Journal of Logic and Computation Advance Access published June 6, 2014

A multi-focused proof system isomorphic to expansion proofs

KAUSTUV CHAUDHURI, *INRIA, 1 rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France.*

STEFAN HETZL, *Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria.*

DALE MILLER, *INRIA and LIX/École Polytechnique, 1 rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France.*

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* formalizms. 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 for abstracting away low-level details from classical cut-free sequent proofs. We show that, among the multi-focused proofs, the *maximally multi-focused* proofs that collect together all possible parallel foci are canonical. Moreover, if we start with a certain focused sequent proof system, such proofs are isomorphic to *expansion proofs*—a well known, minimalistic and parallel generalization of Herbrand disjunctions—for classical first-order logic. This technique appears to be a systematic way to recover the 'essence of proof' from within sequent calculus proofs.

Keywords: Multi-focusing, expansion trees, canonicity.

1 Introduction

The sequent calculus, initially developed by Gentzen for classical and intuitionistic first-order logic [11], has become a standard proof formalizm 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, where every logical connective is defined by introduction rules and where all other inference rules are either structural rules (weakening/contraction) or identity rules (initial/cut). 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 [12] 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 logics can then be derived by analysing 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, sequent proofs themselves are syntactic monsters: they record the exact sequence of inferences rules even when such

doi:10.1093/logcom/exu030

details are not relevant to the essential high-level features of the proof. The most common approach taken to avoid the syntactic morass of the sequent calculus is one of *revolution*. New proof formalizms different from the sequent calculus are proposed that are supposedly free of *syntactic bureaucracy*. Usually, such formalizms are more parallel or geometric than sequent proofs. The following list of examples of such systems is not exhaustive.

(1) The *mating method* [2] and the *connection method* [5] represent proofs as a graph structure among the literals in (an expansion of) a formula.
(2) *Expansion trees* [29] record only the instantiations of quantifiers using a tree structure.
(3) *Proof-nets* [12] eschew inference rules for more geometric representations of proofs in terms of axiom and cut linkages.
(4) *Atomic flows* [14] track only the flow of atoms in a proof and can expose the dynamics of cut-elimination.
(5) Even Gentzen's *natural deduction* calculus [11, 34] is a more parallel representation of proofs given that trees play a more intimate role in the structuring of inferences.

While such formalizms are capable of abstracting away from many low-level syntactic details, it is worth noting that they are not without problems. At a basic level, showing when a proposed structure is correct—that it actually represents a 'proof'—requires checking global criteria such as connectedness, acyclicity or well-scoping. Such formalizms 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 formalizms 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*) [12]. Even adding the multiplicative units is tricky [24] and for larger fragments such as *MALL* with units the problem of finding a polynomial time checkable proof-net formalizm remains open.

In this article, we argue that many of the benefits of such revolutionary approaches can be achieved directly in the sequent calculus tradition by using a more *evolutionary* approach that involves selecting suitable abstractions. Our technique can be described using the following broad outline.

- We begin by limiting ourselves to cut-free *focused* proofs [1, 26]. Focusing is based on the observation that it is sufficient for provability to consider only those cut-free sequent proofs that are organized into an alternation of two kinds of *phases* for the principal formulas. Briefly, in the *positive phase*, information—such as witnesses for existential formulas or multiplicative splits of contexts—is added to the proof. This phase is inherently non-deterministic from a proof search perspective. The other, *negative phase* is a choice-free reduction of a given sequent to simpler premise sequents; this phase consumes no essential information. Once we commit to focused proofs, we can ignore details such as the precise manner in which the steps *inside* a phase are performed; only the boundaries between the phases are important.
- Focusing phases can sometimes permute over each other in a manner similar to the way inference rules can permute over each other. If two phases have no inter-dependencies and can be done in parallel, then it is possible to allow both phases to be merged into a single phase. To describe such parallel phases in a proof, we generalize focusing to *multi-focusing*, which enables the most important descriptive tool in our technique. Two multi-focused proofs that are equivalent in terms of the underlying rule permutations of the sequent calculus might nevertheless have different levels of parallelism in their phase structure. We choose to limit our attention to those multi-focused proofs where the phases are as parallel as possible (reading

from the end-sequent upwards), which we call the *maximally multi-focused* proofs (or often just *maximal proofs*).

- As a final step, we observe that if we choose the sequent rules carefully, then the maximal multi-focused proofs are both unique and syntactically canonical in the following sense: two permutatively equivalent multi-focused proofs have the *same* maximal form. It is important to note that focusing, multi-focusing, and maximality are general concepts that may be applied to essentially any cut-free sequent calculus: for example, in [7] these concepts are defined for multiplicative-additive linear logic (*MALL*). The uniqueness of maximal proofs is, however, sensitive to particular rule permutations.

In this article, we apply this technique to establish two new results.

(1) We give a multi-focused sequent calculus for classical first-order logic and show that the maximal proofs obtained therefrom are unique representatives of their permutative equivalence classes (Theorem 4.7). We give a precise condition for rule permutations under which such uniqueness theorems can be proven for any focused sequent calculus.

(2) We then show that such maximal proofs are isomorphic to expansion proofs [29], a generalization of Herbrand disjunctions for classical first-order (and even higher-order) logic. This result is surprising because it is known that expansion trees can be exponentially more compact than sequent proofs [4].

In Section 2, we give some background on the sequent calculus, on focusing, and on expansion trees. Section 3 introduces the focused sequent calculus *LKE* that will be used to develop our connection to expansion proofs. Section 4 then analyses the equivalence classes of sequent proofs that have the same expansion proof. This leads to a reverse mapping from expansion proofs to sequent proofs, called *sequentialization* (Section 4.3). Finally, Section 5 presents and discusses the isomorphism between maximal proofs and expansion proofs. Some related work is discussed in Section 6.

2 Sequent calculus, focusing, and expansion proofs

We use the usual syntax for (first-order) *formulas* $(A,B,\ldots)$ and connectives drawn from $\{\top,\wedge,\bot,\vee,\neg,\forall,\exists\}$. *Atomic formulas* $(a,b,\ldots)$ are of the form $p(t_1,\ldots,t_n)$ where p represents a predicate symbol and $t_1,\ldots,t_n$ are first-order terms $(n\geq 0)$. Formulas are restricted to negation-normal form (i.e. only atomic formulas can be $\neg$-prefixed) and two formulas are identical if they are α-equivalent. We use the term *literal* to refer to either an atomic formula or a negated atomic formula. We assume that all bound variables in a formula are pairwise distinct. We write $(A)^\perp$ to stand 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 $\exists\vec{x}.A$ for $\exists x_1.\ldots\exists x_n.A$, $\forall\vec{x}.A$ for $\forall x_1.\ldots\forall x_n.A$, and $[\vec{t}/\vec{x}]$ for $[t_1/x_1]\cdots[t_n/x_n]$ if $\vec{x}=(x_1,\ldots,x_n)$ and $\vec{t}=(t_1,\ldots,t_n)$.

2.1 *Sequent calculus:* LKN

We use one-sided sequents $\vdash\Gamma$ 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 $\exists$ are invertible. Since invertible rules are associated with the *negative* polarity in focusing, we use the *N* in *LKN* to highlight the fact that is a variant

$$\frac{}{\vdash \Gamma, \neg a, a}\ \mathsf{init} \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \wedge B}\ \wedge \quad \frac{}{\vdash \Gamma, \top}\ \top \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B}\ \vee \quad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\ \bot$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x.\, A}\ \forall \quad \frac{\vdash \Gamma, [t/x]A}{\vdash \Gamma, \exists x.\, A}\ \exists \quad \frac{\vdash \Gamma, \Delta}{\vdash \Gamma}\ \mathsf{contr}$$

Notes:

1. In the $\forall$ rule, the principal formula is implicitly α-converted so x is not free in the conclusion.
2. In the contr rule, $\emptyset \neq \Delta \subseteq_{\text{set}} \Gamma$. Here, $\Delta \subseteq_{\text{set}} \Gamma$ denotes the set inclusion of the underlying sets of the multisets Δ and Γ.

FIGURE 1. Rules of *LKN*.

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.

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, (A)^{\perp}}{\vdash \Gamma}\ \mathsf{cut} \qquad \frac{}{\vdash \Gamma, (A)^{\perp}, A}\ \mathsf{arbinit} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, A}\ \mathsf{weak} \qquad \frac{\vdash \Gamma}{\vdash [t/x]\Gamma}\ \mathsf{subst}$$

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 $A_1 \vee \cdots \vee A_n$.

2.2 *Focused sequent calculus:* LKF

In the 1980s, logic programming was placed on strong proof-theoretic foundations by describing the search for cut-free sequent proofs therein as an alternation of *goal-reduction* and *back-chaining* phases [30]. Andreoli [1] subsequently generalized this treatment to identify a class of *focused proofs* in the sequent calculus for classical linear logic. Comprehensive focused sequent calculi have since been built for intuitionistic and classical logics [21, 26], and focusing is increasingly being seen as a technique for unraveling the structure of proofs.

The *LKF* focused proof system, as presented in [26], deals with formulas in which the classical connectives and constants are divided into two disjoint and dual *polarity* classes: the *negatives* $\{\mathsf{t}, \&, \mathsf{f}, \invamp, \forall\}$ and the *positives* $\{\mathsf{1}, \otimes, \mathsf{0}, \oplus, \exists\}$.[1] A non-atomic formula is negative or positive if its top-level connective is negative or positive, respectively. Polarity is extended to all formulas by arbitrarily classifying atomic formulas as positive; hence, negated atoms are negative.

From the perspective of truth, there is no difference between the positive and the negative variants of a single unpolarized connective; i.e. $A \otimes B$ and $A \& B$ are equiprovable with $A \wedge B$, as are $A \oplus B$, $A \invamp B$ and $A \vee B$, etc . However, the introduction rules for the two polarized variants of a connective are different, which leads to different proofs for different choices of polarized variants of a connective. In general, the introduction rules for negative connectives are all invertible, meaning that the conclusion of any of these introduction rules is equivalent to its premises. The introduction rules for the positive connectives are not necessarily invertible.

Figure 2 contains a *multi-focused* variant of the *LKF* system from [26]. The two phases of such *LKF* proofs are depicted using two different sequent forms: *negative sequents* of the form $\vdash \Gamma \Uparrow \Delta$ and *positive sequents* of the form $\vdash \Gamma \Downarrow \Delta$. In either form, Γ is a multiset of literals or positive

[1] We use the glyphs $\otimes$, $\invamp$, etc . from linear logic even though their interpretation is classical.

Negative rules

$$\frac{}{\vdash \Gamma \Uparrow \Delta, \mathsf{t}} \qquad \frac{\vdash \Gamma \Uparrow \Delta, A \quad \vdash \Gamma \Uparrow \Delta, B}{\vdash \Gamma \Uparrow \Delta, A \mathbin{\&} B} \qquad \frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \Delta, \mathsf{f}} \qquad \frac{\vdash \Gamma \Uparrow \Delta, A, B}{\vdash \Gamma \Uparrow \Delta, A \mathbin{⅋} B}$$

$$\frac{\vdash \Gamma \Uparrow \Delta, A}{\vdash \Gamma \Uparrow \Delta, \forall x.\, A}$$

Positive rules

$$\frac{}{\vdash \Gamma \Downarrow 1} \qquad \frac{\vdash \Gamma \Downarrow \Delta, A \quad \vdash \Gamma \Downarrow \Theta, B}{\vdash \Gamma \Downarrow \Delta, \Theta, A \otimes B} \qquad \frac{\vdash \Gamma \Downarrow \Delta, A}{\vdash \Gamma \Downarrow \Delta, A \oplus B} \qquad \frac{\vdash \Gamma \Downarrow \Delta, B}{\vdash \Gamma \Downarrow \Delta, A \oplus B} \qquad \frac{\vdash \Gamma \Downarrow \Delta, [t/x]A}{\vdash \Gamma \Downarrow \Delta, \exists x.\, A}$$

Structural rules

$$\frac{\vdash \Gamma, L \Uparrow \Delta}{\vdash \Gamma \Uparrow \Delta, L}\ \mathsf{store} \qquad \frac{}{\vdash \Gamma, \neg a \Downarrow a}\ \mathsf{init} \qquad \frac{\vdash \Gamma \Downarrow \Delta}{\vdash \Gamma \Uparrow \cdot}\ \mathsf{decide} \qquad \frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Downarrow \Delta}\ \mathsf{release}$$

Notes:

1. In the $\forall$ rule, the principal formula is implicitly α-converted so x is not free in the conclusion.
2. In the store rule, L is a literal or a positive formula.
3. In the decide rule, Δ contains only positive formulas and $\emptyset \neq \Delta \subseteq_{\text{set}} \Gamma$.
4. In the release rule, Δ contains no positive formulas.

