA_{foc}}$ into $\mathbf{DA_{Foc}}$

Theorem 4.2 (Embedding of $\mathbf{DA_{foc}}$ into $\mathbf{DA_{Foc}}$) *For any configuration Δ and type A, we have that if $\Delta\Longrightarrow_w A$ with one focalised formula and no asynchronous formula occurrence, then $\Delta\Longrightarrow A$ with the same formula focalised. If $\Delta\Longrightarrow_w A$ with no focalised formula and with at least one asynchronous formula, then $\Delta\Longrightarrow A$.*

Proof. We proceed by induction on the size of $\mathbf{DA_{foc}}$ sequents.[4] We consider Cut-free $\mathbf{DA_{foc}}$ proofs which match the sequents of this theorem. If the last rule is logical (i.e., it is not an instance of the *foc* rule) the i.h. applies directly and we get $\mathbf{DA_{Foc}}$ proofs of the same end-sequent. Now, let us suppose that the last rule is not logical, i.e. it is an instance of the *foc* rule. Let us suppose that the end sequent $\Delta\Longrightarrow_w A$ is a synchronous sequent. Suppose for example that the focalised formula is in the succedent:

$$(13)\quad \frac{\Delta\Longrightarrow_w \boxed{P}}{\Delta\Longrightarrow_w P}\,foc$$

The sequent $\Delta\Longrightarrow_w \boxed{P}$ arises from a synchronous rule to which we can apply i.h.. Let us suppose now that the end-sequent contains at least one asynchronous formula. We see three cases which are illustrative:

$$(14)\quad \text{a.}\quad \Delta\langle\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w \boxed{P}$$
$$\text{b.}\quad \Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w B\uparrow_k A$$
$$\text{c.}\quad \Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w A\&B$$

We have by Eta expansion that $\overrightarrow{A\odot_k B}\Longrightarrow_w \overrightarrow{\boxed{A\odot_k B}}$. We apply to this sequent the invertible $\odot_k$ left rule, whence $\overrightarrow{A}\,|_k\,\overrightarrow{B}\Longrightarrow_w \overrightarrow{\boxed{A\odot_k B}}$. In case (14a), we have the following proof in $\mathbf{DA_{foc}}$:

$$(15)\quad \frac{\dfrac{\overrightarrow{A}\,|_k\,\overrightarrow{B}\Longrightarrow_w \overrightarrow{\boxed{A\odot_k B}} \qquad \Delta\langle\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w \boxed{P}}{\Delta\langle\overrightarrow{A}\,|_k\,\overrightarrow{B}\rangle\Longrightarrow_w \boxed{P}}\,p\text{-}Cut_1}{\Delta\langle\overrightarrow{A}\,|_k\,\overrightarrow{B}\rangle\Longrightarrow_w P}\,foc$$

[4] For a given type A, the *size* of A, $|A|$, is the number of connectives in A. By recursion on configurations we have:

$$\begin{aligned} |\Lambda| &::= 0\\ |\overrightarrow{A},\Delta| &::= |A|+|\Delta|, \text{for } sA=0\\ |1,\Delta| &::= |\Delta|\\ |A\{\Delta_1:\cdots:\Delta_{sA}\}| &::= |A|+\sum_{i=1}^{sA}|\Delta_i| \end{aligned}$$

Moreover, we have:

$|\Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w A| = |\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w A|$

$|\Delta\Longrightarrow_w \boxed{P}| = |\Delta\Longrightarrow_w P|$

To the above $\mathbf{DA_{foc}}$ proof we apply Cut-elimination and we get the Cut-free $\mathbf{DA_{foc}}$ end-sequent $\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle \Longrightarrow_w P$. We have $|\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w P| < |\Delta\langle\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w P|$. We can apply then i.h. and we derive the provable $\mathbf{DA_{Foc}}$ sequent $\Delta\langle\overrightarrow{A}\,|_k\,\overrightarrow{B}\rangle\Longrightarrow P$ to which we can apply the left $\odot_k$ rule. We have obtained $\Delta\langle\overrightarrow{A\odot_k B}\rangle\Longrightarrow P$. In the same way, we have that $\overrightarrow{\boxed{B\uparrow_k A}}|_k\overrightarrow{A}\Longrightarrow_w B$. Thus, in case (14b), we have the following proof in $\mathbf{DA_{foc}}$:

$$(16)\quad \dfrac{\dfrac{\Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w B\uparrow_k A \qquad \overrightarrow{\boxed{B\uparrow_k A}}|_k\overrightarrow{A}\Longrightarrow_w B}{\Delta\langle\overrightarrow{\boxed{Q}}\rangle\,|_k\,\overrightarrow{A}\Longrightarrow_w B}\,p\text{-}Cut_2}{\Delta\langle\overrightarrow{Q}\rangle\,|_k\,\overrightarrow{A}\Longrightarrow_w B}\,foc$$

As before, we apply Cut-elimination to the above proof. We get the Cut-free $\mathbf{DA_{foc}}$ end-sequent $\Delta\langle\overrightarrow{Q}\rangle|_k\overrightarrow{A}\Longrightarrow_w B$. It has size less than $|\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w B\uparrow_k A|$. We can apply i.h. and we get the $\mathbf{DA_{Foc}}$ provable sequent $\Delta\langle\overrightarrow{Q}\rangle|_k\overrightarrow{A}\Longrightarrow B$ to which we apply the $\uparrow_k$ right rule.

In case (14c):

$$(17)\quad \dfrac{\Delta\langle\overrightarrow{\boxed{\mathrm{Q}}}\rangle\Longrightarrow_w A\&B}{\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w A\&B}\,foc$$

by applying the *foc* rule and the invertibility of $\&R$ we get the provable $\mathbf{DA_{foc}}$ sequents $\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w A$ and $\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w B$. These sequents have smaller size than $\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w A\&B$. The aforementioned sequents have a Cut-free proof in $\mathbf{DA_{foc}}$. We apply i.h. and we get $\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow A$ and $\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow B$. We apply the & right rule in $\mathbf{DA_{Foc}}$, and we get $\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow A\&B$. □

By this theorem we obtain the completeness of strong focalisation.

5 Example

We can have coordinate unlike types with nominal and adjectival complementation of *is*:

(18) [**Tully**]+**is**+[[**Cicero**+**and**+**humanist**]] : Sf

Lexical lookup of types yields:

(19) $[\blacksquare Nt(s(m)) : b], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus(\exists g(CNg/CN))$:
$\lambda A\lambda B(Pres\ (A\to C.[B=C]; D.((D\ \lambda E[E=B])\ B))), [[\blacksquare\forall gNt(s(g)) : 007,$
$\blacksquare\forall f\forall a((\blacksquare((((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf))\backslash[]^{-1}[]^{-1}((((\langle\rangle Na\backslash Sf)/$
$(\exists bNb\oplus\exists g(CNg/CN))\backslash(\langle\rangle Na\backslash Sf)))/\blacksquare((((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf))) :$
$\lambda F\lambda G\lambda H\lambda I[((G\ H)\ I)\wedge((F\ H)\ I)], \Box\forall n(CNn/CNn) : \hat{}\lambda J\lambda K[(J\ K)\wedge(\check{}teetotal\ K)]]]\ \Rightarrow\ Sf$

The bracket modalities $\langle\rangle$ and $[]^{-1}$mark as syntactic domains subjects and coordinate structures which are weak and strong islands respectively. The quantifiers and first-order structure mark agreement features such as third person singular for any gender for *is*. The normal modality □ marks semantic intensionality and ■ marks rigid designator semantic intensionality. The example has positive and negative additive disjunction so that the derivation in Figures 12–16 illustrates both synchronous and asynchronous focusing additives. This delivers the correct semantics: $[(Pres\ [t=c])\wedge(Pres\ (\check{}humanist\ t))]$.

