A Systematic Approach to Canonicity in the Classical Sequent Calculus
Kaustuv Chaudhuri, Stefan Hetzl, Dale Miller

To cite this version:

HAL Id: hal-00772396
https://inria.hal.science/hal-00772396
Submitted on 10 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Systematic Approach to Canonicity in the Classical Sequent Calculus

Kaustuv Chaudhuri$^1$    Stefan Hetzl$^2$    Dale Miller$^1$

$^1$INRIA & LIX/Ecole Polytechnique, France
{kaustuv.chaudhuri,dale.miller}@inria.fr

$^2$Vienna University of Technology, Austria
hetzl@logic.at

January 10, 2013

Abstract

The sequent calculus is often criticized for requiring proofs to contain large amounts of low-level syntactic details that can obscure the essence of a given proof. Because each inference rule introduces only a single connective, sequent proofs can separate closely related steps—such as instantiating a block of quantifiers—by irrelevant noise. Moreover, the sequential nature of sequent proofs forces proof steps that are syntactically non-interfering and permutable to nevertheless be written in some arbitrary order. The sequent calculus thus lacks a notion of canonicity: proofs that should be considered essentially the same may not have a common syntactic form. To fix this problem, many researchers have proposed replacing the sequent calculus with proof structures that are more parallel or geometric. Proof-nets, matings, and atomic flows are examples of such revolutionary formalisms. We propose, instead, an evolutionary approach to recover canonicity within the sequent calculus, which we illustrate for classical first-order logic. The essential element of our approach is the use of a multi-focused sequent calculus as the means of abstracting away the details from classical cut-free sequent proofs. We show that, among the multi-focused proofs, the maximally multi-focused proofs that make the foci as parallel as possible are canonical. Moreover, such proofs are isomorphic to expansion proofs—a well known, minimalistic, and parallel generalization of Herbrand disjunctions—for classical first-order logic. This technique is a systematic way to recover the desired essence of any sequent proof without abandoning the sequent calculus.

1 Introduction

The sequent calculus, initially described by Gentzen for classical and intuitionistic first-order logic [10], has become a standard proof formalism for a wide variety of logics. One of the chief reasons for its ubiquity is that it defines provability in a logic parsimoniously and modularly, with every logical connective defined by introduction rules, and with the logical properties defined by structural rules. Sequent rules can thus be seen as the atoms of logical inference. Different logics can be described simply by choosing different atoms. For instance, linear logic [11] differs from classical logic by removing the structural rules of weakening and contraction, and letting the multiplicative and the additive variants of introduction rules introduce different connectives.
The proof-theoretic properties of the logics can then be derived by analyzing these atoms of inference. For example, the cut-elimination theorem directly shows that the logic is consistent.

Yet, despite its success as a framework for establishing proof-theoretic properties, the sequent proofs themselves seem to obscure the “essence” of a proof. One quickly feels that sequent proofs are syntactic monsters: they record the exact sequence of inferences and detours even when it is not really relevant to the essential high level features of the proof.

The usual approach over the years to dealing with this syntactic morass of the sequent calculus—and some other proof systems with similar issues—is one of revolution. Instead of the sequent calculus, new proof formalisms are proposed that are supposedly free of syntactic bureaucracy. Usually, such formalisms are more parallel or geometric than sequent proofs. We list here several examples—not an exhaustive list—of such revolutionary proof systems.

2. Expansion trees [27] record only the instantiations of quantifiers using a tree structure.
4. Atomic flows [13] track only the flow of atoms in a proof and can expose the dynamics of cut-elimination.
5. Even Gentzen’s natural deduction calculus [10] is arguably a principally different representation of proofs.

These revolutionary approaches continue by providing a means of de-sequentializing sequent proofs into the new formalism, and then arguing that two sequent proofs are essentially the same if they de-sequentialize to the same form. While compelling, it is worth noting that such approaches are not without problems. At a basic level, showing when a proposed structure is correct—that it actually represents a “proof”—generally requires checking global criteria such as connectedness, acyclicity, or well-scoping. Such formalisms generally lack local correctness criteria, wherein a partial (unfinished) proof object can be ensured to have only correct finished forms. By contrast, every instance of a rule in a (partial) sequent proof can easily be checked to be an instance of a proper rule schema.

A second and bigger issue with such revolutionary formalisms is that none of them is as general as the sequent calculus. Proof-nets, to pick an example, are only well defined for the unit-free multiplicative linear logic (MLL) [11]. Even adding the multiplicative units is tricky [22] and for larger fragments such as MALL with units the problem of finding a proof-net formalism remains open.

In this paper, we consider instead an evolutionary approach to extracting the essence of sequent proofs without discarding the sequent calculus. We simply add abstractions to the sequent calculus as follows.

1. Analysis of the permutation properties of sequent rules shows that some rules are invertible, and hence require no choice, while others are non-invertible and the proofs must record the choices made for them. These two classes of rules can be used to organize sequent proofs in such a way that the inference atoms coalesce into larger inference molecules—several small inference steps combine into synthetic steps or actions. The essential information in a proof is then moved to the action boundaries. Focusing [1] is the general technique for this kind of synthesis for cut-free sequent calculi, and it can be described as a simple
local modification of the usual sequent rules that preserves completeness. We then simply
remove unfocused proofs.

2. The standard focusing technique can be extended to allow multi-focusing, where multiple
actions can be done in parallel, simultaneously. The exact order of the inferences constituting
two simultaneous actions can then be elided from sequent proofs. Proofs with the
same parallel action structure are identified, which we call action equivalence.

3. Finally, if we insist on as much parallelism as possible, i.e., on maximal multi-focusing, then
such proofs are action-canonical. That is, two equivalent maximal multi-focused proofs
can be shown to be action equivalent. Thus, for each multi-focused proof, its equivalent
maximal form is action canonical.

In this paper, we apply this method to classical first-order logic. We show not only that the
evolutionary approach gives us canonical sequent proofs at the level of the action abstraction but
also that these proofs induce the same notion of identity as expansion proofs [27], an existing
parallel (revolutionary) approach for classical first-order (and higher-order) logic. This result is
surprising because it is known that expansion trees can be more compact than sequent proofs by
an exponential factor [4].