FIGURE 2. Rules for the multi-focused version of *LKF*.

formulas, and Δ is a multiset of arbitrary formulas. In the positive sequent $\vdash \Gamma \Downarrow \Delta$, we say that the formulas in Δ are its *foci* and we require Δ to be non-empty. (We write $\vdash \Gamma \Updownarrow \Delta$ to stand for either sequent form when describing *LKF* proofs.) The rules for positive sequents define a *positive phase*, and likewise those of the negative sequents define a *negative phase*. Mediating between the phases are the structural rules decide and release. A positive phase begins (reading bottom up) with a decide, followed by positive rules; eventually the foci become negative in which case the proof enters the negative phase with release; the negative phase consists of negative introduction rules for the negative connectives, or the store structural rule that transfers a literal or a positive formula to the other zone. Notice that unlike *LKN* where contraction may be applied arbitrarily, contraction is present in *LKF* only as part of the decide rule. As a result, the only formulas that are contracted in *LKF* are positive formulas.

We formally state the soundness and completeness of *LKF* with respect to *LKN* (and hence to Gentzen's *LK*) in terms of an injection.

DEFINITION 2.1
If B is a formula in *LKF*, then $[B]$ denotes the formula in which all polarized variants of connectives in B are mapped to their unpolarized variants, i.e. $\otimes$ and $\&$ to $\wedge$, $\oplus$ and $⅋$ to $\vee$, etc . If Γ is a multiset of formulas then $[\Gamma]$ is defined to be the multiset $\{[B] \mid B \in \Gamma\}$. If π is an *LKF* proof, we write $[\pi]$ for the *LKN* proof that:

- replaces all sequents of the form $\vdash \Gamma \Updownarrow \Delta$ with $\vdash [\Gamma], [\Delta]$;
- removes all instances of the rules store and release; and
- renames decide to contr in π.

THEOREM 2.2 (*LKF* vs. *LKN*)
(1) If π is an *LKF* proof of $\vdash \Gamma \Updownarrow \Delta$, then $[\pi]$ is an *LKN* proof of $\vdash [\Gamma], [\Delta]$ (soundness).
(2) If $\vdash [\Delta]$ is provable in *LKN*, then $\vdash \cdot \Uparrow \Delta$ is provable in *LKF* (completeness).

PROOF. Soundness is immediate by observing that $[-]$ preserves *LKN* validity. Completeness follows by observing that every singly focused proof in the *LKF* calculus of [26], which is complete for *LK* (and hence also for *LKN*), is also a proof in the multi-focused version of the calculus in Figure 2. ■

The *LKF* proof system can be seen as a framework for describing a range of focused proof systems for classical logic. The ordinary (unpolarized) connectives $\{\top,\wedge,\bot,\vee\}$ can be mapped to a positive variant $\{\mathsf{1},\otimes,\mathsf{0},\oplus\}$ or a negative variant $\{\mathsf{t},\&,\mathsf{f},\invamp\}$. Indeed, each occurrence of each unpolarized connective in a formula can be individually mapped to a positive or a negative variant in its polarized form. Different choices of polarization do not affect provability but can greatly affect the structure of proofs.

2.3 Expansion trees and expansion proofs

Herbrand's theorem [17] tells us that recording how quantifiers are instantiated is sufficient to describe a proof in classical first-order logic. Gentzen also noticed this for (cut-free) proofs of prenex normal sequents via the *mid-sequent* theorem [11]. Miller defined *expansion trees* [29] for full higher-order logic as a structure to record such substitution information without restriction to prenex normal form. We shall use a first-order version of this notion here.

DEFINITION 2.3
Expansion trees, *eigenvariables* and a function Sh (read *shallow formula of*) that maps an expansion tree to a formula are defined as follows:

(1) If $A\in\{\top,\bot\}$, or if A is a literal, then A is an expansion tree with top node A, and $\mathsf{Sh}(A)=A$.
(2) If E is an expansion tree with $\mathsf{Sh}(E)=[y/x]A$ and y is not an eigenvariable of any node in E, then $E'=\forall x.A+^{y}E$ is an expansion tree with top node $\forall x.A$ and $\mathsf{Sh}(E')=\forall x.A$. The variable y is called an *eigenvariable* of (the top node of) E'. The set of eigenvariables of all nodes in an expansion tree is called the *eigenvariables of* the tree.
(3) If $\{t_1,\ldots,t_n\}$ (with $n\geq 0$) is a set of terms and $E_1,\ldots,E_n$ are expansion trees with pairwise disjoint eigenvariable sets and with $\mathsf{Sh}(E_i)=[t_i/x]A$ for $i\in 1..n$, 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 $\mathsf{Sh}(E')=\exists x.A$. The terms $t_1,\ldots,t_n$ are known as the *expansion terms* of (the top node of) E'. The order of writing the expansions is immaterial; if $\phi:1..n\to 1..n$ is a permutation, then

$$\left(\exists x.A+^{t_1}E_1\cdots+^{t_n}E_n\right)\;=\;\left(\exists x.A+^{t_{\phi(1)}}E_{\phi(1)}\cdots+^{t_{\phi(n)}}E_{\phi(n)}\right).$$

(4) If E_1 and E_2 are expansion trees that share no eigenvariables and $\circ\in\{\wedge,\vee\}$, then $E_1\circ E_2$ is an expansion tree with top node $\circ$ and $\mathsf{Sh}(E_1\circ E_2)=\mathsf{Sh}(E_1)\circ\mathsf{Sh}(E_2)$.

We consider the eigenvariables of an expansion tree to be bound over the entire expansion tree, so systematic changes to eigenvariable names (α-conversion) result in equal trees. The requirement of eigenvariables in different subtrees being disjoint ensures that no eigenvariable is used to instantiate two different universal quantifiers within a given expansion tree. Sequent proofs are often described with a similar condition, known as *regularity*, that demands that any eigenvariable used in the $\forall$ rule be globally unique. Regularity is not essential for the sequent calculus because the correctness of each proof is locally checkable, so the same eigenvariable might be used in different branches of a proof. However, the correctness criterion for expansion trees (defined below) is global and hence needs globally unique variable names.

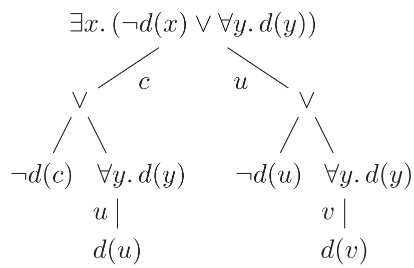
There is a simple way to coerce a formula into an expansion tree: use the bound variable of a universally quantified subformula as the eigenvariable of its corresponding expansion, and use the empty set of terms to expand an existentially quantified formula. Whenever we use a formula to denote an expansion tree, we shall assume that we use this coercion. It is also natural to include a notion of sequents of expansion trees.

DEFINITION 2.4
If $E_1,\ldots,E_n$ are expansion trees with pairwise disjoint eigenvariable sets, then $\mathcal{E} = E_1,\ldots,E_n$ is an *LKN expansion sequent*. The *shallow sequent of* $\mathcal{E}$, written $\mathsf{Sh}(\mathcal{E})$, is the *LKN* sequent $\vdash \mathsf{Sh}(E_1),\ldots,\mathsf{Sh}(E_n)$.

EXAMPLE 2.5
Consider the formula $D = \exists x.(\neg d(x) \vee \forall y.d(y))$. The expression

$$D +^{c} \big(\neg d(c) \vee (\forall y.d(y) +^{u} d(u))\big) +^{u} \big(\neg d(u) \vee (\forall y.d(y) +^{v} d(v))\big)$$

is an expansion tree. Observe that the two eigenvariables u and v used to expand $\forall y.d(y)$ are distinct, even though u is used to expand an existential elsewhere. The nature of expansion trees becomes more apparent if drawn as trees with labels on the arcs denoting eigenvariables or expansions terms:



DEFINITION 2.6 (Labels and Dominators)
In the expansion tree $\forall x.A +^{x} E$ (resp. in $\exists x.A +^{t_1} E_1 \cdots +^{t_n} E_n$), we say that x (resp. t_i) *labels* the top node of E (resp. E_i, for any $i \in 1..n$). A term t *dominates* a node in an expansion tree if it labels a parent node of that node in the tree. An expansion term t in $\mathcal{E}$ is said to be a *topmost term* of $\mathcal{E}$ if its corresponding existential expansion node is not dominated by any other expansion term in $\mathcal{E}$

Expansion trees as described are only a basic data structure for storing quantifier instances; not all of them denote proofs. We say that an expansion tree is *correct* if it indeed denotes a proof. The shallow formula of an expansion tree discards all the quantifier instances and is therefore not suitable for defining the correctness criterion; we will need the following representation that preserves the instances.

DEFINITION 2.7
For an expansion tree E, the quantifier-free formula $\mathsf{Dp}(E)$, called the *deep formula of* E, is defined as:

- $\mathsf{Dp}(E) = E$ if $E \in \{\top, \bot\}$ or if E is a literal;
- $\mathsf{Dp}(E_1 \circ E_2) = \mathsf{Dp}(E_1) \circ \mathsf{Dp}(E_2)$ for $\circ \in \{\wedge, \vee\}$;
- $\mathsf{Dp}(\forall x.A +^{y} E) = \mathsf{Dp}(E)$; and
- $\mathsf{Dp}(\exists x.A +^{t_1} E_1 \cdots +^{t_n} E_n) = \mathsf{Dp}(E_1) \vee \cdots \vee \mathsf{Dp}(E_n)$ if $n > 0$, and $\mathsf{Dp}(\exists x.A) = \bot$.

We write $\mathsf{Dp}(E_1,\ldots,E_n)$ to mean $\mathsf{Dp}(E_1) \vee \cdots \vee \mathsf{Dp}(E_n)$.

The correctness criterion also uses a dependency relation on expansion terms.

DEFINITION 2.8
Let $\mathcal{E}$ be an expansion tree or expansion sequent and let $<^0_{\mathcal{E}}$ be the binary relation on the occurrences of expansion terms in $\mathcal{E}$ defined by $t <^0_{\mathcal{E}} s$ 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 $<^0_{\mathcal{E}}$, is called the *dependency relation* of $\mathcal{E}$.

Viewed as a sequent proof, the dependency $t <_{\mathcal{E}} s$ means that all $\exists$ introductions with t as the witness term must be lower in the proof than those with s as the witness.

DEFINITION 2.9 (Correctness)
An expansion tree or an expansion sequent $\mathcal{E}$ is said to be *correct* if $<_{\mathcal{E}}$ is acyclic and $\mathsf{Dp}(\mathcal{E})$ is a tautology; we also say that $\mathcal{E}$ is an *expansion proof of* $\mathsf{Sh}(\mathcal{E})$.

EXAMPLE 2.10
Let E be the expansion tree of Example 2.5. It has two expansion terms: c and u. Observe that $c <_E u$ because the node labeled with c dominates the $\forall$-node with eigenvariable u. However $u \not<_E c$, so $<_E$ is acyclic. Furthermore, $\mathsf{Dp}(E) = \neg d(c) \vee d(u) \vee \neg d(u) \vee d(v)$, which is a tautology. So, E is an expansion proof of the formula $\mathsf{Sh}(E) = \exists x.(\neg d(x) \vee \forall y.d(y))$.