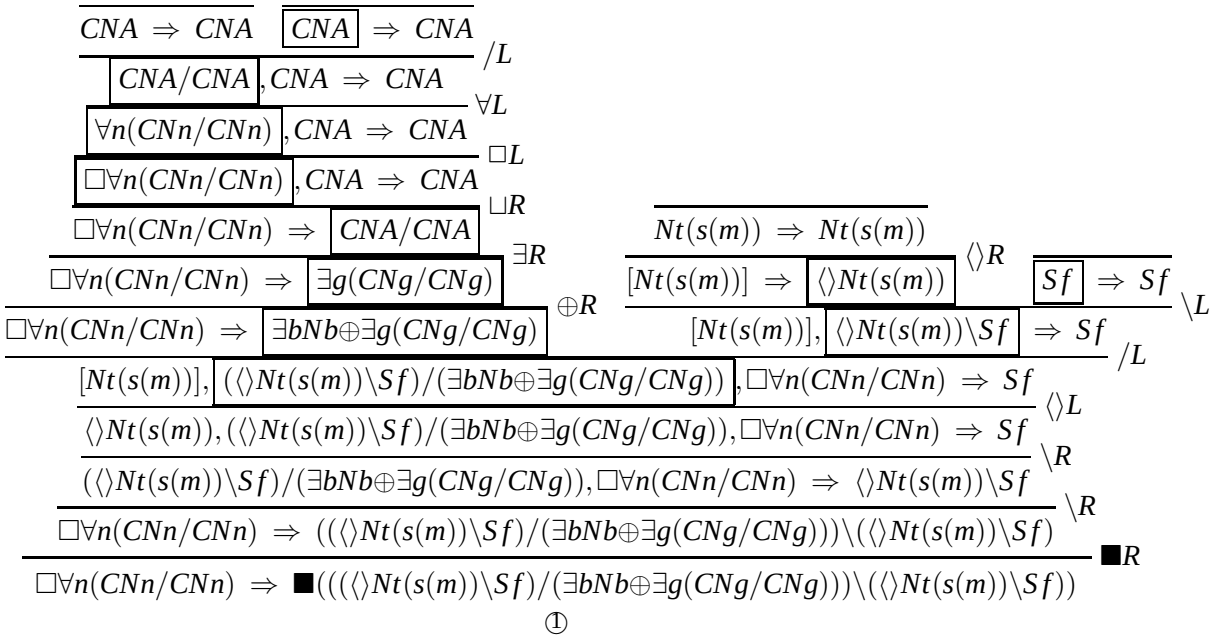


Figure 12: Coordination of unlike types, Part I

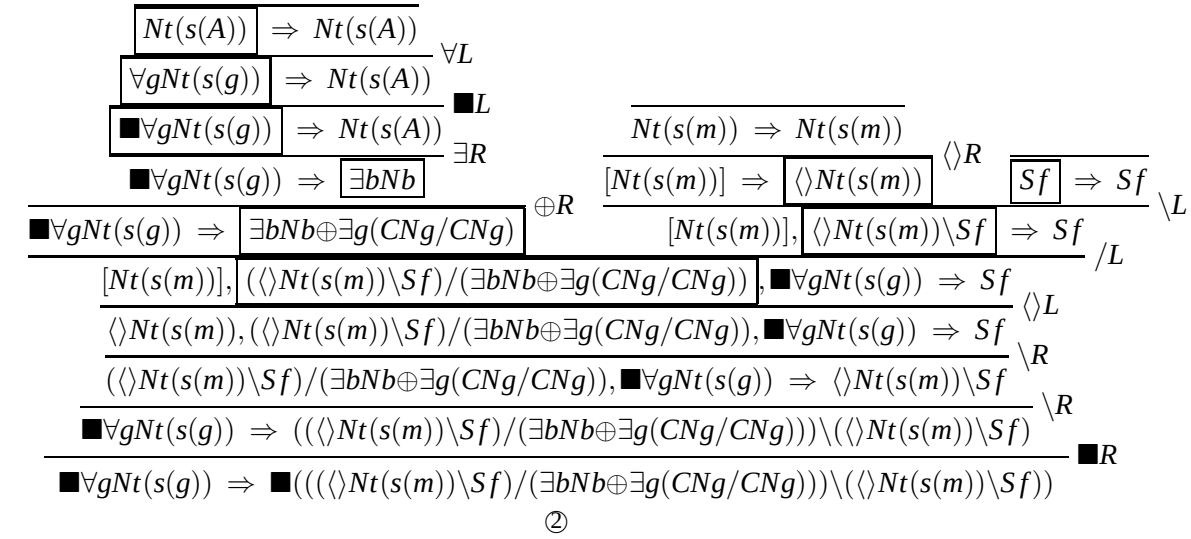


Figure 13: Coordination of unlike types, Part II

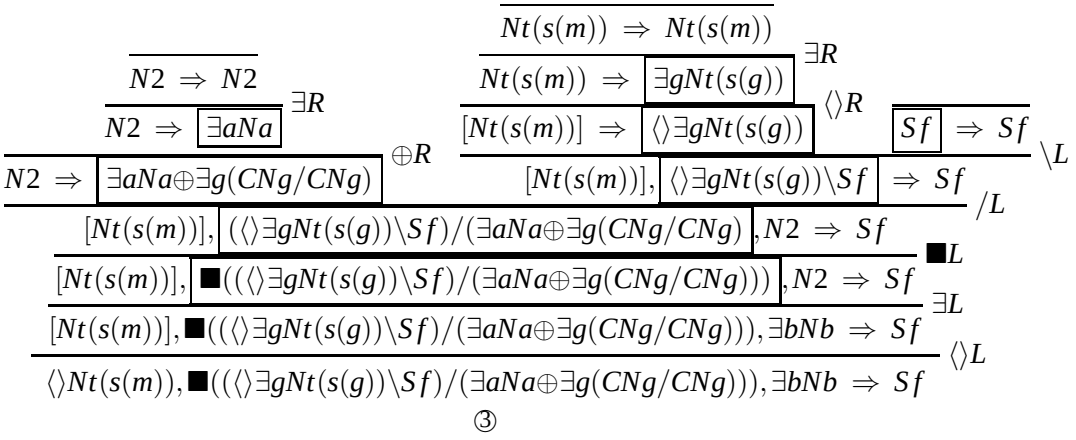


Figure 14: Coordination of unlike types, Part III

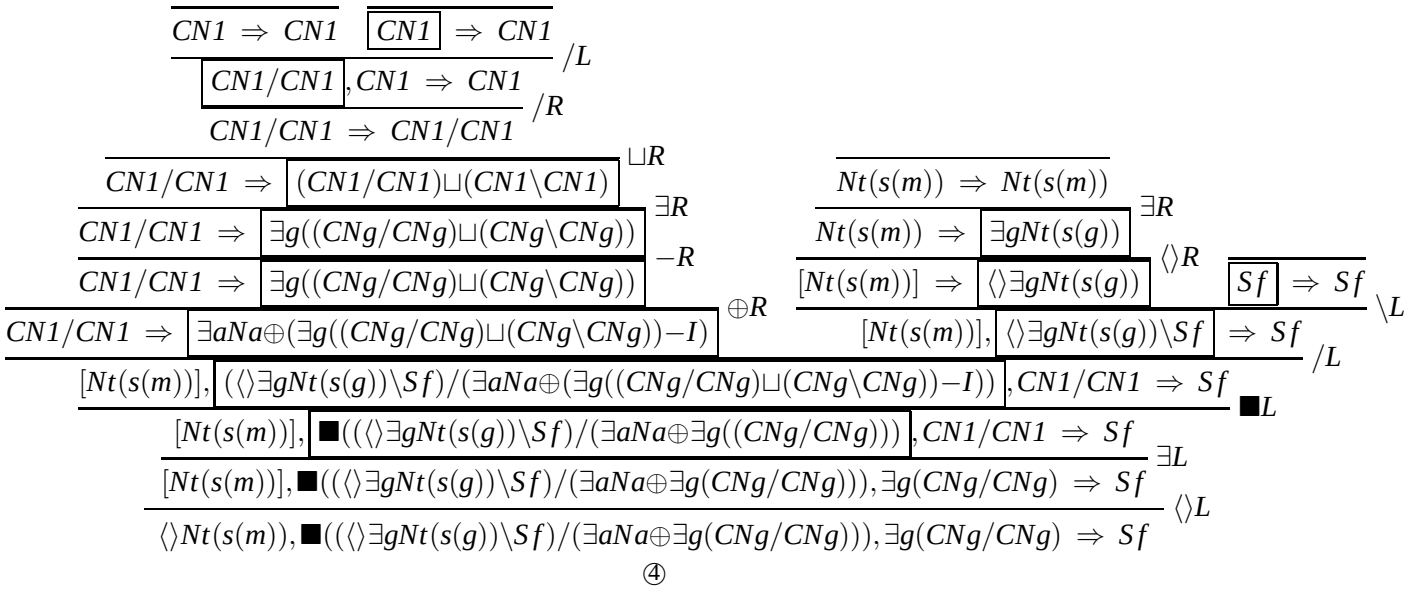


Figure 15: Coordination of unlike types, Part IV

③ ④

$$\frac{\boxed{Nt(s(m))} \Rightarrow Nt(s(m))}{\boxed{\blacksquare Nt(s(m))} \Rightarrow Nt(s(m))}\ \blacksquare L$$

$$\frac{\boxed{\blacksquare Nt(s(m))} \Rightarrow Nt(s(m))}{[\blacksquare Nt(s(m))] \Rightarrow \boxed{\langle\rangle Nt(s(m))}}\ \langle\rangle R \qquad \frac{}{\boxed{Sf} \Rightarrow Sf}$$

$$\frac{[\blacksquare Nt(s(m))] \Rightarrow \boxed{\langle\rangle Nt(s(m))} \qquad \boxed{Sf} \Rightarrow Sf}{[\blacksquare Nt(s(m))], \boxed{\langle\rangle Nt(s(m))\backslash Sf} \Rightarrow Sf}\ \backslash L$$

$$\frac{}{\langle\rangle Nt(s(m)), \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), \exists bNb\oplus\exists g(CNg/CNg) \Rightarrow Sf}\ \oplus L$$

$$\frac{\langle\rangle Nt(s(m)), \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), \exists bNb\oplus\exists g(CNg/CNg) \Rightarrow Sf}{\blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), \exists bNb\oplus\exists g(CNg/CNg) \Rightarrow \langle\rangle Nt(s(m))\backslash Sf}\ \backslash R$$

$$\frac{\blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), \exists bNb\oplus\exists g(CNg/CNg) \Rightarrow \langle\rangle Nt(s(m))\backslash Sf}{\blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg)) \Rightarrow (\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg))}\ /R$$

$$\frac{\cdots \qquad [\blacksquare Nt(s(m))], \boxed{\langle\rangle Nt(s(m))\backslash Sf} \Rightarrow Sf}{[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), \boxed{((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Nt(s(m))\backslash Sf)} \Rightarrow Sf}\ \backslash L$$

$$\frac{\cdots}{[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [\boxed{[\,]^{-1}(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg))))\backslash(\langle\rangle Nt(s(m))\backslash Sf))}] \Rightarrow Sf}\ [\,]^{-1}L$$

②

$$\frac{\cdots}{[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [[\boxed{[\,]^{-1}[\,]^{-1}(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg))))\backslash(\langle\rangle Nt(s(m))\backslash Sf))}]] \Rightarrow Sf}\ [\,]^{-1}L$$

①

$$\frac{\cdots}{\begin{array}{c}[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [[\blacksquare\forall gNt(s(g)),\\ \boxed{\blacksquare(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Nt(s(m))\backslash Sf))\backslash[\,]^{-1}[\,]^{-1}(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Nt(s(m))\backslash Sf))}]] \Rightarrow Sf\end{array}}\ \backslash L$$

$$\frac{\cdots}{\begin{array}{c}[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [[\blacksquare\forall gNt(s(g)),\\ \boxed{(\blacksquare(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Nt(s(m))\backslash Sf))\backslash[\,]^{-1}[\,]^{-1}(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Nt(s(m))\backslash Sf)))/\blacksquare(((\langle\rangle Nt(s(m))\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Nt(s(m))\backslash Sf))},\\ \Box\forall n(CNn/CNn)]] \Rightarrow Sf\end{array}}\ /L$$