In section 2, we give some background on the sequent calculus and multi-focusing. Section 3
provides the definition of expansion trees and their interconversion with sequent proofs. Section 4
presents the main technical result that maximal multi-focused proofs are isomorphic to expansion
proofs. We begin with a quick summary of related work.

1.1 Related Work

1.1.1 Denotational Semantics of Classical Proofs

It is well known that cut-elimination using Gentzen’s cut-reduction rules is non-confluent for
$LK$ proofs [12, 3, 17]. It is generally believed that classical logic lacks a denotational semantics
for proofs akin to Cartesian-closed categories (CCC) for intuitionistic logic or $*$-autonomous
categories for linear logic. For example, if one tries to enrich the usual CCC semantics for
intuitionistic logic with an involutive negation, then the CCC degenerates into a poset that
equates all proofs of a formula (Joyal’s paradox) [23].

This problem has been attacked from both the syntactic and the semantic ends. Of the
syntactic approaches, one can recover confluence (up to a small equivalence relation) as well as
strong normalization by fixing particular cut-reduction strategies [8]. If one refrains from fixing
a reduction strategy one may still obtain a strongly normalizing though non-confluent system
by using sufficiently strong local reductions [31, 32]. Another approach is to carry out cut-
elimination in a more abstract formalism, similar to a proof-net, on the level of quantifiers (see [14]
and [25]). The reduction in such a setting is typically not confluent and strong normalization
is open [25] or known not to hold [14]. Confluence (up to the equivalence relation of having
the same expansion tree) as well as normalization can be recovered for a class of proofs [19]
by considering a maximal abstract reduction based on tree grammars [18] which contains all
concrete reductions. Extension of these results to all proofs is open.

From the semantic end, briefly, there are two principal approaches. The first approach rejects
the involutive negation, which results in negation having a computational content that can be
reified in the $\lambda\mu$ calculus with a semantics in terms of control categories (see [15] for a survey).
The second approach rejects the Cartesian structure for conjunctions, which requires a variant
of proof-nets called flow graphs for the proofs and a semantics in terms of enriched Boolean
categories [21, 30].
1.1.2 Cut-Free Formalisms

This paper deals with the question of recovering the essence of cut-free sequent proofs. There are a number of alternative approaches to this question. For example, the notion of proof-nets while well-behaved on MLL does not scale nicely to larger logics. Girard sketched a design of proof-nets for classical logic [12] that was subsequently fully formalized by Robinson [29], but these nets differentiate between some sequent proofs that are related by rule permutations because of the non-canonicity of weakening nodes. Similar problems also exist for the B/N-net formalisms [22] based on flow graphs, or the combinatorial proofs of Hughes [20]. It is possible to recover the canonicity lost with Robinson’s proof-nets by removing weakening (with the use of MIX) and rigidly controlling contraction [26]. This results in expansion nets, which are related to expansion trees [27], but are limited to the propositional fragment.

Expansion trees, because they generalize Herbrand disjunctions, are applicable to first-order and even higher-order logics. They achieve this generality by recording only the quantifier instances in a tree structure, and therefore have an expensive correctness criterion involving checking that the deep formula for an expansion tree is a tautology. The mating method [2] or the connection method [5] represents these tautological checks using graph structures, but the correctness criteria for such structures are no less expensive to check than deciding whether the deep formula is a tautology.

To our knowledge, there has been only a single attempt to produce canonical proof structures directly in the sequent calculus, in this case for ⊤-free propositional MALL [7]. This attempt also used multi-focusing as its abstraction mechanism, and it is actually the first place where the concept of maximally multi-focused proofs appears in the literature. It is important to note that the notion of a maximal multi-focused proof strictly generalizes existing canonical forms in other contexts. For example, for intuitionistic logic, if one uses the focused sequent calculus LJF [24] with just the two negative connectives of implication and universal quantification and with negative atomic formulas, then maximal multi-focused proofs are the same as singly focused proofs. Moreover, they correspond to the familiar β-normal η-long forms of the typed λ-calculus [9].

2 Background: Sequent Calculus, Focusing, and Canonicity

We use the usual syntax for (first-order) formulas (A, B, . . . ) and connectives drawn from {∧, ∨, ⊤, ⊥, ¬, ∀, ∃}. Atomic formulas (a, b, . . . ) are of the form p(t1, . . . , tn) where p represents a predicate symbol and t1, . . . , tn are first-order terms (n ≥ 0). Formulas are assumed to be identical up to α-equivalence and in negation-normal form (i.e., only atomic formulas can be ¬-prefixed). We use literal to refer to either an atomic formula or a negated atomic formula. We write (A)“ for the De Morgan dual of A, and [t/x]A for the capture-avoiding substitution of term t for x in A. We also write ∃x · A for ∃x1, . . . , ∃xn · A, ∀x · A for ∀x1, . . . , ∀xn · A, and [t/x] for [t1/x1] · · · [tn/xn] if x = x1, . . . , xn and t = t1, . . . , tn.

2.1 Sequent Calculus

We use one-sided sequents ⊢ Γ in which Γ is a multiset of formulas. Figure 1 contains the inference rules for our sequent calculus that we call LKN. There is no cut rule, the initial rule is restricted to atomic formulas, and all the rules except for ∃ are invertible. Since invertible rules are associated with the negative polarity in focused proof systems, we use the N in LKN to highlight the fact that is a variant of Gentzen’s LK calculus in which most rules are invertible.
The following rules are admissible in $LKN$; in these rules, $A$ can be any formula.

$$
\frac{\vdash \Gamma, A}{\vdash \Gamma} \quad \frac{\vdash \Gamma, (A)^\perp}{\vdash \Gamma} \quad \frac{\vdash \Gamma, (A)^\perp, A}{\vdash \Gamma, A} \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x. A} \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, \exists x. A} \quad \frac{\vdash \Gamma, \Delta}{\vdash \Gamma} \quad \frac{\vdash \Gamma}{\vdash \Gamma} \quad \frac{\vdash \Gamma}{\vdash \Gamma} \quad \frac{\vdash \Gamma}{\vdash \Gamma} \quad \frac{\vdash \Gamma}{\vdash \Gamma} \quad \frac{\vdash \Gamma}{\vdash \Gamma}
$$

These admissible rules easily allow us to mimic any of the other standard inference rules for this logic in $LKN$, including Gentzen’s original $LK$ calculus, so completeness is immediate. Soundness is equally trivial as every rule preserves classical validity under the interpretation of a sequent $\vdash A_1, \ldots, A_n$ as the formula $A_1 \lor \cdots \lor A_n$.