THEOREM 2.11
Let $\mathcal{E}$ be an expansion proof containing at least one expansion term. Then, one of the topmost occurrences of expansion terms of $\mathcal{E}$ is $<_{\mathcal{E}}$-minimal.

PROOF. Let S be the set of topmost occurrences of expansion terms of $\mathcal{E}$ and suppose that none of them is $<_{\mathcal{E}}$-minimal. That is, for every $s \in S$, there is an occurrence of an expansion term t in $\mathcal{E}$ such that $t <_{\mathcal{E}} s$. Let $s \in S$ be given and let t in $\mathcal{E}$ be such that $t <_{\mathcal{E}} s$. By Definition 2.8, every dominator t' of t also satisfies $t' <_{\mathcal{E}} s$. Since every occurrence of expansion terms in $\mathcal{E}$ is either in S or is dominated by some term in S, it must follow that there is an $s' \in S$ such that $s' <_{\mathcal{E}} s$. Therefore, for every $s \in S$ there is a $s' \in S$ such that $s' <_{\mathcal{E}} s$, i.e. there is an infinite $<_{\mathcal{E}}$-descending chain in S. But S is finite and $<_{\mathcal{E}}$ is acyclic, so this is impossible. ■

One important property of expansion proofs is that there is a straightforward mapping from *LKN* (or even *LK*) proofs to expansion proofs, defined by induction on the structure of *LKN* proofs. For the contracted formulas in instances of contr and the side formulas in instances of binary rules (i.e. $\wedge$), it will be necessary to *merge* two expansion trees of the same formula. To define merging formally, we slightly generalize the syntax of expansion trees to add a new kind of merging node.

DEFINITION 2.12
An *expansion tree with merges* is defined by the same inductive definition as expansion trees in Definition 2.3 to which we add the following clause:

(5) If E_1 and E_2 are expansion trees with merges that share no eigenvariables and have the same shallow formula, then $E_1 \sqcup E_2$ is an expansion tree with merges with top node $\sqcup$ (called a *merge node*), and $\mathsf{Sh}(E_1 \sqcup E_2) = \mathsf{Sh}(E_1)$.

Expansion sequents with merges are defined in the natural way.

We shall define a rewrite operation $\mapsto$ on expansion trees with merges that removes the merge nodes. Some care has to be taken in its definition, as illustrated by the following example.

EXAMPLE 2.13
Consider this expansion sequent with merges:

$$(\forall x A +^{u} E_1) \sqcup (\forall x A +^{v} E_2), \exists x B +^{f(u)} F_1 +^{f(v)} F_2.$$

When propagating the merge node into the subtrees of the two trees being merged, the two eigenvariables u and v will need to be united, say to a new eigenvariable w. As eigenvariables are global, the result of this union is that the two expansion terms $f(u)$ and $f(v)$ in the second element of the sequent will also be identified, violating the set-nature of the expansions of an existential formula. This will then require merging the two trees $[w/u, w/v]F_1$ and $[w/u, w/v]F_2$. Thus, reducing a merge node might cause new merge nodes to appear in other parts of the expansion tree or expansion sequent.

This example shows that not only do merges require changing eigenvariables, but also that performing such changes might induce new merges. Thus, the rewrite $\mapsto$ that removes merges will generally need to traverse the tree several times before normalizing.

DEFINITION 2.14 (Eigenvariable Substitution)
Let E be an expansion tree with merges. The expansion tree $\langle w/u\rangle E$ stands for that tree with merges that results from replacing the eigenvariable u with w. It is defined by structural induction on expansion trees with merges as follows.

(1) If $E \in \{\top, \bot\}$ or if E is a literal, then $\langle w/u\rangle E = [w/u]E$ (ordinary substitution).
(2) For $\circ \in \{\wedge, \vee, \sqcup\}$, $\langle w/u\rangle (E_1 \circ E_2) = \langle w/u\rangle E_1 \circ \langle w/u\rangle E_2$
(3) Let $\{s_1, \ldots, s_k\}$ be $\{[w/u]t_1, \ldots, [w/u]t_n\}$. Then,

$$\langle w/u\rangle \big(\exists x.A +^{t_1} E_1 \cdots +^{t_n} E_n\big) = \exists x.\langle w/u\rangle A \quad +^{s_1} \bigsqcup_{\substack{i \in 1..n \\ [w/u]t_i = s_1}} \langle w/u\rangle E_i \quad \cdots \quad +^{s_k} \bigsqcup_{\substack{i \in 1..n \\ [w/u]t_i = s_k}} \langle w/u\rangle E_i.$$

(4) $\langle w/u\rangle (\forall x.A +^{v} E) = \forall x.[w/u]A +^{[w/u]v} \langle w/u\rangle E.$

The merge operation is then defined in terms of the following rewrite on expansion trees (or expansion sequents) with merges.

DEFINITION 2.15 (Expansion Contexts)
An *expansion context*, written as $\mathcal{E}[\cdot]$, denotes an expansion tree or an expansion sequent with merges containing a single occurrence of a *hole* $\cdot$. If E is an expansion tree with merges that does not share any eigenvariables with $\mathcal{E}[\cdot]$, then $\mathcal{E}[E]$ stands for that expansion tree or expansion sequent with merges where the hole is replaced by E.

DEFINITION 2.16
The merge rewrite operation $\mapsto$ is generated from the following cases.

(1) If $E \in \{\top, \bot\}$ or if E is a literal, then $\mathcal{E}[E \sqcup E] \mapsto \mathcal{E}[E]$.
(2) $\mathcal{E}[(E_1 \circ E_2) \sqcup (E_1' \circ E_2')] \mapsto \mathcal{E}[(E_1 \sqcup E_1') \circ (E_2 \sqcup E_2')]$ for $\circ \in \{\wedge, \vee\}$.
(3) $\mathcal{E}[(\forall x.A +^{u} E) \sqcup (\forall x.A +^{w} E')] \mapsto \langle w/u\rangle \mathcal{E}[\forall x.A +^{w} (E \sqcup E')]$.

(4) Suppose $\{s_1,\ldots,s_m\}\cap\{t_1,\ldots,t_n\}=\emptyset$. Then,

$$\begin{aligned}\mathcal{E}\big[\ &(\exists x.A +^{r_1} E_1 \cdots +^{r_l} E_l +^{s_1} F_1 \cdots +^{s_m} F_m)\\ \sqcup\ &(\exists x.A +^{r_1} E_1' \cdots +^{r_l} E_l' +^{t_1} G_1 \cdots +^{t_n} G_n)\big] \mapsto\\ &\qquad \mathcal{E}\big[\exists x.A +^{r_1} (E_1 \sqcup E_1') \cdots +^{r_l} (E_l \sqcup E_l') +^{s_1} F_1 \cdots +^{s_m} F_m +^{t_1} G_1 \cdots +^{t_n} G_n\big].\end{aligned}$$

This definition is extended to expansion sequents with merges in the natural way.

THEOREM 2.17
The reduction system $\mapsto$ on expansion trees or sequents with merges is confluent and strongly normalizing. Its normal forms have no merge nodes.

PROOF. There are no critical pairs, so the reduction system is locally confluent. The system is strongly normalizing because every rewrite either reduces the number of eigenvariables or reduces a merge node to a finite number of simpler merge nodes for strict subtrees. Finally, it is immediate by inspection that all subtrees rooted at merge nodes can be reduced. ■

DEFINITION 2.18 (Substitution and Merging)
If E_1 and E_2 are expansion trees that have the same shallow formula and that share no eigenvariables, then their *merge*, written $E_1 \cup E_2$, is the unique (up to renaming of eigenvariables) normal form of $E_1 \sqcup E_2$ under $\mapsto$. If E is an expansion tree, then $[w/u]E$ is defined to be the unique normal form of $\langle w/u\rangle E$ under $\mapsto$. These constructions are lifted to expansion sequents in the natural way.

We can now use merges to define an explicit function from *LKN* proofs to expansion proofs.

DEFINITION 2.19
The *expansion sequent of* an *LKN* proof π, written $\mathsf{Exp}(\pi)$, is given by induction on the structure of π. It has the following cases.

(1) If π is a proof of $\vdash \Gamma$ by init or $\top$, then $\mathsf{Exp}(\pi)=\Gamma$ (using the trivial coercion of formulas to expansion trees).

(2) Suppose $\pi = \dfrac{\overset{\pi_A}{\vdash \Gamma, A} \quad \overset{\pi_B}{\vdash \Gamma, B}}{\vdash \Gamma, A\wedge B}\ \wedge$, $\mathsf{Exp}(\pi_A)=\mathcal{E}_A, E_A$, and $\mathsf{Exp}(\pi_B)=\mathcal{E}_B, E_B$, where E_A (resp. E_B) is the expansion tree corresponding to A (resp. E_B to B), and $\mathcal{E}_A$ (resp. $\mathcal{E}_B$) is the expansion sequent corresponding to Γ in the left (resp. right) premise. Then, $\mathsf{Exp}(\pi)=\mathcal{E}_A \cup \mathcal{E}_B, E_A \wedge E_B$.

(3) Suppose $\pi = \dfrac{\overset{\pi_A}{\vdash \Gamma, A}}{\vdash \Gamma, \forall x.A}\ \forall$, $\mathsf{Exp}(\pi_A)=\mathcal{E}, E$ where E is the expansion tree corresponding to A and $\mathcal{E}$ is the expansion sequent corresponding to Γ in the premise. Let y be an eigenvariable that does not occur in $\mathcal{E}$. Then, $\mathsf{Exp}(\pi)=\mathcal{E}, \forall x.A +^y [y/x]E$.

(4) Suppose $\pi = \dfrac{\overset{\pi_A}{\vdash \Gamma, [t/x]A}}{\vdash \Gamma, \exists x.A}\ \exists$ and $\mathsf{Exp}(\pi_A)=\mathcal{E}, E$ where E is the expansion tree corresponding to $[t/x]A$ and $\mathcal{E}$ is the expansion sequent corresponding to Γ in the premise. Then, $\mathsf{Exp}(\pi)= \mathcal{E}, \exists x.A +^t E$.

(5) Suppose $\pi = \dfrac{\overset{\pi'}{\vdash A_1,\ldots,A_n,\Delta}}{\vdash A_1,\ldots,A_n}\ \mathsf{contr}$ where Δ contains k_i copies of A_i (for $i \in 1..n$). Further suppose that $\mathsf{Exp}(\pi')=E_1,\ldots,E_n,F_{1,1},\ldots,F_{1,k_1},\ldots,F_{n,1},\ldots,F_{n,k_n}$ where E_i is the expansion tree corresponding to A_i, and $F_{i,1},\ldots,F_{i,k_i}$ are the expansion trees corresponding to the k_i copies

of A_i in Δ (for $i \in 1..n$). Then,

$$\mathsf{Exp}(\pi) = E_1 \cup \bigcup_{j \in 1..k_1} F_{1,j}, \quad \ldots, \quad E_n \cup \bigcup_{j \in 1..k_n} F_{n,j}.$$

(6) If π ends with a $\vee$ or a $\bot$ introduction rule, then $\mathsf{Exp}(\pi)$ is defined in the obvious way.

The expansion sequents constructed in this fashion have no merge nodes. We can extend this definition to *LKF* by setting $\mathcal{E}(\pi) = \mathcal{E}([\pi])$ (Definition 2.1) for any *LKF* -proof π.

THEOREM 2.20
If π is an *LKN* or an *LKF* proof, then $\mathsf{Exp}(\pi)$ is an expansion proof.

PROOF. That $\mathsf{Dp}(\mathsf{Exp}(\pi))$ is a tautology can be shown by structural induction on π and following Definition 2.19. Acyclicity of $<_{\mathsf{Exp}(\pi)}$ follows from the side condition of the $\forall$-rule in *LKN* (or *LKF*) and the appropriate choice of variable names in Definition 2.19. ■

3 Representing expansion proofs: LKE

The previous section ended with a mapping Exp from any sequent proof, focused or not, to expansion proofs. There is a dual operation, called *sequentialization*, that produces a sequent proof from an expansion proof. Expansion trees contain only the quantifier instances, so an expansion proof might be sequentialized to many different proofs that are all themselves mapped back to that proof by Exp. Indeed, the merge operation used to define Exp combines duplicated subproofs which can cause an exponential decrease in the size of the smallest proof [4].