$$\frac{\cdots}{\begin{array}{c}[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [[\blacksquare\forall gNt(s(g)),\\ \boxed{\forall a((\blacksquare(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf))\backslash[\,]^{-1}[\,]^{-1}(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf)))/\blacksquare(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf)))},\\ \Box\forall n(CNn/CNn)]] \Rightarrow Sf\end{array}}\ \forall L$$

$$\frac{\cdots}{\begin{array}{c}[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [[\blacksquare\forall gNt(s(g)),\\ \boxed{\forall f\forall a((\blacksquare(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf))\backslash[\,]^{-1}[\,]^{-1}(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg))))\backslash(\langle\rangle Na\backslash Sf)))/\blacksquare(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf)))},\\ \Box\forall n(CNn/CNn)]] \Rightarrow Sf\end{array}}\ \forall L$$

(

$$\begin{array}{c}[\blacksquare Nt(s(m))], \blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus\exists g(CNg/CNg))), [[\blacksquare\forall gNt(s(g)),\\ \boxed{\blacksquare\forall f\forall a((\blacksquare(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf))\backslash[\,]^{-1}[\,]^{-1}(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf)))/\blacksquare(((\langle\rangle Na\backslash Sf)/(\exists bNb\oplus\exists g(CNg/CNg)))\backslash(\langle\rangle Na\backslash Sf)))},\\ \Box\forall n(CNn/CNn)]] \Rightarrow Sf\end{array}$$

References

[1] Vito Michele Abrusci & Roberto Maieli (2015): *Cyclic Multiplicative-Additive Proof Nets of Linear Logic with an Application to Language Parsing*. In Annie Foret, Glyn Morrill, Reinhard Muskens & Rainer Osswald, editors: *Preproceedings of the 20th Conference on Formal Grammar*, ESSLLI 2015, Barcelona, pp. 39–54.

[2] J. M. Andreoli (1992): *Logic programming with focusing in linear logic*. *Journal of Logic and Computation* 2(3), pp. 297–347, doi:10.1093/logcom/2.3.297.

[3] Kaustuv Chaudhuri (2006): *The Focused Inverse Method for Linear Logic*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA. AAI3248489.

[4] Kaustuv Chaudhuri, Dale Miller & Alexis Saurin (2008): *Canonical Sequent Proofs via Multi-Focusing*. In: *Fifth IFIP International Conference On Theoretical Computer Science - TCS 2008, IFIP 20th World Computer Congress, TC 1, Foundations of Computer Science, September 7-10, 2008, Milano, Italy*, pp. 383–396. Available at http://dx.doi.org/10.1007/978-0-387-09680-3_26.

[5] Mario Fadda (2010): *Geometry of Grammar: Exercises in Lambek Style*. Ph.D. thesis, Universitat Politècnica de Catalunya, Barcelona.

[6] H. Hendriks (1993): *Studied flexibility. Categories and types in syntax and semantics*. Ph.D. thesis, Universiteit van Amsterdam, ILLC, Amsterdam.

[7] Dominic J. D. Hughes & Rob J. van Glabbeck (2005): *Proof nets for unit-free multiplicative-additive linear logic*. *ACM Transactions on Computational Logic (TOCL)* 6(4), pp. 784–842, doi:10.1145/1094622.1094629.

[8] M. Kanazawa (1992): *The Lambek calculus enriched with additional connectives*. *Journal of Logic, Language and Information* 1, pp. 141–171, doi:10.1007/BF00171695.