The reflexive-symmetric-transitive-congruence closure of the permutation steps defines the equivalence relation $\sim$ over $LKN$ proofs. One of the standard goals of proof theory is to find canonical syntactic representatives of the permutative equivalence classes for a given sequent calculus. We shall employ focusing to produce such representatives of $LKN$ proofs, following a technique introduced in [7] for $\top$-free multiplicative-additive linear logic ($\text{MALL}$) using the technical device of multi-focusing.

There is one critical difference between the approach of [7] and that of this paper: we restrict permutation steps to cases where both of the rules being permuted have at least one premise. In other words, $\top/r$ and init/r permutation steps are impossible for any rule $r$; in particular, we disallow the following permutation step.

$$
\frac{\vdash \Gamma, A, \Delta, \top}{\vdash \Gamma, \top, \Delta} \quad \quad \rightarrow \quad \quad \frac{\vdash \Gamma, \top, \Delta}{\vdash \Gamma, \top}
$$

If such permutation steps were to be allowed, then the induced equivalence on $LKN$ proofs would equate arbitrary sub-proofs and defeat any attempt at canonicity. Observe that preventing such permutations does not affect the classical symmetries, i.e., $A$ continues to be identical to $(A^\perp)^\perp$.

### 2.2 Focused Sequent Calculus

The proof-theoretic analysis of the logic programming paradigm developed in the 1980s accounted for notions of goal-reduction and back-chaining as two alternating phases in the construction of (cut-free) sequent proofs [28]. Andreoli [1] developed the notion of focused sequent proofs for classical linear logic as a generalization of this earlier work in logic programming. Subsequently, focused sequent calculus proofs have been written for intuitionistic and classical logics [24]. Such
proof systems are increasingly being seen as general proof-theoretic tools for uncovering structures within proofs.

A focused calculus partitions formulas into positive and negative polarities based on the permutation properties of their sequent rules. Similarly, the introduction rules in a focused calculus appear in either one of two phases. The asynchronous or negative phase consists of applying all available invertible rules to the negative non-atomic formulas, in an arbitrary order, until none remains. The synchronous or positive phase is then launched per sequent by focusing on one or more positive formulas using a rule called decide. In this phase, non-invertible rules are applied to the focused formulas and, crucially, the focus is maintained on the positive subformulas in the premises of the applied rule. The positive phase persists until the focused formulas all become negative; the proof then switches back to the negative phase by a rule named release.

Formally, we will use a sequent calculus that closely resembles the LKF system as given in [24], with some important differences. First, LKF allows only a single focus formula while our calculus will allow multiple foci. (It is a simple matter to add multi-focusing to LKF.) A second and bigger difference is that the LKF proof system contains a positive and negative version of both conjunction and disjunction, while we will use only the negative versions of these connectives. This choice is motivated by our desire to model the Herbrand disjunctions underlying expansion proofs, where the propositional content is elided. The last difference is that the positive phase in LKF can contain instances of the initial and \( \exists \)-introduction rules, but for our goal of obtaining a variant of Herbrand’s theorem we will need a clean separation of quantification rules and propositional rules. The critical issue is that in LKF there is only a single proof of \( \vdash \neg p(a), \exists x. p(x) \vdash \cdot \), while there are infinitely many expansion proofs of \( \neg p(a) \lor \exists x. p(x) \) that simply differ in their numbers of instances of the existential quantifier. One way to limit the focusing strength of LKF to obtain these other proofs is to replace all the occurrences of positive literals \( L \) with a delayed literal (\( L \land \top \)), which is equivalent but of negative polarity.

In Figure 2 we present our focused sequent calculus LKNF. It can be seen as the multi-focused

---

**Figure 2: Rules of LKNF.**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
</table>
| Invertible | \[ \vdash \Gamma, L \uparrow \Delta, L \vdash \Gamma \uparrow \Delta, L \]
| Invertible | \[ \vdash \Gamma \uparrow \Delta, A \vdash \Gamma \uparrow \Delta, B \] & \[ \vdash \Gamma \uparrow \Delta, \top \]
| Invertible | \[ \vdash \Gamma \uparrow \Delta, A, B \vdash \Gamma \uparrow \Delta, A \lor B \]
| Existential | \[ \vdash \Gamma \downarrow \Delta, [t/x] A \]
| Structural | \[ \vdash \Gamma \downarrow \Delta, \exists x. A \]

Notes:
1. In the store rule, \( L \) is a literal or an existential formula.
2. In the \( \forall \) rule, the principal formula is implicitly \( \alpha \)-converted so \( x \) is not free in the conclusion.
3. In the decide rule, \( \Delta \) contains only existential formulas and \( \emptyset \neq \Delta \subseteq \text{set } \Gamma \).
4. In the release rule, \( \Delta \) contains no existential formulas.
variant of LKF with only negative propositional connectives and implicitly delayed positive literals. Since the positive phase of LKNF only involves the existential quantifier, we rename the “positive phase” of LKF as the “existential phase”. The two phases of LKNF proofs are depicted using two different sequent forms: negative sequents of the form ⊢ Γ ⇑ ∆ and positive sequents of the form ⊢ Γ ⇓ ∆. In either form, Γ is a multiset of literals or existential formulas, and ∆ is a multiset of arbitrary formulas. In the positive sequent ⊢ Γ ⇓ ∆, we say that the formulas in ∆ are its foci and we require ∆ to be non-empty. We write ⊢ Γ ⇕ ∆ to stand for either sequent form.

The inference rules of LKNF are divided into three classes. The invertible rules all apply to negative sequents and contain no essential non-determinism. The existential rule is non-invertible: the witness terms must be recorded in the proof. The final class of structural rules includes: the init rule for initial sequents; the decide rule where a number of existential formulas are copied, possibly more than once, to the foci of a new positive phase; and the release rule to leave the positive phase when none of the foci is an existential formula.

LKNF is sound and complete with respect to LKN; to make this statement precise, we inject LKNF proofs to LKN proofs.

**Definition 1.** For any LKNF proof π, we write [π] to stand for that LKN proof that:

- replaces all sequents of the form ⊢ Γ ⇑ ∆ with ⊢ Γ, ∆;
- removes all instances of the rules store and release; and
- renames decide to contr in π.

