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 $CL_1 = \{\wedge_l, \vee_l, \rightarrow_r, \vee_r\}$ and $CL_2 = \{\wedge_r, \vee_r, \rightarrow_l\}$.

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 $\mathscr{P} = C_1, \ldots, C_n$ 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 $C_1, \ldots, C_n$ 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

$$\frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_1 \wedge A_2 \vdash \Delta}\ \wedge_l^a \quad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_l^m \quad \frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee_l^a \quad \frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'}\ \vee_l^m$$

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\ \wedge_r^a \quad \frac{\Gamma \vdash \Delta, A \quad \Gamma' \vdash \Delta', B}{\Gamma, \Gamma' \vdash \Delta, \Delta', A \wedge B}\ \wedge_r^m \quad \frac{\Gamma \vdash \Delta, A_i}{\Gamma \vdash \Delta, A_1 \vee A_2}\ \vee_r^a \quad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}\ \vee_r^m$$

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: $CL_1 = \{\wedge_r^a, \wedge_l^m, \vee_r^m, \vee_l^a, \wedge_l^a, \vee_r^a\}$ and $CL_2 = \{\wedge_r^m, \wedge_l^a, \vee_r^a, \vee_l^m\}$, 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 $\{\wedge_l^a, \vee_r^a\}$. Therefore, it is admissible in the negative phase. For the positive phase, on the other hand, contraction will not permute up, for example, $\wedge_l^a$. 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.