[9] J. Lambek (1961): *On the Calculus of Syntactic Types*. In Roman Jakobson, editor: *Structure of Language and its Mathematical Aspects, Proceedings of the Symposia in Applied Mathematics XII*, American Mathematical Society, Providence, Rhode Island, pp. 166–178, doi:10.1090/psapm/012/9972.

[10] J. Lambek (1988): *Categorial and Categorical Grammars*. In Richard T. Oehrle, Emmon Bach & Deidre Wheeler, editors: *Categorial Grammars and Natural Language Structures*, *Studies in Linguistics and Philosophy* 32, D. Reidel, Dordrecht, pp. 297–317, doi:10.1007/978-94-015-6878-411.

[11] Joachim Lambek (1958): *The mathematics of sentence structure*. *American Mathematical Monthly* 65, pp. 154–170, doi:10.2307/2310058.

[12] Olivier Laurent (2004): *A proof of the Focalization property of Linear Logic*. Unpublished manuscript, CNRS - Université Paris VII.

[13] Richard Moot (2014): *Extended Lambek Calculi and First-Order Linear Logic*. In Michael Moortgat Claudia Casadio, Bob Coeke & Philip Scott, editors: *Categories and Types in Logic, Language and Physics: Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday*, *LNCS, FoLLI Publications in Logic, Language and Information* 8222, Springer, Berlin, pp. 297–330. Available at http://dx.doi.org/10.1007/978-3-642-54789-8_17.

[14] Richard Moot & Christian Retoré (2012): *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*. Springer, Heidelberg, doi:10.1007/978-3-642-31555-8.

[15] G. Morrill (1990): *Grammar and Logical Types*. In Martin Stockhof & Leen Torenvliet, editors: *Proceedings of the Seventh Amsterdam Colloquium*, pp. 429–450.

[16] Glyn Morrill & Oriol Valentín (2010): *Displacement Calculus*. *Linguistic Analysis* 36(1–4), pp. 167–192. Available at http://arxiv.org/abs/1004.4181. Special issue Festschrift for Joachim Lambek.

[17] Glyn Morrill, Oriol Valentín & Mario Fadda (2011): *The Displacement Calculus*. *Journal of Logic, Language and Information* 20(1), pp. 1–48, doi:10.1007/s10849-010-9129-2.

[18] Glyn V. Morrill (2011): *Categorial Grammar: Logical Syntax, Semantics, and Processing*. Oxford University Press, New York and Oxford.

[19] Robert J. Simmons (2012): *Substructural Logical Specifications*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh.

Appendix: Cut Elimination

We prove this by induction on the complexity (d,h) of top-most instances of *Cut*, where d is the size[5] of the cut formula and h is the length of the derivation the last rule of which is the Cut rule. There are four cases to consider: Cut with axiom in the minor premise, Cut with axiom in the major premise, principal Cuts, and permutation conversions. In each case, the complexity of the Cut is reduced. In order to save space, we will not be exhaustive showing all the cases because many follow the same pattern. In particular, for any synchronous logical rule there are always four cases to consider corresponding to the polarity of the subformulas. Here, and in the following, we will show only one representative example. Concerning continuous and discontinuous formulas, we will show only the discontinuous cases (discontinuous connectives are less known than the continuous ones of the plain Lambek Calculus). For the continuous instances, the reader has only to interchange the meta-linguistic wrap $|_k$ with the meta-linguistic concatenation $',',$ $\odot_k$ with $\bullet$, $\uparrow_k$ with $/$ and $\downarrow_k$ with $\backslash$. The units cases (principal case and permutation conversion cases) are completely trivial.

Proof. - *Id* cases:

$$(20)\ \frac{\overrightarrow{P}\Longrightarrow_w \boxed{P} \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w B\diamond \text{ foc}}{\Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w B\diamond \text{ foc}}\ p\text{-}Cut_1 \quad \rightsquigarrow \quad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w B\diamond \text{ foc}$$

$$\frac{\Delta\Longrightarrow_w N\diamond \text{ foc} \qquad \overrightarrow{\boxed{N}}\Longrightarrow_w N}{\Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w B\diamond \text{ foc}}\ p\text{-}Cut_2 \quad \rightsquigarrow \quad \Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w B\diamond \text{ foc}$$

The attentive reader may have wondered whether the following *Id* case could arise:

$$(21)\ \frac{\overrightarrow{\boxed{Q}}\Rightarrow Q \qquad \Gamma\langle\overrightarrow{Q}\rangle\Rightarrow A}{\Gamma\langle\boxed{\mathrm{Q}}\rangle\Rightarrow A}\ n\text{-}Cut_i$$

If Q were a primitive type q, and Γ were not the empty context, we would have then a Cut-free underivable sequent. For example, if the right premise of the Cut rule in (21) were the derivable sequent $q, q\backslash s\Rightarrow s$, we would have then as conclusion:

$$(22)\ \boxed{\mathrm{q}}, q\backslash s\Rightarrow s$$

Since the primitive type q in the antecedent is focalised, there is no possibility of applying the $\backslash$ left rule, which is a synchronous rule that needs that its active formula to be focalised. Principal cases:

• *foc* cases:

$$(23)\ \frac{\dfrac{\Delta\Longrightarrow_w \boxed{P}}{\Delta\Longrightarrow_w P}\ foc \qquad \Gamma\langle\overrightarrow{P}\rangle\Longrightarrow_w A\diamond \text{ foc}}{\Gamma\langle\Delta\rangle\Longrightarrow_w A\diamond \text{ foc}}\ n\text{-}Cut_1 \quad \rightsquigarrow \quad \frac{\Delta\Longrightarrow_w \boxed{P} \qquad \Gamma\langle\overrightarrow{P}\rangle\Longrightarrow_w A\diamond \text{ foc}}{\Gamma\langle\Delta\rangle\Longrightarrow_w A\diamond \text{ foc}}\ p\text{-}Cut_1$$

[5] The size of $|A|$ is the number of connectives appearing in A.

$$(24)\quad \dfrac{\Delta\Longrightarrow_w N \qquad \dfrac{\Delta\langle\boxed{N}\rangle\Longrightarrow_w A}{\Gamma\langle\overrightarrow{N}\rangle\Longrightarrow_w A}\,foc}{\Gamma\langle\Delta\rangle\Longrightarrow_w A}\,n\text{-}Cut_2 \quad\rightsquigarrow\quad \dfrac{\Delta\Longrightarrow_w N \qquad \Gamma\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w A}{\Gamma\langle\Delta\rangle\Longrightarrow_w A}\,p\text{-}Cut_2$$

• logical connectives:

$$(25)\quad \dfrac{\dfrac{\Delta|_k\overrightarrow{P_1}\Longrightarrow_w P_2\diamond\,\text{foc}}{\Delta\Longrightarrow_w P_2\uparrow_k P_1\diamond\,\text{foc}}\uparrow_k R \qquad \dfrac{\Gamma_1\Longrightarrow_w\boxed{P_1} \qquad \Gamma_2\langle\overrightarrow{P_2}\rangle\Longrightarrow_w A}{\Gamma_2\langle\overrightarrow{\boxed{P_2\uparrow_k P_1}}|_k\Gamma_1\rangle\Longrightarrow_w A}\uparrow_k L}{\Gamma_2\langle\Delta|_k\Gamma_1\rangle\Longrightarrow_w A\diamond\,\text{foc}}\,p\text{-}Cut_2 \quad\rightsquigarrow$$

$$\dfrac{\Gamma_1\Longrightarrow_w\boxed{P_1} \qquad \dfrac{\Delta|_k\overrightarrow{P_1}\Longrightarrow P_2\diamond\,\text{foc} \qquad \Gamma_2\langle\overrightarrow{P_2}\rangle\Longrightarrow_w A}{\Gamma_2\langle\Delta|_k\overrightarrow{P_1}\rangle\Longrightarrow_w A\diamond\,\text{foc}}\,n\text{-}Cut_1}{\Gamma_2\langle\Delta|_k\Gamma_1\rangle\Longrightarrow_w A\diamond\,\text{foc}}\,p\text{-}Cut_1$$