The multi-focused *LKF* proof system has nearly all the ingredients for defining such a sequentialization operation. Because expansion trees elide the propositional structure, the corresponding sequent proofs cannot allow choice in the inference rules used to introduce the propositional connectives. In terms of *LKF* this means that all but the existential connectives must be treated as negative (i.e. invertible). The phases of the *LKF* proofs would then be an alternation of existential instantiations, which correspond to the expansions, and of a pure reduction of the sequent based on the other logical connectives which corresponds to the non-expansion arcs in the expansion tree.

Although this intuition is simple, it has some issues that break the isomorphism between expansion proofs and arbitrary *LKF* proofs restricted to negative connectives.

(A) In *LKF* there is only a single proof of $\vdash \cdot \Uparrow \neg p(a), \exists x.p(x)$:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\vdash \neg p(a), \exists x.p(x) \Downarrow p(a)}\ \mathsf{init}}{\vdash \neg p(a), \exists x.p(x) \Downarrow \exists x.p(x)}\ \exists}{\vdash \neg p(a), \exists x.p(x) \Uparrow \cdot}\ \mathsf{decide}}{\vdash \cdot \Uparrow \neg p(a), \exists x.p(x)}\ \mathsf{store} \times 2$$

All the steps in the proof are forced; in particular, the proof must finish with an init after applying decide, which prevents all but the single instance of the existential formula. However, there are infinitely many expansion proofs of $\neg p(a) \vee \exists x.p(x)$ that simply differ in their expansions of the existential formula.

(B) Similarly, in *LKF* there is only (essentially) a single proof of $\vdash \cdot \Uparrow \mathsf{t}, \exists x. \neg p(x)$, which does not instantiate the existential formula, but there are infinitely many expansion proofs of $\top \vee \exists x. \neg p(x)$ with different instances of the existential.

(C) In every expansion proof of $p(a) \vee \exists x. \neg p(x)$, the existential is expanded by a *set* of witnesses, i.e. each of its expansions corresponds to a different witness term. On the other hand, in an *LKF* proof of $\vdash \cdot \Uparrow p(a) \vee \exists x. \neg p(x)$ it is possible to have the intermediate sequent $\vdash p(a), \exists x. \neg p(x), \neg p(b), \neg p(b) \Uparrow \cdot$ which corresponds to expanding the existential by the same witness term b twice:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\vdash p(a), \exists x. \neg p(x), \neg p(b), \neg p(b) \Uparrow \cdot}{\vdash p(a), \exists x. \neg p(x) \Uparrow \neg p(b), \neg p(b)}\ \mathsf{store} \times 2}{\vdash p(a), \exists x. \neg p(x) \Downarrow \neg p(b), \neg p(b)}\ \mathsf{release}}{\vdash p(a), \exists x. \neg p(x) \Downarrow \exists x. \neg p(x), \exists x. \neg p(x)}\ \exists \times 2}{\vdash p(a), \exists x. \neg p(x) \Uparrow \cdot}\ \mathsf{decide}}{\vdash \cdot \Uparrow p(a), \exists x. \neg p(x)}\ \mathsf{store} \times 2$$

In other words, issues (A) and (B) indicate that expansion proofs can contain more irrelevant 'junk' than the *LKF* proofs, while issue (C) illustrates that *LKF* proofs inherently treat the expansions as a multiset rather than as a set.

Focusing in *LKF* is aggressive by design. Issues (A) and (B) demonstrate that we need to dampen the effects of focusing in *LKF* somewhat: in particular, the init and t introduction rules finish the proof too early, before the existentials in the context can be instantiated. To relax these rules, we use the standard mechanism of *delaying* connectives that end the phase instead of the proof (by release or store, as appropriate) so that the formulas in the context can be focused on.

DEFINITION 3.1 (Delays)
The unary connectives $\mathcal{C}(-)$ or $\partial(-)$, standing for *positive delays* or *negative delays* respectively, are defined as follows

$$\mathcal{C}(A) = \exists x.A \qquad\qquad \partial(A) = \forall x.A$$

where x is selected in some systematic fashion to be not free in A.

All subformulas of A are also subformulas of $\mathcal{C}(A)$ or $\partial(A)$; moreover, $\mathcal{C}(A)$ and $\partial(A)$ are equiprovable and $[\mathcal{C}(A)] \equiv [\partial(A)]$ (Definition 2.1). If $\partial(A)$ is present in the focus of a positive sequent, then no further positive rules are applicable to it; eventually, after release is applied, the vacuous universal quantifier will be removed before further negative processing of A. Likewise, if $\mathcal{C}(A)$ is present to the right of $\Uparrow$ in a negative sequent, then the only rule applicable to it is store, after which a subsequent decide on it will copy $\mathcal{C}(A)$ before instantiating the vacuous existential quantifier and further positive processing of A. Observe that $\partial(A)$ where A is already negative has essentially the same behaviour as A; likewise for $\mathcal{C}(A)$ where A is already positive.

We will only use $\partial(-)$ for our purposes. Formally we define a pair of encodings from unpolarized (*LKN*) formulas to polarized (*LKF*) formulas that track where in the sequent the formula would end up in an *LKF* proof.

Definition 3.2
The pair of maps $(-)^{\rfloor}$ and $(-)^{\lceil}$ from unpolarized to polarized formulas are recursively defined as follows.

$$
\begin{aligned}
(A\vee B)^{\rfloor} &= (A)^{\rfloor} \parr (B)^{\rfloor} & \qquad (A\vee B)^{\lceil} &= (A)^{\lceil} \parr (B)^{\lceil}\\
(\bot)^{\rfloor} &= \mathsf{f} & (\bot)^{\lceil} &= \mathsf{f}\\
(A\wedge B)^{\rfloor} &= (A)^{\rfloor} \,\&\, (B)^{\rfloor} & (A\wedge B)^{\lceil} &= (A)^{\lceil} \,\&\, (B)^{\lceil}\\
(\top)^{\rfloor} &= \mathsf{1} & (\top)^{\lceil} &= \partial(\mathsf{1})\\
(\forall x.A)^{\rfloor} &= \forall x.\,(A)^{\rfloor} & (\forall x.A)^{\lceil} &= \forall x.\,(A)^{\lceil}\\
(\exists x.A)^{\rfloor} &= \exists x.\,(A)^{\lceil} & (\exists x.A)^{\lceil} &= \exists x.\,(A)^{\lceil}\\
(a)^{\rfloor} &= a & (a)^{\lceil} &= \partial(a)\\
(\neg a)^{\rfloor} &= \neg a & (\neg a)^{\lceil} &= \neg a
\end{aligned}
$$

These maps are naturally lifted to multisets of formulas.

These definitions are based on the intuition that *LKF* proofs of sequents of the form $\vdash (\Gamma)^{\rfloor} \Updownarrow (\Delta)^{\lceil}$ will correspond to expansion proofs. Existential formulas, atoms, and $\top$ are the only formulas that are translated to positive formulas, and the latter two are only positive in the $(-)^{\rfloor}$ translation, i.e. to the left of $\Updownarrow$. Because $(a)^{\lceil} = \partial(a)$, whenever $(a)^{\lceil}$ occurs among the foci, the init rule of *LKF* is prevented and it will eventually have to be $\mathsf{release}$d (after removing the ∂) and then the atom a (which is the same as $(a)^{\rfloor}$) is stored. This solves issue (A), because, in the example above, the $\exists$ introduction rule is now followed by $\mathsf{release}$ instead of init, which enables future decides on the existential formula. Issue (B) is solved similarly by preventing t from occurring in the image of the translation; whenever $\partial(\mathsf{1})$ appears on the right in a negative sequent, it will need to be stored (after stripping the ∂).

This leaves just issue (C). In the unfocused calculus *LKN*, where contraction is freely available, it is never necessary to instantiate an existential formula the same way twice, as one can simply contract the instantiated version instead. Expansion trees therefore treat the expansions (i.e. instantiations) of existential formulas as a set rather than as a multiset. It is a simple matter to add a restriction to the $\exists$ introduction rule of *LKN* that prevents duplicated copies:

$$
\frac{[t/x]A \notin \Gamma \quad \vdash \Gamma, [t/x]A}{\vdash \Gamma, \exists x.A}
$$

In the focused setting, such a restriction would break completeness because the foci themselves are not necessarily contractible. Consider, for instance, the formula $\exists x.\exists y.\neg p(x,y)$. In *LKN*, one could instantiate the outer existential with a to get $\exists y.\neg p(a,y)$ which is then contracted and instantiated with b and c to get $\neg p(a,b)$ and $\neg p(a,c)$. In *LKF*, we would instead have to contract the outermost existential formula twice and instantiate the *vector* (x,y) with (a,b) and (a,c), which repeats the instantiation of x by a.

Nevertheless, it is possible to recover a kind of non-redundant instantiation of existentials in *LKF* if we restrict the $\mathsf{release}$ rule to check that a block of existentials from a formula that was decided on have not been instantiated in the same way more than once, either in the same or in an earlier positive phase. To make this restriction formal we would require naming and tracking subformula relationships in the proof system, which is tedious but straightforward. Instead of taking this formal approach, we simply stipulate that all *LKF* proofs mentioned in the rest of the article implicitly have

Propositional

$$\frac{\vdash \Gamma \Uparrow \Delta, A \quad \vdash \Gamma \Uparrow \Delta, B}{\vdash \Gamma \Uparrow \Delta, A \wedge B}\ \wedge \qquad \frac{\vdash \Gamma \Uparrow \Delta, A, B}{\vdash \Gamma \Uparrow \Delta, A \vee B}\ \vee \qquad \frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \Delta, \bot}\ \bot \qquad \frac{\vdash \Gamma \Uparrow \Delta, A}{\vdash \Gamma \Uparrow \Delta, \forall x.\, A}\ \forall$$

Existential

$$\frac{\vdash \Gamma \Downarrow \Delta, [t/x]A}{\vdash \Gamma \Downarrow \Delta, \exists x.\, A}\ \exists$$

Structural

$$\frac{}{\vdash \Gamma, \top \Uparrow \cdot}\ \top \qquad \frac{}{\vdash \Gamma, \neg a, a \Uparrow \cdot}\ \mathsf{init}$$

$$\frac{\vdash \Gamma, L \Uparrow \Delta}{\vdash \Gamma \Uparrow \Delta, L}\ \mathsf{store} \qquad \frac{\vdash \Gamma \Downarrow \Delta}{\vdash \Gamma \Uparrow \cdot}\ \mathsf{decide} \qquad \frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Downarrow \Delta}\ \mathsf{release}$$

Notes:

1. In the $\forall$ rule, the principal formula is implicitly α-converted so x is not free in the conclusion.
2. In the store rule, L is $\top$, a literal, or an existential formula.
3. In the decide rule, Δ contains only existential formulas and $\emptyset \neq \Delta \subseteq_{\mathrm{set}} \Gamma$.
4. In the release rule, Δ contains no existential formulas. Moreover, every formula in Δ corresponds to a unique block of existential instantiations of a subformula of the end-sequent in all the existential phases below.

FIGURE 3. Rules of *LKE*.

this restriction on the release rules. (We state this restriction as a side condition to the release rule in the *LKF* -related proof system *LKE* in Figure 3.) As with *LKN*, this restriction does not break completeness: contraction is available for the context Γ in $\vdash \Gamma \Updownarrow \Delta$, so one can always reuse the instances.