**Theorem 1 (LKNF vs. LKN).**

1. If π is an LKNF proof of ⊢ Γ ⇑ ∆, then [π] is an LKN proof of ⊢ Γ, ∆ (soundness).

2. If ⊢ ∆ is provable in LKN, then ⊢ · ⇑ ∆ is provable in LKNF (completeness).

**Proof.** Soundness is immediate by inspection. Completeness follows by observing that the LKF calculus of [24], which is complete for LK (and hence also for LKN), is simply a singly focused fragment of LKNF if all its connectives are negatively biased and delays are inserted as needed around literals.

We can also define an equivalence over LKNF proofs in terms of rule permutations. The permutations in the focused setting are subtle; certain permutations such as decide/store are simply impossible. We therefore exploit the injection of definition 1 to bootstrap the LKNF equivalence using the LKN equivalence.

**Definition 2.** Two LKNF proofs π₁ and π₂ of the same sequent are equivalent, written π₁ ∼ π₂, iff [π₁] ∼ [π₂].

### 2.3 Canonicity

The main benefit of focusing is that the introduction rules of the unfocused calculus (LKN) coalesce into larger synthetic rules that represent actions. Every action begins at the bottom with an instance of decide, and the action ends with premises of the form ⊢ Γ ⇑ ·. The underlying LKN rules inside a single action can be freely permuted with each other, and it is not important to record their particular sequence. In other words, two equivalent LKNF proofs should be considered “the same” if they use the decide rules in the same way; we call such proofs action equivalent.
Definition 3. Two LKNF proofs \( \pi_1 \) and \( \pi_2 \) of the same sequent are action equivalent, written \( \pi_1 \cong \pi_2 \), iff they are equivalent (definition 2) and are tree-isomorphic for the instances of the decide rules.

Action equivalence gives us the “essence” of cut-free focused sequent proofs. Since two action equivalent proofs have the same decide rules, one can reason about such proofs by induction on the decision depth—i.e., the depth of the decide rules—in the LKNF proof. If from a proof we simply elide all but the decide rules, and record the existential witnesses along with these instances of decide, we can then obtain a canonical synthetic representation of the proof directly in the sequent calculus. (It is indeed possible to build a sequent calculus that uses solely synthetic sequent rules [6].)

Two equivalent LKNF derivations need not be action equivalent as they may perform the decide steps in a different order or with different foci. However, each equivalence class of LKNF proofs does have a canonical form where the foci of each decide rule are selected to be as numerous as possible.

Definition 4 (Maximality). Given an LKNF proof \( \pi \) that ends in an instance of decide, we write foci \((\pi)\) for the foci in the premise of that instance of decide. We say that the instance is maximal iff for every \( \pi' \sim \pi \), it is the case that \( foci(\pi') \subseteq_{\text{multiset}} foci(\pi) \). An LKNF proof is maximal iff every instance of decide in it is maximal.

The two main properties of maximal proofs are that equivalent maximal proofs are action equivalent, and that for every proof there is an equivalent maximal proof. This pair of results guarantees that the maximal proofs are canonical (action equivalent) representatives of their \( \sim \)-equivalence classes. Similar theorems have appeared in [7, 6].

Theorem 2 (Canonicity).

1. Every LKNF proof has an equivalent maximal proof.
2. Two equivalent maximal LKNF proofs are action equivalent.

Proof. Because init/contr and \( \top/\text{contr} \) permutations are disallowed, equivalent proofs have the same multiset union of all the foci of their decide rules. Using the consolidated form of contr/contr permutations, the foci of the instances of decide can be divided or combined as needed. Therefore, there is a merge operation that, starting from the bottom of an LKNF proof and going upwards, permutes and merges foci into the lowermost decide instances by splitting them from higher instances. This merge operation obviously terminates (by induction on the decision depth); moreover, the result is maximal by definition 4.

To see that two given equivalent maximal proofs are action equivalent, suppose the contrary. Then there is a lowermost instance of decide in the two proofs that have an incomparable multiset of foci (if they were comparable, then either one of the proofs is not maximal or they are action equivalent). Since the proofs are equivalent, these two decide rules themselves permute; hence, their foci can be merged as above, contradicting our assumption that they are maximal.

Definition 5. Theorem 2 shows that for every LKNF proof \( \pi \) there is a unique action equivalence class corresponding to the maximal proofs of \( \pi \). We write \( \text{max}(\pi) \) for this class.

In other words, \( \text{max}(\pi) \) is the maximally parallel structure of decide and existential inferences corresponding to \( \pi \). A simple corollary of the completeness of LKNF and canonicity is Herbrand’s theorem for prenex formulas.
Corollary 3 (Herbrand’s theorem). The formula $\exists \vec{x}. A$, where $A$ is quantifier-free, is valid if and only if there is a sequence of vectors of terms $\vec{t}_1, \ldots, \vec{t}_n$ such that the disjunction $[\vec{t}_1/\vec{x}]A \lor \cdots \lor [\vec{t}_n/\vec{x}]A$ is valid.

Proof. One direction is trivial. Suppose $\exists \vec{x}. A$ is valid, i.e., the LKN sequent $\vdash \exists \vec{x}. A$ is provable. By theorem 1 $\vdash \vdash \exists \vec{x}. A$ is provable in LKNF, i.e., $\vdash \exists \vec{x}. A \vdash$ is provable as only store applies to the former. Because $A$ is quantifier-free, the decide rule can only apply to $\exists \vec{x}. A$; thus, the equivalent maximal proof (which exists by Theorem 2) performs only (at most) a single decide at the bottom, producing a number of focused copies of $\exists \vec{x}. A$. In the positive phase, the $\exists$s are removed from the foci to give the required term vectors. 

3 Expansion Trees

Herbrand’s theorem [16] tells us that recording how quantifiers are instantiated is sufficient to describe a proof of a prenex normal formula. Gentzen [10] noticed this also in (cut-free) proofs of a prenex normal sequents via the mid-sequent. Miller [27] defined expansion trees for full higher-order logic as a structure to record such substitution information without restriction to prenex normal form. We will use a first-order version of this notion here.