The case of $\downarrow_k$ is entirely similar to the $\uparrow_k$ case.
The case of $\downarrow_k$ is entirely similar to the $\uparrow_k$ case.

$$(26)\quad \dfrac{\dfrac{\Delta_1\Longrightarrow_w\boxed{P} \qquad \Delta_2\Longrightarrow_w N}{\Delta_1|_k\Delta_2\Longrightarrow_w\boxed{P\odot_k N}}\odot_k R \qquad \dfrac{\Gamma\langle\overrightarrow{P}|_k\overrightarrow{N}\rangle\Longrightarrow_w A\diamond\,\text{foc}}{\Gamma\langle\overrightarrow{P\odot_k N}\rangle\Longrightarrow_w A\diamond\,\text{foc}}\odot_k L}{\Gamma\langle\Delta_1|_k\Delta_2\rangle\Longrightarrow_w A\diamond\,\text{foc}}\,p\text{-}Cut_1 \quad\rightsquigarrow$$

$$\dfrac{\Delta_2\Longrightarrow_w N \qquad \dfrac{\Delta_1\Longrightarrow\boxed{P} \qquad \Gamma\langle\overrightarrow{P}|_k\overrightarrow{N}\rangle\Longrightarrow_w A\diamond\,\text{foc}}{\Gamma\langle\Delta_1|_k\overrightarrow{N}\rangle\Longrightarrow_w A\diamond\,\text{foc}}\,p\text{-}Cut_1}{\Gamma\langle\Delta_1|_k\Delta_2\rangle\Longrightarrow_w A\diamond\,\text{foc}}\,n\text{-}Cut_2$$

$$(27)\quad \dfrac{\dfrac{\Delta\Longrightarrow_w Q\diamond\,\text{foc} \qquad \Delta\Longrightarrow_w A\diamond\,\text{foc}}{\Delta\Longrightarrow_w Q\&A\diamond\,\text{foc}}\&R \qquad \dfrac{\Gamma\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w B}{\Gamma\langle\overrightarrow{\boxed{Q\&A}}\rangle\Longrightarrow_w B}\&L}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\diamond\,\text{foc}}\,p\text{-}Cut_2 \quad\rightsquigarrow$$

$$\dfrac{\Delta\Longrightarrow Q\diamond\,\text{foc} \qquad \Gamma\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w B}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\diamond\,\text{foc}}\,p\text{-}Cut_2$$

$$(28)\quad \dfrac{\dfrac{\Delta\Longrightarrow_w M\diamond\,\text{foc} \qquad \Delta\Longrightarrow_w A\diamond\,\text{foc}}{\Delta\Longrightarrow_w M\&A\diamond\,\text{foc}}\&R \qquad \dfrac{\Gamma\langle\overrightarrow{M}\rangle\Longrightarrow_w B}{\Gamma\langle\overrightarrow{\boxed{M\&A}}\rangle\Longrightarrow_w B}\&L}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\diamond\,\text{foc}}\,p\text{-}Cut_2 \quad\rightsquigarrow$$

$$\dfrac{\Delta\Longrightarrow M\diamond\,\text{foc} \qquad \Gamma\langle\overrightarrow{M}\rangle\Longrightarrow_w B}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\diamond\,\text{foc}}\,n\text{-}Cut_1$$

- Left commutative *p-Cut* conversions:

$$(29)\quad \dfrac{\dfrac{\Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w N}{\Delta\langle\overrightarrow{Q}\rangle\Longrightarrow_w N}\,foc \qquad \Gamma\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{\Gamma\langle\Delta\langle\overrightarrow{Q}\rangle\rangle\Longrightarrow_w C}\,p\text{-}Cut_2 \quad\rightsquigarrow$$

$$\dfrac{\Delta\langle\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w N \qquad \Gamma\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{\dfrac{\Gamma\langle\Delta\langle\overrightarrow{\boxed{Q}}\rangle\rangle\Longrightarrow_w C}{\Gamma\langle\Delta\langle\overrightarrow{Q}\rangle\rangle\Longrightarrow_w C}\,foc}\,p\text{-}Cut_2$$

$$(30)\quad \dfrac{\dfrac{\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w\boxed{P}}{\Delta\langle\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w\boxed{P}}\,\odot_k L \qquad \Gamma\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Gamma\langle\Delta\langle\overrightarrow{A\odot_k B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1 \quad\rightsquigarrow$$

$$\dfrac{\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w\boxed{P} \qquad \Gamma\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\dfrac{\Gamma\langle\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}{\Gamma\langle\Delta\langle\overrightarrow{A\odot_k B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,\odot_k L}\,p\text{-}Cut_1$$

$$(31)\quad \dfrac{\dfrac{\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w N\diamond \text{foc}}{\Delta\langle\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w N\diamond \text{foc}}\,\odot_k L \qquad \Gamma\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{\Gamma\langle\Delta\langle\overrightarrow{A\odot_k B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2 \quad\rightsquigarrow$$

$$\dfrac{\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w N\diamond \text{foc} \qquad \Gamma\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{\dfrac{\Gamma\langle\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}{\Gamma\langle\Delta\langle\overrightarrow{A\odot_k B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,\odot_k L}\,p\text{-}Cut_2$$

$$(32)\quad \dfrac{\dfrac{\Gamma_1\Longrightarrow_w\boxed{P_1} \qquad \Gamma_2\langle\overrightarrow{\boxed{N_1}}\rangle\Longrightarrow_w N}{\Gamma_2\langle\overrightarrow{\boxed{N_1\uparrow_k P_1}}|_k\Gamma_1\rangle\Longrightarrow_w N}\,\uparrow_k L \qquad \Theta\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{\Theta\langle\Gamma_2\langle\overrightarrow{\boxed{N_1\uparrow_k P_1}}|_k\Gamma_1\rangle\rangle\Longrightarrow_w C}\,p\text{-}Cut_2 \quad\rightsquigarrow$$

$$\dfrac{\dfrac{\Gamma_1\Longrightarrow_w\boxed{P_1} \qquad \Theta\langle\Gamma_2\langle\overrightarrow{\boxed{N_1}}\rangle\rangle\Longrightarrow_w C}{\Theta\langle\Gamma_2\langle\overrightarrow{\boxed{N_1\uparrow_k P_1}}|_k\Gamma_1\rangle\rangle\Longrightarrow_w C}\,\uparrow_k L \qquad \Gamma_1\langle\overrightarrow{\boxed{N_1}}\rangle\Longrightarrow_w N \qquad \Theta\langle\overrightarrow{\boxed{N}}\rangle\Longrightarrow_w C}{}\,p\text{-}Cut_2$$

$$(33)\quad \dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w\boxed{P} \qquad \Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w\boxed{P}}{\Gamma\langle\overrightarrow{A\oplus B}\rangle\Longrightarrow_w\boxed{P}}\,\oplus L \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1 \quad\rightsquigarrow$$

$$\dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w\boxed{P}\qquad\Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{A}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1\qquad\dfrac{\Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w\boxed{P}\qquad\Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,\oplus L$$

(34)
$$\dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w N\diamond \text{foc}\qquad\Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w N\diamond \text{foc}}{\Gamma\langle\overrightarrow{A\oplus B}\rangle\Longrightarrow_w N\diamond \text{foc}}\,\oplus L\qquad\Delta\langle\boxed{\overrightarrow{N}}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2\quad\rightsquigarrow$$

$$\dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w N\diamond \text{foc}\qquad\Delta\langle\boxed{\overrightarrow{N}}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\langle\overrightarrow{A}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2\qquad\dfrac{\Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w N\diamond \text{foc}\qquad\Delta\langle\boxed{\overrightarrow{N}}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\langle\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\,\oplus L$$