The reader may wonder why issue (C) is not dealt with by using a variant definition of expansion trees that uses multisets or lists of expansions as in [29]. We use sets for the following reason: we heavily rely on Theorem 4.2 (proved in the next section) which states that rule permutations do not change the associated expansion proof. This result would not be true in a setting of expansion trees based on multisets or lists. Consider the following rule permutation:

$$\cfrac{\cfrac{\begin{matrix}\pi_1\\ \vdash \Gamma, A, [t/x]C\end{matrix} \quad \begin{matrix}\pi_2\\ \vdash \Gamma, B, [t/x]C\end{matrix}}{\vdash \Gamma, A \wedge B, [t/x]C}\ \wedge}{\vdash \Gamma, A \wedge B, \exists x.C}\ \exists \quad \sim \quad \cfrac{\cfrac{\begin{matrix}\pi_1\\ \vdash \Gamma, A, [t/x]C\end{matrix}}{\vdash \Gamma, A, \exists x.C}\ \exists \quad \cfrac{\begin{matrix}\pi_2\\ \vdash \Gamma, B, [t/x]C\end{matrix}}{\vdash \Gamma, B, \exists x.C}\ \exists}{\vdash \Gamma, A \wedge B, \exists x.C}\ \wedge$$

In the proof on the left, there is only one instance of $\exists x.C$, but there are two in the proof on the right. Because we want to admit this permutation, the common expansion tree representing both proofs must ignore the order and the multiplicity of the expansions.

While we can in principle continue using *LKF* and this encoding as our proof system, it will serve our purposes better to define a version of *LKF*, which we call *LKE*, specialized for the above encodings $(-)^{\uparrow}$ and $(-)^{\downarrow}$. The rules of *LKE* are displayed in Figure 3. Like *LKF*, the rules of *LKE* can be divided into three classes. The *propositional* rules contain almost all the negative rules of *LKF*, except for t (which does not exist in the image of the encodings). Every propositional rule has at least one premise, and no atomic sub-formulas are lost when moving from conclusion to premises.

The positive phase of *LKF* is present in *LKE* in only a degenerate *existential* phase consisting of a single rule. The remaining connectives, *viz.* positive atoms and 1, have specialized rules incorporating their focused *LKF* behaviour; in either case, the formula must be the sole principal formula of an *LKF* decide instance, after which the proof branch must immediately terminate with init or the 1 (i.e. $(\top)^{\downarrow}$) introduction rule, respectively. These derived positive *LKF* phases are added as new *structural* rules to *LKE*. The decide rule in *LKE* therefore only copies existential formulas into the foci, possibly more than once. The remaining structural rules of store and release are the same as in *LKF*.

THEOREM 3.3
The *LKE* sequent $\vdash \Gamma \Updownarrow \Delta$ is derivable in *LKE* iff the *LKF* sequent $\vdash (\Gamma)^{\rfloor} \Updownarrow (\Delta)^{\lceil}$ is derivable in *LKF*.

PROOF. A simple induction on the structure of proofs in *LKE* of the sequent $\vdash \Gamma \Updownarrow \Delta$ yields a proof in *LKF* of the sequent $\vdash (\Gamma)^{\rfloor} \Updownarrow (\Delta)^{\lceil}$. The converse is similarly proved by an induction on the structure of *LKF* proofs of sequents of the form $\vdash (\Gamma)^{\rfloor} \Updownarrow (\Delta)^{\lceil}$. ■

DEFINITION 3.4
For any *LKE* proof π, we write $[\pi]$ for that *LKN* proof that:

- replaces all sequents of the form $\vdash \Gamma \Updownarrow \Delta$ with $\vdash \Gamma, \Delta$;
- removes all instances of the rules store and release; and
- renames decide to contr in π.

THEOREM 3.5 (*LKE* vs. *LKN*)
(1) If π is an *LKE* proof of $\vdash \Gamma \Updownarrow \Delta$, then $[\pi]$ is an *LKN* proof of $\vdash \Gamma, \Delta$ (soundness).
(2) If $\vdash \Delta$ is provable in *LKN*, then $\vdash \cdot \Uparrow \Delta$ is provable in *LKE* (completeness).

PROOF. A corollary of Theorems 2.2 and 3.3. ■

4 Permutations, maximality and sequentialization

4.1 Permutations

Because expansion proofs record only the quantifier instances, they are more syntactically canonical than *LKN* proofs: two *LKN* proofs that only differ in a trivial order of inference rules are mapped by Exp to the same expansion tree. The pre-image of Exp defines an equivalence class of *LKN* proofs that are all represented by the same expansion proof. These equivalence classes correspond to a phenomenon that is well studied in the literature on the sequent calculus, that of permutations of inference rules in a sequent proof.

DEFINITION 4.1 (Permutations in *LKN*)
Two proofs π and π' of the same *LKN* sequent are *permutatively equivalent*, written $\pi \sim \pi'$, if the equivalence can be established as the reflexive-transitive-symmetric-congruence closure of the following *local rule permutations*.

(1) Permutations of introduction rules: these are permutations where the order of two neighbouring introduction rules can be locally switched. The following is a characteristic example, where an $\exists$ following a $\wedge$ introduction can be rewritten to an $\wedge$ introduction following two $\exists$ introductions.

$$\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdash \Gamma, A, [t/x]C\end{array} \quad \begin{array}{c}\pi_2\\ \vdash \Gamma, B, [t/x]C\end{array}}{\vdash \Gamma, A \wedge B, [t/x]C}\,\wedge}{\vdash \Gamma, A \wedge B, \exists x.C}\,\exists \quad \sim \quad \dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdash \Gamma, A, [t/x]C\end{array}}{\vdash \Gamma, A, \exists x.C}\,\exists \quad \dfrac{\begin{array}{c}\pi_2\\ \vdash \Gamma, B, [t/x]C\end{array}}{\vdash \Gamma, B, \exists x.C}\,\exists}{\vdash \Gamma, A \wedge B, \exists x.C}\,\wedge$$

(2) Permutations of structural rules: the contr rule permutes with other contr rules and can also be used to merge two neighbouring instances into one common instance:

$$\dfrac{\dfrac{\vdash \Gamma,\Delta,\Theta}{\vdash \Gamma,\Delta}\,\mathsf{contr}}{\vdash \Gamma}\,\mathsf{contr} \quad\sim\quad \dfrac{\vdash \Gamma,\Delta,\Theta}{\vdash \Gamma}\,\mathsf{contr} \quad\sim\quad \dfrac{\dfrac{\vdash \Gamma,\Delta,\Theta}{\vdash \Gamma,\Theta}\,\mathsf{contr}}{\vdash \Gamma}\,\mathsf{contr}$$

As a restriction, we prevent the init and contr rules from permuting, i.e.

$$\dfrac{\overline{\vdash \Gamma,\Delta,a,\neg a}\;\mathsf{init}}{\vdash \Gamma,a,\neg a}\,\mathsf{contr} \quad\not\sim\quad \overline{\vdash \Gamma,a,\neg a}\;\mathsf{init}$$

(3) Permutations of introduction and structural rules: when an introduction rule switches places with a contraction, the contraction may need to be duplicated.

$$\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdash \Gamma,\Delta,A\end{array} \quad \begin{array}{c}\pi_2\\ \vdash \Gamma,\Delta,B\end{array}}{\vdash \Gamma,\Delta,A\wedge B}\,\wedge}{\vdash \Gamma,A\wedge B}\,\mathsf{contr} \quad\sim\quad \dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdash \Gamma,\Delta,A\end{array}}{\vdash \Gamma,A}\,\mathsf{contr} \quad \dfrac{\begin{array}{c}\pi_2\\ \vdash \Gamma,\Delta,B\end{array}}{\vdash \Gamma,B}\,\mathsf{contr}}{\vdash \Gamma,A\wedge B}\,\wedge$$

Note that in the instance of contr in the left derivation, $\Delta \subseteq_{\text{set}} \Gamma, A\wedge B$, while in those of the right derivations, $\Delta \subseteq_{\text{set}} (\Gamma,A)$ and $\Delta \subseteq_{\text{set}} (\Gamma,B)$. So, in general only $\Delta \subseteq_{\text{set}} \Gamma$.
As a restriction, we prevent $\top$ introduction from permuting with contr, i.e.

$$\dfrac{\overline{\vdash \Gamma,\Delta,\top}\;\top}{\vdash \Gamma,\top}\,\mathsf{contr} \quad\not\sim\quad \overline{\vdash \Gamma,\top}\;\top$$

Observe that the two restricted permutations, init/contr and $\top$/contr, would otherwise be the only permutations that could delete a contracted copy of a formula (and its associated subproof) from an *LKN* proof. As contractions are used to implement decides in *LKF* and *LKE*, which are in turn the representatives of expansions, allowing permutations to delete contractions would break the following important theorem.

THEOREM 4.2
If $\pi_1 \sim \pi_2$, then $\mathsf{Exp}(\pi_1) = \mathsf{Exp}(\pi_2)$.

PROOF. By inspection of Definitions 2.19 and 4.1, each local permutation preserves Exp. We give here a representative case of $\wedge/\exists$ permutations, with:

$$\pi_1 = \dfrac{\dfrac{\begin{array}{c}\pi_A\\ \vdash \Gamma,A,[t/x]C\end{array} \quad \begin{array}{c}\pi_B\\ \vdash \Gamma,B,[t/x]C\end{array}}{\vdash \Gamma,A\wedge B,[t/x]C}\,\wedge}{\vdash \Gamma,A\wedge B,\exists x.C}\,\exists \qquad \text{and} \qquad \pi_2 = \dfrac{\dfrac{\begin{array}{c}\pi_A\\ \vdash \Gamma,A,[t/x]C\end{array}}{\vdash \Gamma,A,\exists x.C}\,\exists \quad \dfrac{\begin{array}{c}\pi_B\\ \vdash \Gamma,B,[t/x]C\end{array}}{\vdash \Gamma,B,\exists x.C}\,\exists}{\vdash \Gamma,A\wedge B,\exists x.C}\,\wedge$$

We have, by the inductive hypotheses, that $\mathsf{Exp}(\pi_A) = \mathcal{E}, E_A, E_{[t/x]C}$ and $\mathsf{Exp}(\pi_B) = \mathcal{F}, F_B, F_{[t/x]C}$ where E_A is an expansion tree for A; E_B is an expansion tree for B; $E_{[t/x]C}$ and $F_{[t/x]C}$ are expansion trees for $[t/x]C$; and $\mathcal{E}$ and $\mathcal{F}$ are expansion sequents for Γ. We then have $\mathsf{Exp}(\pi_1) = \mathcal{E} \cup \mathcal{F}, E_A \wedge F_B, \exists x.C +^t (E_{[t/x]C} \cup F_{[t/x]C}) = \mathsf{Exp}(\pi_2)$. ■

The converse is not true. For example, consider these two *LKN* proofs.

$$\dfrac{\dfrac{\dfrac{}{\vdash p(a),\neg p(a)}\,\mathsf{init}}{\vdash \exists x.p(x),\neg p(a)}\,\exists \qquad \dfrac{\dfrac{}{\vdash p(b),\neg p(b)}\,\mathsf{init}}{\vdash \exists x.p(x),\neg p(b)}\,\exists}{\vdash \exists x.p(x),\neg p(a)\wedge\neg p(b)}\,\wedge \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash p(a),p(b),\neg p(a)}\,\mathsf{init} \quad \dfrac{}{\vdash p(a),p(b),\neg p(b)}\,\mathsf{init}}{\vdash p(a),p(b),\neg p(a)\wedge\neg p(b)}\,\wedge}{\vdash \exists x.p(x),\exists x.p(x),\neg p(a)\wedge\neg p(b)}\,\exists\times 2}{\vdash \exists x.p(x),\neg p(a)\wedge\neg p(b)}\,\mathsf{contr}$$

Exp maps both proofs to the same expansion sequent $(\exists x.p(x)+^{a}p(a)+^{b}p(b)),\neg p(a)\wedge\neg p(b)$. However, the proofs are not permutatively equivalent because there is no local permutation that can change the order of the $\exists$ and $\wedge$ rules in the left derivation. Indeed, the numbers of contracted formulas are different in the two proofs, but none of our permutations can delete contracted copies. It is fairly obvious, therefore, that *LKN* simply has too many proofs if we want the permutative equivalence to characterize the identifications made by Exp.