Definition 6. Expansion trees and a function $\text{Sh}(\cdot)$ (for shallow) that maps an expansion tree to a formula are defined as follows:

1. A literal $L$ is an expansion tree with $\text{Sh}(L) = L$ and top node $L$.

2. If $E_1$ and $E_2$ are expansion trees and $\circ \in \{\land, \lor\}$, then $E_1 \circ E_2$ is an expansion tree with top node $\circ$ and $\text{Sh}(E_1 \circ E_2) = \text{Sh}(E_1) \circ \text{Sh}(E_2)$.

3. If $E$ is an expansion tree with $\text{Sh}(E) = [y/x]A$ and $y$ is not an eigenvariable of any node in $E$, then $\forall x. A +^y E$ is an expansion tree with top node $\forall x. A$ and $\text{Sh}(\forall x. A +^y E) = \forall x. A$. The variable $y$ is called an eigenvariable of its top node.

4. If $\{t_1, \ldots, t_n\}$ is a set of terms and $E_1, \ldots, E_n$ are expansion trees with $\text{Sh}(E_i) = [t_i/x]A$ for $i = 1, \ldots, n$, then $E = \exists x. A + t_1 E_1 + \cdots + t_n E_n$ is an expansion tree with top node $\exists x. A$ and $\text{Sh}(E') = \exists x. A$. The terms $t_1, \ldots, t_n$ are known as the expansion terms of its top node. We allow the case where $n = 0$.

Note that the requirement of $y$ not being an eigenvariable of any node in $E$ in the clause for the universal node ensures that each eigenvariable appears only once in an expansion tree. In the context of proofs this condition is often formulated globally and called regularity. The reason for requiring this property of expansion trees is that the correctness criterion is global and hence needs globally unique variable names. In contrast, the correctness of a sequent proof is locally checkable, so the (local) eigenvariable condition is enough. We shall consider eigenvariables within expansion trees to be bound over the entire expansion tree and that systematic changes to eigenvariable names ($\alpha$-conversion) result in equal trees.

There is a simple way to coerce a formula into an expansion tree: use the bound variable of a universally quantified subformula as that quantifiers eigenvariable and use the empty set of terms to expand an existentially quantified formula. Whenever we use a formula to denote an expansion tree, we shall assume that we use this coercion.

Example 1. The expression

$$\exists x. (\neg d(x) \lor \forall y. d(y)) +^v (\neg d(c) \lor (\forall y. d(y) +^u d(u))) +^w (\neg d(u) \lor (\forall y. d(y) +^v d(v)))$$
is an expansion tree that can alternatively be written as follows.

\[ \exists x. (\neg d(x) \lor \forall y. d(y)) \]

\[ \neg d(c) \lor \forall y. d(y) \]

\[ \neg d(u) \lor \forall y. d(y) \]

\[ \neg d(c) \]

\[ \forall y. d(y) \]

\[ \neg d(u) \]

\[ \forall y. d(y) \]

\[ d(u) \]

\[ \neg d(u) \]

\[ d(v) \]

\[ c \]

\[ u \]

\[ v \]

So far, we have only described a basic data structure for storing quantifier instances; we still lack a correctness criterion for deciding when such a tree is a proof. For this criterion we need the following function \( \text{Dp}(\cdot) \) (for deep).

**Definition 7.** For an expansion tree \( E \), the quantifier-free formula \( \text{Dp}(E) \), called the deep formula of \( E \), is defined as:

- \( \text{Dp}(E) = E \) for a literal \( E \),
- \( \text{Dp}(E_1 \circ E_2) = \text{Dp}(E_1) \circ \text{Dp}(E_2) \), for \( \circ \in \{\land, \lor\} \),
- \( \text{Dp}(\forall x. A +^y E) = \text{Dp}(E) \), and
- \( \text{Dp}(\exists x. A +^{t_1} E_1 \ldots +^{t_n} E_n) = \bigvee_{i=1}^n \text{Dp}(E_i) \). If \( n = 0 \) then \( \text{Dp}(\exists x. A) = \bot \).

In addition to considering expansion trees (of formulas) we will also consider expansion sequents (of sequents). If \( S = \vdash A_1, \ldots, A_n \) is a sequent and \( E_1, \ldots, E_n \) are expansion trees with \( \text{Sh}(E_i) = A_i \), then \( \vdash E_1, \ldots, E_n \) is called an expansion sequent of \( S \) if whenever \( E_i \) and \( E_j \) share an eigenvariable then \( i = j \). For an expansion sequent \( \mathcal{E} = \vdash E_1, \ldots, E_n \), define \( \text{Dp}(\mathcal{E}) = \vdash \text{Dp}(E_1), \ldots, \text{Dp}(E_n) \) and \( \text{Sh}(\mathcal{E}) = \vdash \text{Sh}(E_1), \ldots, \text{Sh}(E_n) \). A second component of the correctness criterion involves the following dependency relation.

**Definition 8.** Let \( \mathcal{E} \) be an expansion tree or expansion sequent and let \( <^\mathcal{E} \) be the binary relation on the occurrences of the expansion terms in \( \mathcal{E} \) defined by \( t <^\mathcal{E} s \) if there is an \( x \) which is free in \( s \) and which is the eigenvariable of a node dominated by \( t \). Then \( <^\mathcal{E} \), the transitive closure of \( <^\mathcal{E} \), is called the dependency relation of \( \mathcal{E} \).

In terms of the sequent calculus, \( t <^\mathcal{E} s \) means that the inference corresponding to \( t \) must be below the inference corresponding to \( s \).

**Definition 9.** Let \( A \) be a formula (\( S \) be a sequent). An expansion tree \( \mathcal{E} \) of \( A \) (or respectively an expansion sequent \( \mathcal{E} \) of \( S \)) is called an expansion proof of \( A \) (respectively \( S \)) if \( <^\mathcal{E} \) is acyclic and \( \text{Dp}(\mathcal{E}) \) is a tautology.

**Example 2.** Let \( E \) be the expansion tree of example 1. It has two expansion terms: \( c \) and \( u \). We have \( c <_E u \) because the node labeled with \( c \) dominates the \( \forall \)-node with eigenvariable \( u \). However \( u <_E c \), so \( <_E \) is cyclic; furthermore, \( \text{Dp}(E) = \neg d(c) \lor d(u) \lor \neg d(u) \lor d(v) \), which is a tautology. So, \( E \) is an expansion proof of the formula \( \text{Sh}(E) = \exists x. (\neg d(x) \lor \forall y. d(y)) \).
3.1 Expansions from Proofs

We now turn to describing how to read off an expansion proof from a sequent calculus proof. To that aim, the following merge-operation on expansion trees will be useful.