- Right commutative *p-Cut* conversions (unordered multiple distinguished occurrences are separated by semicolons):

(35)
$$\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\dfrac{\Gamma\langle\overrightarrow{P};\boxed{\overrightarrow{Q}}\rangle\Longrightarrow_w C}{\Gamma\langle\overrightarrow{P};\overrightarrow{Q}\rangle\Longrightarrow_w C}\,foc}{\Gamma\langle\Delta;\overrightarrow{Q}\rangle\Longrightarrow_w C}\,p\text{-}Cut_1\quad\rightsquigarrow\quad\dfrac{\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\Gamma\langle\overrightarrow{P};\boxed{\overrightarrow{Q}}\rangle\Longrightarrow_w C}{\Gamma\langle\Delta;\boxed{\overrightarrow{Q}}\rangle\Longrightarrow_w C}\,p\text{-}Cut_1}{\Gamma\langle\Delta;\overrightarrow{Q}\rangle\Longrightarrow_w C}\,foc$$

(36)
$$\dfrac{\Delta\Longrightarrow_w\boxed{P_1}\qquad\dfrac{\Gamma\langle\overrightarrow{P_1}\rangle\Longrightarrow_w\boxed{P_2}}{\Gamma\langle\overrightarrow{P_1}\rangle\Longrightarrow_w P_2}\,foc}{\Gamma\langle\Delta\rangle\Longrightarrow_w P_2}\,p\text{-}Cut_1\quad\rightsquigarrow\quad\dfrac{\dfrac{\Delta\Longrightarrow_w\boxed{P_1}\qquad\Gamma\langle\overrightarrow{P_1}\rangle\Longrightarrow_w\boxed{P_2}}{\Gamma\langle\Delta\rangle\Longrightarrow_w\boxed{P_2}}\,p\text{-}Cut_1}{\Gamma\langle\Delta\rangle\Longrightarrow_w P_2}\,foc$$

(37)
$$\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\dfrac{\Gamma\langle\overrightarrow{P}\rangle|_k\overrightarrow{A}\Longrightarrow_w B\diamond \text{foc}}{\Gamma\langle\overrightarrow{P}\rangle\Longrightarrow_w B\uparrow_k A\diamond \text{foc}}\,\uparrow_k R}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\uparrow_k A\diamond \text{foc}}\,p\text{-}Cut_1\quad\rightsquigarrow\quad\dfrac{\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\Gamma\langle\overrightarrow{P}\rangle|_k\overrightarrow{A}\Longrightarrow_w B\diamond \text{foc}}{\Gamma\langle\Delta\rangle|_k\overrightarrow{A}\Longrightarrow_w B\diamond \text{foc}}\,p\text{-}Cut_1}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\uparrow_k A\diamond \text{foc}}\,\uparrow_k R$$

(38)
$$\dfrac{\Delta\Longrightarrow_w N\diamond \text{foc}\qquad\Gamma\langle\boxed{\overrightarrow{N}}\rangle\Longrightarrow_w B\uparrow_k A}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\uparrow_k A\diamond \text{foc}}\,p\text{-}Cut_2\quad\rightsquigarrow\quad\dfrac{\dfrac{\Delta\Longrightarrow_w N\diamond \text{foc}\qquad\Gamma\langle\boxed{\overrightarrow{N}}\rangle|_k\overrightarrow{A}\Longrightarrow_w B}{\Gamma\langle\Delta\rangle|_k\overrightarrow{A}\Longrightarrow_w B\diamond \text{foc}}\,p\text{-}Cut_2}{\Gamma\langle\Delta\rangle\Longrightarrow_w B\uparrow_k A\diamond \text{foc}}\,\uparrow_k R$$

(39)
$$\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\dfrac{\Gamma\langle\overrightarrow{P};\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Gamma\langle\overrightarrow{P};\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,\odot_k L}{\Gamma\langle\Delta;\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1\quad\rightsquigarrow\quad\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\Gamma\langle\overrightarrow{P};\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Gamma\langle\Delta;\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_1}{\Gamma\langle\Delta;\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,\odot_k L$$

(40)
$$\dfrac{\Delta\Longrightarrow_w N\diamond \text{foc}\qquad\Gamma\langle\boxed{\overrightarrow{N}};\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w C}{\Gamma\langle\Delta;\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2\quad\rightsquigarrow\quad\dfrac{\dfrac{\Delta\Longrightarrow_w N\diamond \text{foc}\qquad\Gamma\langle\boxed{\overrightarrow{N}};\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w C}{\Gamma\langle\Delta;\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,p\text{-}Cut_2}{\Gamma\langle\Delta;\overrightarrow{A\odot_k B}\rangle\Longrightarrow_w C\diamond \text{foc}}\,\odot_k L$$

(41)
$$\dfrac{\Delta\Longrightarrow_w\boxed{P}\qquad\dfrac{\Gamma\Longrightarrow_w\boxed{P_1}\qquad\Theta\langle\overrightarrow{P_2};\overrightarrow{P}\rangle\Longrightarrow_w C}{\Theta\langle\boxed{\overrightarrow{P_2\uparrow_k P_1}}|_k\Gamma;\overrightarrow{P}\rangle\Longrightarrow_w C}\,\uparrow_k L}{\Theta\langle\boxed{\overrightarrow{P_2\uparrow_k P_1}}|_k\Gamma;\Delta\rangle\Longrightarrow_w C}\,p\text{-}Cut_1\quad\rightsquigarrow$$

$$\dfrac{\Gamma \Longrightarrow_w \boxed{P_1} \qquad \dfrac{\Delta \Longrightarrow_w \boxed{P} \qquad \Theta\langle \overrightarrow{P_2}; \overrightarrow{P} \rangle \Longrightarrow_w C}{\Theta\langle \overrightarrow{P_2}; \Delta \rangle \Longrightarrow_w C}\, p\text{-}Cut_1}{\Theta\langle \overrightarrow{\boxed{P_2 \uparrow_k P_1} |_k \Gamma; \Delta \rangle \Longrightarrow_w C}}\, \uparrow_k L$$

$$(42)\quad \dfrac{\Delta \Longrightarrow_w \boxed{P} \qquad \dfrac{\Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w A \diamond \text{foc} \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w B \diamond \text{foc}}{\Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w A \& B \diamond \text{foc}}\, \&R}{\Gamma\langle \Delta \rangle \Longrightarrow_w A \& B \diamond \text{foc}}\, p\text{-}Cut_1 \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Longrightarrow_w \boxed{P} \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w A \diamond \text{foc}}{\Gamma\langle \Delta \rangle \Longrightarrow_w A \diamond \text{foc}}\, p\text{-}Cut_1 \qquad \dfrac{\Delta \Longrightarrow_w \boxed{P} \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w B \diamond \text{foc}}{\Gamma\langle \Delta \rangle \Longrightarrow_w B \diamond \text{foc}}\, p\text{-}Cut_1}{\Gamma\langle \Delta \rangle \Longrightarrow_w A \& B \diamond \text{foc}}\, \&R$$

$$(43)\quad \dfrac{\Delta \Longrightarrow_w N \diamond \text{foc} \qquad \dfrac{\Gamma\langle \boxed{\overrightarrow{N}} \rangle \Longrightarrow_w A \qquad \Gamma\langle \boxed{\overrightarrow{N}} \rangle \Longrightarrow_w B}{\Gamma\langle \boxed{\overrightarrow{N}} \rangle \Longrightarrow_w A \& B}\, \&R}{\Gamma\langle \Delta \rangle \Longrightarrow_w A \& B \diamond \text{foc}}\, p\text{-}Cut_2 \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Longrightarrow_w N \diamond \text{foc} \qquad \Gamma\langle \boxed{\overrightarrow{N}} \rangle \Longrightarrow_w A}{\Gamma\langle \Delta \rangle \Longrightarrow_w A \diamond \text{foc}}\, p\text{-}Cut_2 \qquad \dfrac{\Delta \Longrightarrow_w N \diamond \text{foc} \qquad \Gamma\langle \boxed{\overrightarrow{N}} \rangle \Longrightarrow_w B}{\Gamma\langle \Delta \rangle \Longrightarrow_w B \diamond \text{foc}}\, p\text{-}Cut_2}{\Gamma\langle \Delta \rangle \Longrightarrow_w A \& B \diamond \text{foc}}\, \&R$$