We can also define an equivalence over *LKF* and *LKE* proofs in terms of rule permutations. Defining local permutations directly in the focused setting is difficult because cases such as $\mathsf{decide}/\mathsf{store}$ are simply impossible, so the permutations will have to be written in a so called *synthetic* form [6, 7]. This would be a technical and unilluminating detour for this article, so we just exploit Definition 2.1 to bootstrap the *LKF* and *LKE* permutative equivalence.

Definition 4.3
Two *LKF* or *LKE* proofs π_1 and π_2 of the same sequent are *permutatively equivalent*, written $\pi_1\sim\pi_2$, iff $[\pi_1]\sim[\pi_2]$ (see Definitions 2.1, 3.4 and 4.1).

This is not the only equivalence on focused proofs: there is at least one other equivalence that we can define based on just the phase structure of a focused proof. To motivate this definition, consider an *LKE* proof of $\vdash\Gamma\Uparrow\cdot$. Assuming the sequent is not proved by init or $\top$ introduction, it must be proved by a decide, which will enter the existential phase, then (after $\mathsf{release}$) the propositional phase, and finally be back to sequents of the form $\vdash\Gamma'\Uparrow\cdot$ after a number of stores. We can view this as an *action* (sometimes also called a *synthetic rule* or *bipole*) between *LKE* sequents of the form $\vdash\Gamma\Uparrow\cdot$, where we simply ignore all the rules except decide, init and $\top$. Two *LKE* proofs that have the same action structure should be considered *action equivalent*.

Definition 4.4
Two *LKE* proofs π_1 and π_2 of the same sequent are *action equivalent*, written $\pi_1\cong\pi_2$, iff they are tree-isomorphic for the instances of the decide, init and $\top$ rules.

Action equivalence gives us a different notion of the essence of an *LKE* proof that is independent of expansion trees and permutations. Because two action equivalent proofs have the same decide rules, one can reason about such proofs by induction on the *decision depth*—i.e. the nesting depth of the decide rules—in the *LKE* 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 would obtain a so called *synthetic proof* using synthetic rules [6].

4.2 Maximality

How are these two notions of equivalence related? One direction is obvious.

Theorem 4.5
If π_1 and π_2 are *LKE* proofs of the same sequent, then $\pi_1\cong\pi_2$ implies $\pi_1\sim\pi_2$.

PROOF. Up to local permutations, there is only a single way to derive an action. As π_1 and π_2 have the same actions, they must be permutatively equivalent. ■

In the other direction, two permutatively equivalent *LKE* proofs need not be action equivalent as they may perform the decide steps in a different order or with different foci. To illustrate, here are two permutatively equivalent *LKE* proofs that are not action equivalent (where $\Gamma = \exists x.p(x), \exists y.\neg p(f(y))$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \Gamma, p(f(c)), \neg p(f(c)) \Uparrow \cdot}\;\mathsf{init}}{\vdash \Gamma, p(f(c)) \Downarrow \exists y.\neg p(f(y))}\;\exists,\mathsf{release},\mathsf{store}}{\vdash \Gamma, p(f(c)) \Uparrow \cdot}\;\mathsf{decide}}{\vdash \Gamma \Downarrow \exists x.p(x)}\;\exists,\mathsf{release},\mathsf{store}}{\vdash \Gamma \Uparrow \cdot}\;\mathsf{decide} \qquad \dfrac{\dfrac{\dfrac{}{\vdash \Gamma, p(f(c)), \neg p(f(c)) \Uparrow \cdot}\;\mathsf{init}}{\vdash \Gamma \Downarrow \exists x.p(x), \exists y.\neg p(f(y))}\;\exists{\times}2,\mathsf{release},\mathsf{store}{\times}2}{\vdash \Gamma \Uparrow \cdot}\;\mathsf{decide}$$

However, each permutative equivalence class of *LKE* proofs does have a canonical (i.e. up to action equivalence) form where, intuitively, the foci of each decide rule are selected to be as numerous as possible. The proof on the right above, for example, has an instance of decide with more foci than the one on the left.

DEFINITION 4.6 (Maximality)
Given an *LKE* proof π that ends in an instance of decide, let $\mathrm{foci}(\pi)$ stand for the multiset of foci in the premise of that instance of decide. We say that this instance of decide is *maximal* iff for every $\pi' \sim \pi$, it is the case that $\mathrm{foci}(\pi') \subseteq_{\mathrm{multiset}} \mathrm{foci}(\pi)$. An *LKE* proof is maximal iff every instance of decide in it is maximal.

It follows directly from the definition that maximality is preserved by action equivalence. The two main properties of maximal proofs are that permutatively equivalent maximal proofs are also action equivalent, and that for every proof there is a permutatively equivalent maximal proof. Thus, the maximal proofs are canonical (action equivalent) representatives of their permutative equivalence classes.

THEOREM 4.7 (Canonicity)
(1) Every *LKE* proof has a permutatively equivalent maximal proof.
(2) Two permutatively equivalent maximal *LKE* proofs are action equivalent.

PROOF. Because init/contr and $\top$/contr permutations are disallowed in *LKN*, equivalent proofs have the same multiset union of all the foci of their decide rules. Using contr/contr permutations, the foci of the instances of decide can be divided or combined as needed. Therefore, there is a focus maximalization operation that, starting from the bottom of an *LKE* 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 4.6.

To see that two given permutatively 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 permutatively equivalent, these two decide instances themselves permute; hence, their foci can be merged, contradicting our assumption that they are maximal. ■

Similar theorems have appeared in [6, 7] for various fragments of multiplicative additive linear logic. It is an important feature of this proof that its argument is generic. It holds for any permutation system for a focused sequent calculus that can guarantee that foci are never deleted as part of a permutation.

DEFINITION 4.8
We write $\mathsf{max}(\pi)$ for the unique action equivalence class of maximal proofs that are permutatively equivalent to π (which exists by Theorem 4.7).

An example of the use of the canonicity theorem is Herbrand's theorem [17] for existential prenex formulas, which is a simple corollary of the completeness of *LKE* for classical first-order logic:

COROLLARY 4.9
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 $[\vec{t}_1/\vec{x}]A\vee\cdots\vee[\vec{t}_n/\vec{x}]A$ is valid.

PROOF. The if-direction is trivial. For proving the only if-direction, suppose $\exists\vec{x}.A$ is valid, i.e. the *LKN* sequent $\vdash\exists\vec{x}.A$ is provable. By Theorem 3.5 $\vdash\cdot\Uparrow\exists\vec{x}.A$ is provable in *LKE*, i.e. $\vdash\exists\vec{x}.A\Uparrow\cdot$ 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 4.7) performs only (at most) a single decide at the bottom, producing a number of focused copies of $\exists\vec{x}.A$. In the existential phase, the $\exists$s are removed from the foci to give the required term vectors. ■

4.3 Sequentialization

Thus far, we have shown that if $\pi_1\sim\pi_2$, then $\mathsf{Exp}(\pi_1)=\mathsf{Exp}(\pi_2)$ (Theorem 4.2) and $\mathsf{max}(\pi_1)=\mathsf{max}(\pi_2)$ (Theorem 4.7). In fact, we can show more: $\mathsf{Exp}(\pi)$ and $\mathsf{max}(\pi)$ are isomorphic. To do this, we will require a means of extracting *LKE* proofs from expansion proofs. We will directly extract a maximal *LKE* proof from an expansion proof, a step we call *sequentialization*. The definition consists of two phases: first we translate an expansion proof to a proof in an intermediate calculus *LKEE* which has the structure of *LKE* but uses expansion sequents instead of ordinary sequents. Secondly we map an *LKEE* proof π to an *LKE* proof $\mathsf{Sh}(\pi)$ which is defined by applying Sh to every expansion tree appearing in the *LKEE* proof. This operation will yield a valid *LKE* proof as the Sh image of an *LKEE* rule will be an *LKE* rule.

In slightly more detail, the sequents of *LKEE* are of the form $\vdash\mathcal{E}\Updownarrow\mathcal{F}$ where $\mathcal{E},\mathcal{F}$ is an expansion sequent. All the other rules of *LKE* except decide are adapted to expansion sequents in the natural way. To illustrate, here are the $\wedge$ and $\exists$ introduction rules in *LKEE* :

$$\frac{\vdash\mathcal{E}\Uparrow\mathcal{F},E\qquad\vdash\mathcal{E}\Uparrow\mathcal{F},F}{\vdash\mathcal{E}\Uparrow\mathcal{F},E\wedge F}\ \wedge\qquad\qquad\frac{\vdash\mathcal{E}\Downarrow\mathcal{F},E}{\vdash\mathcal{E}\Downarrow\mathcal{F},\exists x.A+^{t}E}\ \exists$$

The init and $\top$ rule of *LKEE* are also restricted to:

$$\frac{}{\vdash\mathcal{E},a,\neg a\Uparrow\cdot}\ \mathsf{init}\qquad\qquad\frac{}{\vdash\mathcal{E},\top\Uparrow\cdot}\ \top$$

where $\mathcal{E}$ contains only $\top$s, literals, or trivial existential trees i.e. trees of the form $\exists x.A$. Finally, for the decide rule for *LKEE* , we will use the following notational device.

Definition 4.10 (Expansion Vectors)
The *block notation* $\exists\vec{x}.A+^{\vec{t}_1}E_1\cdots+^{\vec{t}_n}E_n$ (where A is not an existential formula) is used to abbreviate those expansion trees where each $\vec{t}_i$ is a vector of expansion terms for $\vec{x}$, and E_i is an expansion tree for $[\vec{t}_i/\vec{x}]A$. For example, the expansion tree $\exists x.\exists y.p(x,y)+^{t}(\exists y.p(t,y)+^{s_1}p(t,s_1)+^{s_2}p(t,s_2))$ in the ordinary notation can be abbreviated as $\exists(x,y).p(x,y)+^{(t,s_1)}p(t,s_1)+^{(t,s_2)}p(t,s_2)$. Each $\vec{t}_i$ in $E=\exists\vec{x}.A+^{\vec{t}_1}E_1\cdots+^{\vec{t}_n}E_n$ is said to be an *expansion vector* of (the top node of) E. We say that an expansion vector $(t_1,\ldots,t_n)$ is *topmost* in an expansion tree if t_1 is a topmost expansion term of the tree.

Definition 4.11
The relation $<_{\mathcal{E}}$ on occurrences of expansion terms (Definition 2.8) is lifted to occurrences of expansion vectors in the natural way, i.e. $\vec{t}<_{\mathcal{E}}\vec{s}$ iff for every $t\in\vec{t}$ and $s\in\vec{s}$ it is the case that $t<_{\mathcal{E}}s$.

Theorem 2.11 generalizes to occurrences of expansion vectors.

Theorem 4.12
Let $\mathcal{E}$ be an expansion proof containing at least one expansion term. Then, one of the topmost occurrences of expansion vectors is $<_{\mathcal{E}}$-minimal.

Proof. Observe that the $<_{\mathcal{E}}$ relation lifted to occurrences of expansion vectors remains acyclic. Hence, the argument of Theorem 2.11 is just as applicable to expansion vectors. ■

The decide rule of *LKEE* is modified to focus on as many foci as possible as determined by the dependency relation on the expansion sequent in the conclusion. We will show below that this corresponds to maximal *LKE* proofs. Formally, the decide rule of *LKEE* is the following:

$$\frac{\vdash \mathcal{L},\mathcal{G}\Downarrow\mathcal{F}}{\vdash \mathcal{L},\mathcal{E}\Uparrow\cdot}\ \mathsf{decide}$$

where:

(i) $\mathcal{L}$ contains only $\top$s and literals;
(ii) $\mathcal{E}=E_1,\ldots,E_n$ where for every $i\in 1..n$,

$$E_i=\exists\vec{x}.A_i+^{\vec{s}_{i,1}}F_{i,1}\cdots+^{\vec{s}_{i,d_i}}F_{i,d_i}+^{\vec{t}_{i,1}}G_{i,1}\cdots+^{\vec{t}_{i,u_i}}G_{i,u_i}$$

and A_i is not an existential formula;
(iii) $\mathcal{F}=\mathcal{F}_1,\ldots,\mathcal{F}_n$ where for every $i\in 1..n$,

$$\mathcal{F}_i=\left(\exists\vec{x}.A_i+^{\vec{s}_{i,1}}F_{i,1}\right),\ldots,\left(\exists\vec{x}.A_i+^{\vec{s}_{i,d_i}}F_{i,d_i}\right);$$

(iv) $\mathcal{G}=G_1,\ldots,G_n$ where for every $i\in 1..n$,

$$G_i=\exists\vec{x}.A_i+^{\vec{t}_{i,1}}G_{i,1}\cdots+^{\vec{t}_{i,u_i}}G_{i,u_i};$$

(v) and for each $i\in 1..n, j\in 1..d_i$, the expansion vector $\vec{s}_{i,j}$ is $<_{\mathcal{L},\mathcal{E}}$-minimal (Definition 4.11).