**Definition 10.** Let $E_1$ and $E_2$ be expansion trees with $\text{Sh}(E_1) = \text{Sh}(E_2)$. Then their merge $E_1 \cup E_2$ is defined as follows:

1. If $A$ is a literal then $E_1 \cup E_2 = E_1 = E_2 = A$.
2. If $E_1 = E_1' \circ E_1''$ and $E_2 = E_2' \circ E_2''$ for $\circ \in \{\land, \lor\}$, then $E_1 \cup E_2 = (E_1' \cup E_2') \circ (E_1'' \cup E_2'')$.
3. If $E_1 = \forall x. B + v_1 E_1'$ and $E_2 = \forall x. B + v_2 E_2'$, then $E_1 \cup E_2 = \forall x. B + v_1 (E_1' \cup [y_1/y_2]E_2')$.

Alphabetic change of eigenvariable names in $E_1'$ and $E_2'$ might be necessary to do this merge in general.

4. If $E_1 = \exists x. B + r_1 E_{1,1} + \ldots + r_s E_{1,k} + s_1 F_1 + \ldots + s_t F_t$ and $E_2 = \exists x. B + r_1 E_{2,1} + \ldots + r_s E_{2,k} + t_1 G_1 + \ldots + t_m G_m$ where $\{s_1, \ldots, s_t\} \cap \{t_1, \ldots, t_m\} = \emptyset$, then $E_1 \cup E_2 = \exists x. B + r_1 (E_{1,1} \cup E_{2,1}) + \ldots + r_s (E_{1,k} \cup E_{2,k}) + s_1 F_1 + \ldots + s_t F_t + t_1 G_1 + \ldots + t_m G_m$

The merge of expansion sequents is component-wise.

We now present an explicit mapping from LKN proofs to expansion proofs.

**Definition 11.** Let $\pi$ be an LKN-proof. The expansion sequent $E(\pi)$ is defined by induction: if $\pi$ is the initial rule with conclusion $\vdash \Gamma, \lnot a, a$, then let $E(\pi) = \Gamma, \lnot a, a$. (It is straightforward to coerce the formulas in $\Gamma$ into expansion trees.) The analogous translation is needed for the introduction rule for $\top$. The remaining cases for $\pi$ are the following.

\[
\begin{array}{c}
\vdash \Gamma, A \\
(\pi_1) \\
(\pi_2) \\
(\pi') \\
(\pi') \\
(\pi') \\
(\pi') \\
(a) \quad (a) \\
(b) \quad (b) \\
(c) \quad (c) \\
(d) \quad (d) \\
\vdash \Gamma, A \land B \quad \vdash \Gamma, A \quad \vdash \Gamma, A \quad \vdash \Gamma, A \quad \vdash \Gamma, A \quad \vdash \Gamma, A \land B \\
\vdash \Gamma, \lnot a, a, A \quad \vdash \Gamma, \lnot a, a, A \quad \vdash \Gamma, \forall x. A \quad \vdash \Gamma, \forall x. A \quad \vdash \Gamma, \exists x. A \\
\text{contr} \quad \text{contr} \\
\end{array}
\]

For case (a), if $E(\pi_1) = E_1, E_1$ and $E(\pi_2) = E_2, E_2$, then $E(\pi) = E_1 \cup E_2, E_1 \land E_2$. Analogous definitions apply for the other propositional rules. For case (b), if $E(\pi') = E, E$, then $E(\pi) = E, \forall x. A + y [y/x]E$ where $y$ is not an eigenvariable of a node in $E, E$. For case (c), if $E(\pi') = E, E$, then $E(\pi) = E, \exists x. A + E$. Finally, for case (d), let $\Gamma = A_1, \ldots, A_n$ with corresponding expansion trees $E_1, \ldots, E_n$ in $E(\pi')$. For $i \in \{1, \ldots, n\}$ let $k_i$ be the number of copies of $A_i$ in $\Delta$ and let $E_{i,1}, \ldots, E_{i,k_i}$ be the expansion trees corresponding to them. Then $E(\pi) = E_1 \cup E_{1,1}, \ldots, E_n \cup E_{n,k_n}$.

The above definition extends to the focused setting in a straightforward way by defining $E(\pi) = L(\pi)$ for an LKN-proof $\pi$.

**Theorem 4.** If $\pi$ is an LKN- or LKNF-proof, then $E(\pi)$ is an expansion proof.

**Proof.** That $\text{Dp}(E(\pi))$ is a tautology can be shown by induction on the depth of $\pi$ treating each of the cases of definition 11. Acyclicity of $\lambda(\pi)$ follows from the side condition of the $\forall$-rule and the appropriate choice of variable names in definition 11. □
3.2 Sequentialization

For translating expansion trees to LKNF-proofs we will proceed in two phases: first we translate an expansion tree to a proof in an intermediate calculus LKNFE which has the structure of LKNF but instead of working on sequents it works on expansion sequents. Secondly we map an LKNF-proof $\pi$ to an LKNF-proof $Sh(\pi)$ which is defined by applying $Sh(\cdot)$ to every expansion tree appearing in the proof. This operation will indeed yield a valid LKNF-proof as the Sh-image of a LKNF-rule will be a LKNF-rule. In particular, the decide-rule of LKNFE is the following, where $\Delta$ is a choice of some instances which are present in $\Gamma$ and $\Gamma'$ the remaining instances.

$$
\begin{align*}
\vdash \Gamma' \Downarrow \Delta & \quad \text{decide} \\
\vdash \Gamma & 
\end{align*}
$$

Formally: $\Gamma = E_1, \ldots, E_n$ where $E_i = \exists x. A_i +^{t_{i,1}} E_{i,1,1} \cdots +^{t_{i,n_i}} E_{i,n_i}$ and $\Gamma' = E'_1, \ldots, E'_n$ where $E'_i = \exists x. A_i +^{t_{i,1}} E_{i,1,1} \cdots +^{t_{i,k_i}} E_{i,k_i}$ with $0 \leq k_i \leq n_i$ and $\Delta = \Delta_1, \ldots, \Delta_n$ where $\Delta_i = \{ \exists x. A_i +^{t_{i,j}} E_{i,j} \mid k_i < j \leq n_i \}$. The rule for existentials in LKNFE is:

$$
\begin{align*}
\vdash \Gamma \Downarrow \Delta, E & \\
\vdash \Gamma \Downarrow \Delta \mathord{\mathord{\mathord{\vdash \ldots \vdash}}}, \exists x. A +^{t_{,s_2}} E & 
\end{align*}
$$

The other rules are adapted in the natural way.