- Left commutative *n-Cut* conversions:

$$(44)\quad \dfrac{\dfrac{\Delta\langle \overrightarrow{\boxed{Q}} \rangle \Longrightarrow_w P}{\Delta\langle \overrightarrow{Q} \rangle \Longrightarrow_w P}\, foc \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w C}{\Gamma\langle \Delta\langle \overrightarrow{Q} \rangle \rangle \Longrightarrow_w C}\, n\text{-}Cut_1 \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \overrightarrow{\boxed{Q}} \Longrightarrow_w P \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w C}{\Gamma\langle \Delta\langle \overrightarrow{\boxed{Q}} \rangle \rangle \Longrightarrow_w C}\, n\text{-}Cut_1}{\Gamma\langle \Delta\langle \overrightarrow{Q} \rangle \rangle \Longrightarrow_w C}\, foc$$

$$(45)\quad \dfrac{\dfrac{\Delta\langle \overrightarrow{A} |_k \overrightarrow{B} \rangle \Longrightarrow_w P \diamond \text{foc}}{\Delta\langle \overrightarrow{A \odot_k B} \rangle \Longrightarrow_w P \diamond \text{foc}}\, \odot_k L \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w C}{\Gamma\langle \Delta\langle \overrightarrow{A \odot_k B} \rangle \rangle \Longrightarrow_w C \diamond \text{foc}}\, n\text{-}Cut_1 \quad \rightsquigarrow$$

$$\dfrac{\Delta\langle \overrightarrow{A} |_k \overrightarrow{B} \rangle \Longrightarrow_w P \diamond \text{foc} \qquad \Gamma\langle \overrightarrow{P} \rangle \Longrightarrow_w C}{\dfrac{\Gamma\langle \Delta\langle \overrightarrow{A} |_k \overrightarrow{B} \rangle \rangle \Longrightarrow_w C \diamond \text{foc}}{\Gamma\langle \Delta\langle \overrightarrow{A \odot_k B} \rangle \rangle \Longrightarrow_w C \diamond \text{foc}}\, \odot_k L}\, n\text{-}Cut_1$$

$$(46)\quad \dfrac{\dfrac{\Delta\langle \overrightarrow{A} |_k \overrightarrow{B} \rangle \Longrightarrow_w N}{\Delta\langle \overrightarrow{A \odot_k B} \rangle \Longrightarrow_w N}\, \odot_k L \qquad \Gamma\langle \overrightarrow{N} \rangle \Longrightarrow_w C \diamond \text{foc}}{\Gamma\langle \Delta\langle \overrightarrow{A \odot_k B} \rangle \rangle \Longrightarrow_w C \diamond \text{foc}}\, n\text{-}Cut_2 \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\Longrightarrow_w N \qquad \Gamma\langle\overrightarrow{N}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Gamma\langle\Delta\langle\overrightarrow{A}|_k\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\odot_k L}{\Gamma\langle\Delta\langle\overrightarrow{A\odot_k B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_2$$

$$\text{(47)}\quad \dfrac{\dfrac{\Gamma_1\Longrightarrow_w\boxed{P_1} \qquad \Gamma_2\langle\overrightarrow{\boxed{N_1}}\rangle\Longrightarrow_w P}{\Gamma_2\langle\overrightarrow{\boxed{N_1\uparrow_k P_1}}|_k\Gamma_1\rangle\Longrightarrow_w P}\uparrow_k L \qquad \Theta\langle\overrightarrow{P}\rangle\Longrightarrow_w C}{\Theta\langle\Gamma_2\langle\overrightarrow{\boxed{N_1\uparrow_k P_1}}|_k\Gamma_1\rangle\rangle\Longrightarrow_w C}\textit{n-Cut}_1 \quad\rightsquigarrow\quad \dfrac{\Gamma_1\langle\overrightarrow{\boxed{N_1}}\rangle\Longrightarrow_w P \qquad \Theta\langle\overrightarrow{P}\rangle\Longrightarrow_w C}{\dfrac{\Gamma_1\Longrightarrow_w\boxed{P_1} \qquad \Theta\langle\Gamma_2\langle\overrightarrow{\boxed{N_1}}\rangle\rangle\Longrightarrow_w C}{\Theta\langle\Gamma_2\langle\overrightarrow{\boxed{N_1\uparrow_k P_1}}|_k\Gamma_1\rangle\rangle\Longrightarrow_w C}\uparrow_k L}\textit{n-Cut}_1$$

$$\text{(48)}\quad \dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w P\diamond \text{foc} \qquad \Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w P\diamond \text{foc}}{\Gamma\langle\overrightarrow{A\oplus B}\rangle\Longrightarrow_w P\diamond \text{foc}}\oplus L \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_1 \quad\rightsquigarrow\quad \dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w P\diamond \text{foc} \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\langle\overrightarrow{A}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_1 \qquad \dfrac{\Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w P\diamond \text{foc} \qquad \Delta\langle\overrightarrow{P}\rangle\Longrightarrow_w C}{\Delta\langle\Gamma\langle\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_1}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\oplus L$$

$$\text{(49)}\quad \dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w N \qquad \Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w N}{\Gamma\langle\overrightarrow{A\oplus B}\rangle\Longrightarrow_w N}\oplus L \qquad \Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_2 \quad\rightsquigarrow\quad \dfrac{\dfrac{\Gamma\langle\overrightarrow{A}\rangle\Longrightarrow_w N \qquad \Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{A}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_2 \qquad \dfrac{\Gamma\langle\overrightarrow{B}\rangle\Longrightarrow_w N \qquad \Delta\langle\overrightarrow{N}\rangle\Longrightarrow_w C\diamond \text{foc}}{\Delta\langle\Gamma\langle\overrightarrow{B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\textit{n-Cut}_2}{\Delta\langle\Gamma\langle\overrightarrow{A\oplus B}\rangle\rangle\Longrightarrow_w C\diamond \text{foc}}\oplus L$$

- Right commutative *n-Cut* conversions:

$$\text{(50)}\quad \dfrac{\Delta\Longrightarrow_w N \qquad \dfrac{\Gamma\langle\overrightarrow{N};\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w C}{\Gamma\langle\overrightarrow{N};\overrightarrow{Q}\rangle\Longrightarrow_w C}\textit{foc}}{\Gamma\langle\Delta;\overrightarrow{Q}\rangle\Longrightarrow_w C}\textit{n-Cut}_2 \quad\rightsquigarrow\quad \dfrac{\dfrac{\Delta\Longrightarrow_w N \qquad \Gamma\langle\overrightarrow{N};\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w C}{\Gamma\langle\Delta;\overrightarrow{\boxed{Q}}\rangle\Longrightarrow_w C}\textit{n-Cut}_2}{\Gamma\langle\Delta;\overrightarrow{Q}\rangle\Longrightarrow_w C}\textit{foc}$$

$$\text{(51)}\quad \dfrac{\Delta\Longrightarrow_w N \qquad \dfrac{\Gamma\langle\overrightarrow{N}\rangle\Longrightarrow_w\boxed{P}}{\Gamma\langle\overrightarrow{N}\rangle\Longrightarrow_w P}\textit{foc}}{\Gamma\langle\Delta\rangle\Longrightarrow_w P}\textit{n-Cut}_2 \quad\rightsquigarrow\quad \dfrac{\dfrac{\Delta\Longrightarrow_w N \qquad \Gamma\langle\overrightarrow{N}\rangle\Longrightarrow_w\boxed{P}}{\Gamma\langle\Delta\rangle\Longrightarrow_w\boxed{P}}\textit{n-Cut}_2}{\Gamma\langle\Delta\rangle\Longrightarrow_w P}\textit{foc}$$