Intuitively, the decide rule selects for focus those existential expansion trees from the conclusion sequent that corresponds to the minimal expansion vector, and then removes these expansion vectors from consideration in a subsequent decide above.

THEOREM 4.13
If $\mathcal{E} = E_1, \ldots, E_n$ is an expansion proof, then:

(1) $\vdash \cdot \Uparrow \mathcal{E}$ is derivable in *LKEE* , and
(2) $\vdash \cdot \Uparrow \mathsf{Sh}(E_1), \ldots, \mathsf{Sh}(E_n)$ is derivable in *LKE*.

PROOF. (2) follows from (1) as Sh maps *LKEE* proof rules to *LKE* proof rules. To show (1), we observe that an *LKEE* proof can be reconstructed for the end-sequent $\vdash \cdot \Uparrow \mathcal{E}$ without any non-deterministic choices. The instances of decide are determined by the dependency relation, and the instantiations of the $\exists$-inferences of *LKEE* are determined by the expansion trees in their respective conclusion sequents. As each rule of *LKEE* has the property that if the conclusion is an expansion proof then so is each individual premise (which is easily shown by inspection of the rules), since the end-sequent is an expansion proof it follows that every sequent in the *LKEE* derivation will also be an expansion proof. When the proof being reconstructed has no expansion terms, only the propositional phase applies which simply reduces the compound expansion trees to literals and $\top$; since these rules preserve the tautology of deep formulas, eventually each premise must have a $\top$ or a dual pair of literals, which are the basic tautologies. These branches can then be closed by $\top$ introduction or init.

Therefore, it suffices to show that we can always use a decide and the subsequent existential phase to remove at least one expansion term (if one exists) from the conclusion of the form $\vdash \mathcal{F} \Uparrow \cdot$ of an *LKEE* proof, so that the reconstruction can make progress. But, decide will always be applicable in this case by Theorem 4.12, as there is always at least one topmost expansion term that is $<_{\mathcal{F}}$-minimal. In our case, these topmost terms are the expansion terms of topmost existential nodes in $\mathcal{F}$. ■

DEFINITION 4.14 (Sequentialization)
Every expansion proof $\mathcal{E}$ has an *LKEE* proof $\pi_{\mathcal{E}}$ of $\vdash \cdot \Uparrow \mathcal{E}$ by Theorem 4.13. The *LKE* proof $\pi = \mathsf{Sh}(\pi_{\mathcal{E}})$ is called a *sequentialization of* $\mathcal{E}$, written $\mathsf{Seq}(\mathcal{E}, \pi)$.

Sequentialization is designed to produce only maximal proofs.

THEOREM 4.15
For any expansion proof $\mathcal{E}$, if $\mathsf{Seq}(\mathcal{E}, \pi)$ then π is maximal.

PROOF. Suppose $\mathsf{Seq}(\mathcal{E}, \pi_0)$ and π_0 is not maximal. Then, π_0 contains a subproof π ending with an instance of decide that is not maximal, i.e. there exists a proof $\pi' \sim \pi$ and $\mathrm{foci}(\pi) \subset_{\mathrm{multiset}} \mathrm{foci}\left(\pi'\right)$. This must mean that there is an existential formula $\exists x.A$ in $\mathrm{foci}\left(\pi'\right) \setminus \mathrm{foci}(\pi)$ for which there is an expansion term t in $\mathcal{E}$. Since the instance of $\exists$ for this formula was permutable by local permutations down to the instance of decide in π', it must be that t does not mention any of the eigenvariables in π introduced between this instance of decide and the instance of $\exists$ on $\exists x.A$. This in turn means that the term t is $<_{\mathcal{F}}$-minimal where $\mathcal{F}$ is the expansion sequent in the conclusion of the *LKEE* proof that corresponds to π. Hence, it must have been one of the expansion terms selected by decide in the *LKEE* proof, contradicting our assumption that the corresponding $\exists x.A \notin \mathrm{foci}(\pi)$. ■

5 Equivalence

We have seen in the canonicity theorem that every *LKE* proof is permutatively equivalent to a unique action equivalence class of maximal proofs. In this section we will show that these action equivalence classes are isomorphic to expansion proofs. Hence maximality identifies the same sequent proofs as are identified by expansion proofs, i.e. by the pre-image of Exp.

5.1 *Proof homomorphisms*

First, let us make precise our notion of isomorphism. We will consider mappings of proofs to proofs which are homomorphisms with respect *to the rules of LKE*. Note that this approach is different from categorical semantics of proofs where the proofs are interpreted as morphisms. For the purposes of this article, proofs are considered as objects. If φ is a homomorphism from *LKE* proofs to some data structure $\mathcal{S}$, then for each rule of *LKE*, ϕ must map every instance of that rule to an instance of an operation in $\mathcal{S}$. For example, if we have this *LKE* proof:

$$\pi = \cfrac{\begin{matrix}\pi_A \\ \vdash \Gamma \Uparrow \Delta, A\end{matrix} \qquad \begin{matrix}\pi_B \\ \vdash \Gamma \Uparrow \Delta, B\end{matrix}}{\vdash \Gamma \Uparrow \Delta, A \wedge B} \wedge$$

then there must be an operation $\star$ in $\mathcal{S}$ such that $\varphi(\pi) = \varphi(\pi_A) \star \varphi(\pi_B)$.

Concretely, we will consider $\mathsf{Exp} : LKE \to EP$ as our homomorphism where *EP* stands for the set of all expansion proofs, and the operations on *EP* are those of Definition 2.19.

LEMMA 5.1
For all $\mathcal{E} \in EP$, if $\mathsf{Seq}(\mathcal{E}, \pi)$ then $\mathsf{Exp}(\pi) = \mathcal{E}$.

PROOF. By a straightforward induction on $\mathcal{E}$. ■

Thus, Exp has a right-inverse that, for every $\mathcal{E} \in EP$, picks some π such that $\mathsf{Seq}(\mathcal{E}, \pi)$ (which is possible by Theorem 4.5). Hence, Exp is a surjective homorphism.

5.2 *Action equivalence classes*

To establish the isomorphism between action equivalence classes of maximal proofs and expansion proofs, we shall lift Exp, Seq and max to action equivalence classes by quotienting over $\cong$.

(i) As permutations do not affect Exp (Theorem 4.2) and action equivalence implies permutative equivalence (Theorem 4.5), it follows that the mapping $\widetilde{\mathsf{Exp}} : {}^{LKE}/_{\cong} \to EP$ is well defined. The operations on ${}^{LKE}/_{\cong}$ are the rules of *LKE* applied to permutative equivalence classes of *LKE*. $\widetilde{\mathsf{Exp}}$ is a homomorphism with respect to these operations.

(ii) In the other direction, $\widetilde{\mathsf{Seq}} : EP \to {}^{LKE}/_{\cong}$ is immediately defined by mapping $\mathcal{E}$ to the action equivalence class of some π for which $\mathsf{Seq}(\mathcal{E}, \pi)$.

(iii) Finally, we can lift $\mathsf{max} : LKE \to LKE$ to $\widetilde{\mathsf{max}} : {}^{LKE}/_{\cong} \to {}^{LKE}/_{\cong}$ in the natural way, which is possible by Theorem 4.5. As maximality is preserved by action equivalence, it follows that $\widetilde{\mathsf{max}}$ is idempotent.

5.3 *Maximal proofs*

Let *LKEM* stand for that fragment of *LKE* where every proof is maximal and whose end-sequent is of the form $\vdash \Gamma \Uparrow \cdot$.

LEMMA 5.2
If $\Pi \in {}^{LKEM}/_{\cong}$, then $\Pi = \widetilde{\mathsf{Seq}}(\widetilde{\mathsf{Exp}}(\Pi))$.

PROOF. Suppose $\pi \in \Pi$. We show that $\mathsf{Seq}(\mathsf{Exp}(\pi), \pi)$ by induction on the decision depth of π. The cases where π ends with $\top$ introduction or init are trivial. Otherwise, the bottom-most action in π has this form:

$$\dfrac{\dfrac{\begin{array}{ccc} \pi_1 & & \pi_m \\ \vdash \Gamma, \Delta_1 \Uparrow \cdot & \cdots & \vdash \Gamma, \Delta_m \Uparrow \cdot \end{array} \\ \vdots \\ \vdash \Gamma \Uparrow [\vec{t}_1/\vec{x}_1]A_1, \ldots, [\vec{t}_n/\vec{x}_n]A_n}{\vdash \Gamma \Downarrow [\vec{t}_1/\vec{x}_1]A_1, \ldots, [\vec{t}_n/\vec{x}_n]A_n}\ \mathsf{release} \\ \vdots \\ \vdash \Gamma \Downarrow \exists \vec{x}.A_1, \ldots, \exists \vec{x}.A_n}{\vdash \Gamma \Uparrow \cdot}\ \mathsf{decide}$$

where the A_i (for $i \in 1..n$) are non-existential formulas and $\mathsf{Seq}(\mathsf{Exp}(\pi_j), \pi_j)$ (for $j \in 1..m$) by the induction hypothesis. The expansion vectors $\vec{t}_i$ are all $<_{\mathsf{Exp}(\pi)}$-minimal because π is the Sh image of an *LKEE* proof (Definition 4.14). Moreover, all the $<_{\mathsf{Exp}(\pi)}$-minimal topmost terms occur among the $\vec{t}_i$, for otherwise there would be a permutatively equivalent *LKE* proof to π with more foci, contradicting the assumption that π is maximal. Therefore, $\mathsf{Seq}(\mathsf{Exp}(\pi), \pi)$. ■

THEOREM 5.3
$\widetilde{\mathsf{Exp}} : {}^{LKEM}\!/_{\cong} \to EP$ is an isomorphism with inverse $\widetilde{\mathsf{Seq}}$.

PROOF. We have already observed that $\widetilde{\mathsf{Exp}}$ is a homomorphism. By Lemma 5.1 we have $\widetilde{\mathsf{Exp}}(\widetilde{\mathsf{Seq}}(\mathcal{E})) = \mathcal{E}$ for all $\mathcal{E} \in EP$. Together with Lemma 5.2, this shows that $\widetilde{\mathsf{Exp}}$ has both a left and a right inverse, both of which are $\widetilde{\mathsf{Seq}}$. ■

Let us consider some concrete consequences of this isomorphism. We have seen that a maximal proof corresponding to π can be obtained via rule permutations as in the first part of Theorem 4.7. Reading off an expansion tree from π and then re-sequentializing this tree gives an alternative way to compute a maximal proof as the following theorem shows.

THEOREM 5.4
For any $\pi \in LKE$, $\widetilde{\mathsf{Seq}}(\mathsf{Exp}(\pi)) = \mathsf{max}(\pi)$.

PROOF. We have $\mathsf{Exp}(\pi) = \widetilde{\mathsf{Exp}}(\mathsf{max}(\pi))$ by Theorem 4.2. Therefore, by Theorem 5.3, $\widetilde{\mathsf{Seq}}(\mathsf{Exp}(\pi)) = \widetilde{\mathsf{Seq}}(\widetilde{\mathsf{Exp}}(\mathsf{max}(\pi))) = \mathsf{max}(\pi)$. ■

Furthermore, the abstractions of *LKE* proofs provided by expansion trees and by maximal multi-focusing are the same.