When writing down expansion trees for formulas which contain blocks of quantifiers we will abbreviate using a vector notation. For example, the expansion term $\exists x. \exists y. A +^{(t,x)} E_1 +^{(t,y)} E_2$ is abbreviated as $\exists x, y. A +^{(t,s)} E_1 +^{(t,s_2)} E_2$. If the length of a vector is irrelevant, we write $\vec{x}$ for a vector of variables and $\vec{t}$ for a vector of terms.

We distinguish proofs and derivations in a calculus. While the initial sequents of a proof are among those declared in the definition of the calculus, the initial sequents of a derivation are arbitrary. The construction of an LKNF-proof from an expansion proof will be done in a phase-wise manner, the derivation containing the negative phase is defined as follows.

**Definition 12 ($\pi^-$).** Let $\vdash \Gamma \Uparrow \Delta$ be a focused expansion sequent where $\Delta$ consists of non-existential expansion trees only. Define the LKNF-derivation $\pi^\Uparrow_{\Gamma, \Delta} \vdash \Gamma \Uparrow \Delta$ by exhaustive application of negative rules and stores. These lead to expansion sequents $\vdash \Gamma, \Delta \Uparrow \ldots, \vdash \Gamma, \Delta_n \Uparrow$ and to finishing the proof in case $n = 0$.

We now define a derivation corresponding to the positive phase in a way that will have the effect that sequentializations of expansion trees are always maximal. This property will be crucial for the main theorem of this paper.

**Definition 13 ($\pi^+$).** Let $\vdash \Sigma \Uparrow \cdot$ be a focused expansion sequent and define the LKNF-derivation $\pi^\Uparrow_{\Sigma, \Delta} \vdash \Sigma \Uparrow \cdot$ as follows. Let $\Sigma = \Gamma, \Delta$ where $\Gamma$ are the non-existential expansion trees and $\Delta = \{E_1, \ldots, E_n\}$ are the existentials expansion trees of $\Sigma$. Then $E_i = \exists x_i. A_i +^{t_{i,1}} E_{i,1,1} \cdots +^{t_{i,n_i}} E_{i,n_i}$, $E_{i,ni}$ where $A_i$ is a negative formula. For $i \in \{1, \ldots, n\}$ let w.l.o.g. $\{1, \ldots, k_i\} = \{j \mid 1 \leq j \leq n_i, all \ terms \ in \ E_{i,j}\}$ are $<_{\Sigma}$-minimal. Define $\Delta_i'$ as $\{\exists x_i. A_i +^{t_{i,1}} E_{i,1,1}, \ldots, \exists x_i. A_i +^{t_{i,k_i}} E_{i,k_i}\}$ and $\Delta''_i$ as $\{E''_{i,1}, \ldots, E''_{i,n_i}\}$ where $E''_{i,j} = \exists x_i. A_i +^{t_{i,j,1}} E_{i,j,1} \cdots +^{t_{i,j,n_i}} E_{i,j,n_i}$ and apply the decide rule as

$$
\begin{align*}
\vdash \Gamma, \Delta'' \Downarrow \Delta'_1, \ldots, \Delta''_n & \quad \text{decide} \\
\vdash \Sigma & 
\end{align*}
$$

Because all the expansion terms in $\Delta'_i$ are $<_{\Sigma}$-minimal, exhaustive application of existential inferences is possible and, followed by a release, leads to a sequent $\vdash \Gamma, \Delta'' \Uparrow \Theta$ where $\Theta$ consists of non-existential expansion trees only.
Theorem 5 (Sequentialization). If $\mathcal{E}$ is an expansion proof, then $\vdash \uparrow \text{Sh}(\mathcal{E})$ in LKNF.

Proof. First, let the LKNFE-proof $\pi_\mathcal{E}$ of $\vdash \uparrow \mathcal{E}$ be

\[
\begin{align*}
(\psi) & \vdash \Gamma \Downarrow \Delta \\
& \vdots \\
& \vdash \cdot \uparrow \mathcal{E}
\end{align*}
\]

where $\Delta$ consists of non-existent expansion trees only and $\psi$ is obtained by alternating instances of $\pi^-$ and $\pi^+$ for appropriate expansion sequents. This construction can be carried out as $\text{Dp}(\mathcal{F})$ is a tautology for every expansion sequent $\mathcal{F}$ in $\psi$ and it terminates as the number of nodes of the current expansion sequent strictly decreases with each line of the proof. Then $\text{Sh}(\pi_\mathcal{E})$ is indeed an LKNF-proof of $\vdash \cdot \uparrow \text{Sh}(\mathcal{E})$.

Definition 14 (Sequentialization). The LKNF-proof $\text{Sh}(\pi_\mathcal{E})$ constructed in the above proof of the sequentialization theorem will be denoted by $\text{Seq}(\mathcal{E})$.

4 Equivalence

A first central observation concerning the relationship of rule permutations and expansion trees is that the former do not change the latter.

Theorem 6. If $\pi_1$ and $\pi_2$ are LKNF-proofs with $\pi_1 \sim \pi_2$ then $\mathcal{E}(\pi_1) = \mathcal{E}(\pi_2)$.

Proof. Instead of spelling out the proof for every rule permutation, here is just the $\land/\exists$-case. Here, $\pi_1$ contains a subproof of the form (a) below, where $\mathcal{E}(\pi_1') = \mathcal{E}_1, E'$ and $\mathcal{E}(\pi_1'') = \mathcal{E}_2, E''$.

\[
\begin{align*}
(\pi_1') & \vdash \Gamma, A, [t/x]C \\
(\pi_1'') & \vdash \Gamma, B, [t/x]C \\
& \vdash \Gamma, A \land B, [t/x]C \land \\
& \vdash \Gamma, A \land B, \exists x. C \exists \\
\end{align*}
\]

By definition 11, the expansion sequent of this subproof is $\mathcal{E}_1 \cup \mathcal{E}_2, E_1 \land E_2, \exists x. C + {'} (E' \cup E'')$.