$$(52)\ \dfrac{\Delta \Longrightarrow_w P \diamond \text{foc} \qquad \dfrac{\Gamma\langle \overrightarrow{P}\rangle|_k \overrightarrow{A} \Longrightarrow_w B}{\Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w B\uparrow_k A}\uparrow_k R}{\Gamma\langle\Delta\rangle \Longrightarrow_w B\uparrow_k A \diamond \text{foc}}\ n\text{-}Cut_1 \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Longrightarrow_w P \diamond \text{foc} \qquad \Gamma\langle \overrightarrow{P}\rangle|_k \overrightarrow{A} \Longrightarrow_w B}{\Gamma\langle\Delta\rangle|_k \overrightarrow{A} \Longrightarrow_w B \diamond \text{foc}}\ n\text{-}Cut_1}{\Gamma\langle\Delta\rangle \Longrightarrow_w B\uparrow_k A \diamond \text{foc}}\uparrow_k R$$

$$(53)\ \dfrac{\Delta \Longrightarrow_w N \qquad \dfrac{\Gamma\langle \overrightarrow{P}\rangle|_k \overrightarrow{A} \Longrightarrow_w B \diamond \text{foc}}{\Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w B\uparrow_k A \diamond \text{foc}}\uparrow_k R}{\Gamma\langle\Delta\rangle \Longrightarrow_w B\uparrow_k A \diamond \text{foc}}\ n\text{-}Cut_2 \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Longrightarrow_w N \qquad \Gamma\langle \overrightarrow{P}\rangle|_k \overrightarrow{A} \Longrightarrow_w B \diamond \text{foc}}{\Gamma\langle\Delta\rangle|_k \overrightarrow{A} \Longrightarrow_w B \diamond \text{foc}}\ n\text{-}Cut_2}{\Gamma\langle\Delta\rangle \Longrightarrow_w B\uparrow_k A \diamond \text{foc}}\uparrow_k R$$

$$(54)\ \dfrac{\Delta \Longrightarrow_w P \diamond \text{foc} \qquad \dfrac{\Gamma\langle \overrightarrow{P}; \overrightarrow{A}|_k \overrightarrow{B}\rangle \Longrightarrow_w C}{\Gamma\langle \overrightarrow{P}; \overrightarrow{A\odot_k B}\rangle \Longrightarrow_w C}\odot_k L}{\Gamma\langle\Delta; \overrightarrow{A\odot_k B}\rangle \Longrightarrow_w C \diamond \text{foc}}\ n\text{-}Cut_1 \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Longrightarrow_w P \diamond \text{foc} \qquad \Gamma\langle \overrightarrow{P}; \overrightarrow{A}|_k \overrightarrow{B}\rangle \Longrightarrow_w C}{\Gamma\langle\Delta; \overrightarrow{A}|_k \overrightarrow{B}\rangle \Longrightarrow_w C \diamond \text{foc}}\ n\text{-}Cut_1}{\Gamma\langle\Delta; \overrightarrow{A\odot_k B}\rangle \Longrightarrow_w C \diamond \text{foc}}\odot_k L$$

$$(55)\ \dfrac{\Delta \Longrightarrow_w N \qquad \dfrac{\Gamma\langle \overrightarrow{N}; \overrightarrow{A}|_k \overrightarrow{B}\rangle \Longrightarrow_w C \diamond \text{foc}}{\Gamma\langle \overrightarrow{N}; \overrightarrow{A\odot_k B}\rangle \Longrightarrow_w C \diamond \text{foc}}\odot_k L}{\Gamma\langle\Delta; \overrightarrow{A\odot_k B}\rangle \Longrightarrow_w C \diamond \text{foc}}\ n\text{-}Cut_2 \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Longrightarrow_w N \qquad \Gamma\langle \overrightarrow{N}; \overrightarrow{A}|_k \overrightarrow{B}\rangle \Longrightarrow_w C \diamond \text{foc}}{\Gamma\langle\Delta; \overrightarrow{A}|_k \overrightarrow{B}\rangle \Longrightarrow_w C \diamond \text{foc}}\ n\text{-}Cut_2}{\Gamma\langle\Delta; \overrightarrow{A\odot_k B}\rangle \Longrightarrow_w C \diamond \text{foc}}\odot_k L$$

$$(56)\ \dfrac{\Delta \Longrightarrow_w N \qquad \Theta\langle \overrightarrow{\boxed{P_2\uparrow_k P_1}}|_k \Gamma; \overrightarrow{N}\rangle \Longrightarrow_w C}{\Theta\langle \overrightarrow{\boxed{P_2\uparrow_k P_1}}|_k \Gamma; \Delta\rangle \Longrightarrow_w C}\ n\text{-}Cut_2 \quad \rightsquigarrow$$

$$\dfrac{\Gamma \Longrightarrow_w \boxed{P_1} \qquad \dfrac{\Delta \Longrightarrow_w N \qquad \Theta\langle \overrightarrow{P_2}; \overrightarrow{N}\rangle \Longrightarrow_w C}{\Theta\langle \overrightarrow{P_2}; \Delta\rangle \Longrightarrow_w C}\ n\text{-}Cut_2}{\Theta\langle \overrightarrow{\boxed{P_2\uparrow_k P_1}}|_k \Gamma; \Delta\rangle \Longrightarrow_w C}\uparrow_k L$$

$$(57)\ \dfrac{\Delta \Longrightarrow_w P \diamond \text{foc} \qquad \dfrac{\Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w A \qquad \Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w B}{\Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w A\&B}\&R}{\Gamma\langle\Delta\rangle \Longrightarrow_w A\&B \diamond \text{foc}}\ n\text{-}Cut_1 \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Longrightarrow_w P \qquad \Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w A}{\Gamma\langle\Delta\rangle \Longrightarrow_w A \diamond \text{foc}}\ n\text{-}Cut_1 \qquad \dfrac{\Delta \Longrightarrow_w P \diamond \text{foc} \qquad \Gamma\langle \overrightarrow{P}\rangle \Longrightarrow_w B}{\Gamma\langle\Delta\rangle \Longrightarrow_w B \diamond \text{foc}}\ n\text{-}Cut_1}{\Gamma\langle\Delta\rangle \Longrightarrow_w A\&B \diamond \text{foc}}\&R$$

$$(58)\ \dfrac{\Delta \Longrightarrow_w N \qquad \dfrac{\Gamma\langle \overrightarrow{N}\rangle \Longrightarrow_w A \diamond \text{foc} \qquad \Gamma\langle \overrightarrow{N}\rangle \Longrightarrow_w B \diamond \text{foc}}{\Gamma\langle \overrightarrow{N}\rangle \Longrightarrow_w A\&B \diamond \text{foc}}\&R}{\Gamma\langle\Delta\rangle \Longrightarrow_w A\&B \diamond \text{foc}}\ n\text{-}Cut_2 \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Longrightarrow_w N \qquad \Gamma\langle \overrightarrow{N}\rangle \Longrightarrow_w A \diamond \text{foc}}{\Gamma\langle\Delta\rangle \Longrightarrow_w A \diamond \text{foc}}\ n\text{-}Cut_2 \qquad \dfrac{\Delta \Longrightarrow_w N \qquad \Gamma\langle \overrightarrow{N}\rangle \Longrightarrow_w B \diamond \text{foc}}{\Gamma\langle\Delta\rangle \Longrightarrow_w B \diamond \text{foc}}\ n\text{-}Cut_2}{\Gamma\langle\Delta\rangle \Longrightarrow_w A\&B \diamond \text{foc}}\&R$$

This completes the proof. □