THEOREM 5.5
For $\pi_1, \pi_2 \in LKE$, $\mathsf{Exp}(\pi_1) = \mathsf{Exp}(\pi_2)$ iff $\mathsf{max}(\pi_1) = \mathsf{max}(\pi_2)$.

PROOF. For the left-to-right direction let $\mathcal{E} = \mathsf{Exp}(\pi_1) = \mathsf{Exp}(\pi_2)$. Theorem 5.4 then implies that $\mathsf{max}(\pi_1) = \widetilde{\mathsf{Seq}}(\mathcal{E}) = \mathsf{max}(\pi_2)$. The right-to-left direction follows directly from Theorem 4.2. ■

6 Related work

It is generally believed that classical logic lacks a denotational semantics for proofs akin to Cartesian-closed categories (CCC) for intuitionistic logic or $\star$-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) [25]. In terms of the sequent calculus, this problem manifests as follows: cut-elimination using Gentzen's cut-reduction rules is neither confluent nor strongly normalizing for *LK* proofs [3, 13, 19]. To force confluence, for instance, one would have to equate all cut-free proofs of a formula which again trivializes the semantics.

There have been both syntactic and semantic approaches to identifying classes of sequent proofs where such collapses do not occur. 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 in the sequent calculus [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 [38, 39]. Another approach is to carry out cut-elimination in a more abstract formalizm, similar to a proof-net, on the level of quantifiers [15, 28]. The reduction in such a setting is typically not confluent and strong normalization is open [28] or known not to hold [15]. Confluence (up to the equivalence relation of having the same expansion tree) as well as normalization can be recovered for a class of proofs [20] 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 [16] 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 [23, 37].

There are also a number of alternative answers to the question of when two *cut-free* sequent proofs are identical. Generally speaking, such answers are limited to the propositional fragment, and are primarily concerned with abstracting the propositional structure of sequent proofs [13, 22, 24, 27, 33, 35]. In the first-order case, it is more common to ignore the propositional structure and instead consider only the first-order content of proofs. Expansion trees [29], which are a generalization of Herbrand disjunctions, are perhaps the most minimalistic of such approaches as they record only the quantifier instances in a tree structure. (Indeed, the notion of expansion trees generalizes readily to even higher-order logic, which is the domain where it was initially developed.) The correctness criterion for expansion trees—that the deep formula is a tautology—is in co-NP. Specialized techniques such as the mating method [2] or the connection method [5] have been developed to represent these tautological checks using graph structures, but the worst case complexity of these techniques remains high.

To our knowledge, there has been only a single attempt to produce canonical proof structures directly in the sequent calculus, in this case for propositional *MALL* (with a certain restriction on $\top$) [7, 36]. This attempt also used multi-focusing as its abstraction mechanism, and it is actually the first place where the concept of maximal proofs appears in the literature. Multi-focusing was first proposed in [9, 31] as a natural extension of Andreoli's focusing system [1] for linear logic, and a similar concept has been independently developed in game semantics [32]. Although we have shown that maximal proofs are isomorphic to expansion proofs in this article, they can be exponentially larger than expansion proofs [4]. However, correctness of any sequent proof is easy to check as one simply needs to check that every inference in the proof is an instance of a proper rule schema. Indeed, even open (unfinished) sequent proofs can be seen to be correct, while the correctness condition for expansion trees only makes sense for completed proofs.

It is important to note that the notion of maximal proof strictly generalizes existing canonical forms in other contexts. For example, for intuitionistic logic, if one uses the focused sequent calculus *LJF* [26] with just the two negative connectives of implication and universal quantification and with negative atomic formulas, then maximal proofs are the same as singly focused proofs. Moreover, they are isomorphic to the β-normal η-long forms of the typed λ-calculus [10].

7 Conclusion

We have illustrated that, instead of discarding the sequent calculus in search of canonical proof systems, sequent proofs can be systematically abstracted into more canonical forms. In this paper, we have imposed a particular focusing discipline on classical sequent proofs—negatively polarized propositional connectives with minor use of delays—and have then showed that maximal multi-focusing in the sequent calculus yields the parallel and minimalistic notion of proofs based on expansion trees.

We leave untouched the question of maximality for the unrestricted permutations, i.e. without preventing $\top/\mathsf{contr}$ or $\mathsf{init}/\mathsf{contr}$ permutations. It is easy to show that, although maximal proofs do exist in this larger setting, they are not unique, and therefore the natural notion of equality for maximal proofs (action equivalence) does not provide canonical representatives for the permutative equivalence classes of maximal proofs. It is worth investigating the properties of such non-canonical maximal proofs. For example, are there natural geometric structures that correspond to maximal proofs in more permutatively permissive systems? Similar questions can be asked about the full *LKF* system, with both positive and negative propositional connectives, and for the related focused sequent calculi for intuitionistic logic and linear logic.

Acknowledgments

This research has been funded in part by the ERC Advanced Grant ProofCert and by the ANR-FWF project STRUCTURAL.

References

[1] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, **2**, 297–347, 1992.

[2] P. B. Andrews. Theorem proving via general matings. *Journal of ACM*, **28**, 193–214, 1981.

[3] M. Baaz and S. Hetzl. On the non-confluence of cut-elimination. *Journal of Symbolic Logic*, **76**, 313–340, 2011.

[4] M. Baaz, S. Hetzl, and D. Weller. On the complexity of proof deskolemization. *Journal of Symbolic Logic*, **77**, 669–686, 2012.

[5] W. Bibel. Matrices with connections. *Journal of the ACM*, **28**, 633–645, 1981.

[6] K. Chaudhuri. Focusing strategies in the sequent calculus of synthetic connectives. In *LPAR: International Conference on Logic, Programming, Artificial Intelligence and Reasoning*, vol. 5330 of *LNCS*, I. Cervesato, H. Veith, and A. Voronkov, eds, pp. 467–481. Springer, 2008.

[7] K. Chaudhuri, D. Miller, and A. Saurin. Canonical sequent proofs via multi-focusing. In *Fifth International Conference on Theoretical Computer Science*, vol. 273 of *IFIP*, G. Ausiello, J. Karhumäki, G. Mauri, and L. Ong, eds, pp. 383–396. Springer, 2008.

[8] V. Danos, J.-B. Joinet, and H. Schellinx. A new deconstructive logic: linear logic. *Journal of Symbolic Logic*, **62**, 755–807, 1997.

[9] O. Delande and D. Miller. A neutral approach to proof and refutation in MALL. In *23th Symp. on Logic in Computer Science*, F. Pfenning, ed, pp. 498–508. IEEE Computer Society Press, 2008.

[10] A. Felty. Transforming specifications in a dependent-type lambda calculus to specifications in an intuitionistic logic. In *Logical Frameworks*. G. Huet and G. D. Plotkin, eds, Cambridge University Press, 1991.

[11] G. Gentzen. Investigations into logical deduction. In *The Collected Papers of Gerhard Gentzen*, M. E. Szabo, ed., pp. 68–131. North-Holland, 1969. Translation of articles that appeared in 1934-35.

[12] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, **50**, 1–102, 1987.

[13] J.-Y. Girard. A new constructive logic: classical logic. *Mathematical Structures in Computer Science*, **1**, 255–296, 1991.

[14] A. Guglielmi, T. Gundersen, and M. Parigot. A proof calculus which reduces syntactic bureaucracy. In *Proceedings of the 21st International Conference on Rewriting Techniques and Applications (RTA 2010)*, vol. 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, C. Lynch, ed., pp. 135–150, Edinburgh, United Kingdom, July 2010. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[15] W. Heijltjes. Classical proof forestry. *Annals of Pure and Applied Logic*, **161**, 1346–1366, 2010.

[16] H. Herbelin and A. Saurin. λ-calculus and Λ-calculus: a capital difference. Unpublished manuscript, 2010.

[17] J. Herbrand. *Recherches sur la Théorie de la Démonstration*. PhD Thesis, University of Paris, 1930.

[18] S. Hetzl. Applying tree languages in proof theory. In *Language and Automata Theory and Applications (LATA) 2012*, vol. 7183 of *LNCS*, A.-H. Dediu and C. Martín-Vide, eds, pp. 301–312. Springer, 2012.

[19] S. Hetzl. The computational content of arithmetical proofs. *Notre Dame Journal of Formal Logic*, **53**, 289–296, 2012.

[20] S. Hetzl and L. Straßburger. Herbrand-confluence for cut-elimination in classical first-order logic. In *Computer Science Logic (CSL) 2012*, vol. 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, P. Cégielski and A. Durand, eds, pp. 320–334. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

[21] J. M. Howe. *Proof Search Issues in Some Non-Classical Logics*. PhD Thesis, University of St Andrews, Dec. 1998. Available as University of St Andrews Research Report CS/99/1.

[22] D. J. D. Hughes. Proofs without syntax. *Annals of Mathematics*, **143**, 1065–1076, 2006.

[23] F. Lamarche and L. Straßburger. Naming proofs in classical propositional logic. In *Typed Lambda Calculi and Applications, TLCA 2005*, vol. 3461 of *LNCS*, P. Urzyczyn, ed., pp. 246–261. Springer, 2005.

[24] F. Lamarche and L. Straßburger. From proof nets to the free *-autonomous category. *Logical Methods in Computer Science*, **2**, 1–44, 2006.

[25] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986.

[26] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoretical Computer Science*, **410**, 4747–4768, 2009.

[27] R. McKinley. Expansion nets: proof-nets for propositional classical logic. In *LPAR: Logic for Programming, Artificial Intelligence, and Reasoning*, vol. 6397 of *LNCS*, C. G. Fermüller and A. Voronkov, eds, pp. 535–549, Yogyakarta, Indonesia, Oct. 2010. Springer, 2010.

[28] R. McKinley. Proof nets for Herbrand's theorem. *ACM Transactions on Computational Logic*, **14**, 2013. To appear.

[29] D. Miller. A compact representation of proofs. *Studia Logica*, **46**, 347–370, 1987.

[30] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, **51**, 125–157, 1991.

[31] D. Miller and A. Saurin. From proofs to focused proofs: a modular proof of focalization in linear logic. In *CSL 2007: Computer Science Logic*, vol. 4646 of *LNCS*, J. Duparc and T. A. Henzinger, eds, pp. 405–419. Springer, 2007.

[32] S. Mimram. Focusing in asynchronous games. In *CiE: Programs, Proofs, Processes*, vol. 6158 of *LNCS*, F. Ferreira, B. Löwe, E. Mayordomo, and L. M. Gomes, eds, pp. 331–341. Springer, 2010.

[33] N. Novaković. *Sémantique Algébrique des Ressources pour la Logique Classique*. PhD Thesis, Institut National Polytechnique de Lorraine, 2011.

[34] D. Prawitz. *Natural Deduction*. Almqvist & Wiksell, 1965.

[35] E. P. Robinson. Proof nets for classical logic. *Journal of Logic and Computation*, **13**, 777–797, 2003.

[36] A. Saurin. *Une étude logique du contrôle (appliquée à la programmation fonctionnelle et logique)*. PhD Thesis, Ecole Polytechnique, Sept. 2008.

[37] L. Straßburger. What is the problem with proof nets for classical logic? In *Programs, Proofs, Processes, 6th Conference on Computability in Europe (CiE 2010)*, vol. 6158 of *LNCS*, F. Ferreira, B. Löwe, E. Mayordomo, and L. M. Gomes, eds, pp. 406–416, Ponta Delgada, Azores, Portugal, June 2010. Springer.

[38] C. Urban. *Classical Logic and Computation*. PhD Thesis, University of Cambridge, 2000.

[39] C. Urban and G. M. Bierman. Strong normalisation of cut-elimination in classical logic. *Fundamenta Informaticae*, **45**, 123–155, 2001.

Received 21 December 2012