The corresponding subproof in $\pi_2$ has the form (b) above and the corresponding expansion sequent is $\mathcal{E}_1 \cup \mathcal{E}_2, E_1 \land E_2, (\exists x. C + {'} E') \cup (\exists x. C + {'} E'')$ which by definition 10 is equal to $\mathcal{E}_1 \cup \mathcal{E}_2, E_1 \land E_2, \exists x. C + {'} (E' \cup E'')$.

We now turn back to the sequentialization procedure for constructing an LKNF-proof from an expansion proof. The procedure used in Theorem 5 has been designed for producing only maximal proofs as shown in the following lemma.

Lemma 7. If $\mathcal{E}$ is an expansion proof, then $\text{Seq}(\mathcal{E})$ is maximal.

Proof. Suppose $\text{Seq}(\mathcal{E})$ is not maximal, then it contains a subproof $\pi$ ending with a decide inference s.t. there exists a proof $\pi'$ with $\pi \sim \pi'$ and $\text{foci}(\pi) \subseteq \text{multiset foci}(\pi')$. So there is an existential formula $\exists x. A$ in $\text{foci}(\pi') \setminus \text{foci}(\pi)$ to which in $\pi_\mathcal{E}$ corresponds an expansion $\exists x. A + {'} E'$. As rule permutations allow to shift down the instantiation of the expansion term $t$ over all $\forall$-inferences, the term $t$ must be $<_x$-minimal for $\mathcal{F}$ being the expansion sequent corresponding to the conclusion sequent of $\pi$ in $\text{Seq}(\mathcal{E})$. This is a contradiction to the choice of $\Delta''$ and $\Delta_1'$ made in definition 13.

Lemma 8. If $\pi$ is a maximal LKNF-proof, then $\pi \cong \text{Seq}(\mathcal{E}(\pi))$.  

13
Proof. We proceed by induction on the decision depth of \( \pi \). If \( \pi \) ends with a positive phase, it is of the form (a) below where the \( A_i \) are non-existential formulas and \( \pi' \cong \text{Seq}(\mathcal{E}(\pi')) \) by induction hypothesis.

\[
\begin{align*}
&\vdash \Gamma_1 \Downarrow \bar{f}_1 A_1, \ldots, \bar{f}_n A_n \\
&\vdash \Gamma_2 \Downarrow \exists x. A_1, \ldots, \exists x. A_n \\
&\vdash \Gamma_3 \Downarrow \\
&\vdash \Gamma_4 \Downarrow \\
\end{align*}
\]

As \( \pi \) is maximal, the existential inferences in this phase are in 1-1 correspondence to the \( \prec_{\mathcal{E}(\pi)} \)-minimal expansion terms of \( \mathcal{E}(\pi) \). Therefore, by definition 13, \( \text{Seq} \) creates the shown segment of \( \pi \) from \( \mathcal{E}(\pi) \) up to permutations of the existential inferences inside this segment.

If \( \pi \) ends with a negative phase, then it is of the form (b) above where \( \Delta \) does not contain an existential formula. If \( n = 0 \), then \( \pi \) consists only of this phase and we are done. Otherwise we have \( \pi_i \cong \text{Seq}(\mathcal{E}(\pi_i)) \) for \( i = 1, \ldots, n \) by induction hypothesis. For fixed \( \Delta \), the sequents \( \vdash \Gamma_1 \uparrow \Delta, \ldots, \vdash \Gamma_n \uparrow \Delta \) are uniquely determined and there are no decide and existential inferences in the negative phase so we obtain \( \pi \cong \text{Seq}(\mathcal{E}(\pi)) \).

A maximal proof corresponding to \( \pi \) can be obtained via rule permutations as in the first part of theorem 2. Reading off an expansion tree from \( \pi \) and then re-sequentializing this tree gives an alternative way to compute a maximal proof as the following theorem shows.

**Theorem 9.** For any LKNF proof \( \pi \): \( \text{Seq}(\mathcal{E}(\pi)) \in \text{max}(\pi) \).

**Proof.** By the first part of theorem 2 there is a \( \pi' \sim \pi \) with \( \pi' \in \text{max}(\pi) \). Applying lemma 8 to \( \pi' \) shows that \( \pi' \cong \text{Seq}(\mathcal{E}(\pi')) \) and hence \( \text{Seq}(\mathcal{E}(\pi')) \in \text{max}(\pi) \) but by theorem 6 we have \( \mathcal{E}(\pi') = \mathcal{E}(\pi) \), so we obtain \( \text{Seq}(\mathcal{E}(\pi)) \in \text{max}(\pi) \).

We can now finally obtain the equivalence of expansion trees and maximal proofs with respect to the induced identity notion for proofs. This theorem is our main technical result about proofs in first-order classical logic: the abstractions of LKNF proofs provided by expansion trees and by maximal multi-focusing are the same.

**Theorem 10.** Let \( \pi_1, \pi_2 \) be LKNF proofs. Then \( \mathcal{E}(\pi_1) = \mathcal{E}(\pi_2) \) iff \( \text{max}(\pi_1) = \text{max}(\pi_2) \).

**Proof.** For the left-to-right direction let \( E = \mathcal{E}(\pi_1) = \mathcal{E}(\pi_2) \). Theorem 9 then implies that that \( \text{Seq}(E) \) is in both \( \text{max}(\pi_1) \) and \( \text{max}(\pi_2) \), so \( \text{max}(\pi_1) = \text{max}(\pi_2) \). The right-to-left direction follows directly from theorem 6.

**5 Conclusion**

We have illustrated that, instead of discarding the sequent calculus in search of canonical proof systems, sequent proofs can be systematically abstracted by (maximal) multi-focusing into canonical structures. In this paper, we have imposed a particular focusing discipline on classical sequent proofs—negatively polarized propositional connectives and delayed literals—and have then showed that maximal multi-focusing in the sequent calculus yields the parallel and minimalistic notion of proofs based on expansion trees. Our framework is obviously generative as well: there are other polarizations within classical logic and in focused proof systems for intuitionistic and linear logics. Maximal multi-focusing yields different canonical structures for these other polarizations.
Acknowledgments  This research has been funded in part by the ERC Advanced Grant ProofCert and by the ANR-FWF project STRUCTURAL.

References


