ELSEVIER Theoretical Computer Science 211 (1999) 129–188

Theoretical Computer Science

Fundamental Study

Representing scope in intuitionistic deductions

Matthew Stone [*]

Department of Computer and Information Science, University of Pennsylvania, 200 South 33rd St., Philadelphia PA 19104, USA

Received June 1996; revised August 1997
Communicated by J. Gallier

Abstract

Intuitionistic proofs can be segmented into *scopes* which describe when assumptions can be used. In standard descriptions of intuitionistic logic, these scopes occupy contiguous regions of proofs. This leads to an explosion in the search space for automated deduction, because of the difficulty of planning to apply a rule inside a particular scoped region of the proof. This paper investigates an alternative representation which assigns scope explicitly to formulas, and which is inspired in part by semantics-based translation methods for modal deduction. This calculus is simple and is justified by direct proof-theoretic arguments that transform proofs in the calculus so that scopes match standard descriptions. A Herbrand theorem, established straightforwardly, lifts this calculus to incorporate unification. The resulting system has no impermutabilities whatsoever – rules of inference may be used equivalently anywhere in the proof. Nevertheless, a natural specification describes how λ-terms are to be extracted from its deductions. © 1999—Elsevier Science B.V. All rights reserved

Contents

[*] E-mail: matthew@linc.cis.upenn.edu.

1. Background

This paper is an exploration of the relationship between scope, proof structure and proof search in intuitionistic logic. The links between these notions can be framed in an intuitive way.

1.1. Scope and structure

In intuitionistic proofs, information that is assumed as part of proving some statement can only be used in proving that statement. Intuitionistic proofs derive their discipline of scope from this constraint. The scope of an assumed formula identifies the statements to whose proof the assumption may contribute. The scope of an assumed value identifies the statements which may be instantiated to refer to this value. Conversely, the scope of a formula to be proved identifies the assumed values at which it may be instantiated and the assumed formulas that may contribute to its proof. Both the initial steps that link assumptions and conclusions and the logical rules that combine proofs must be formulated to enforce this discipline of scope.

The intuitionistic discipline of scope underlies the Curry–Howard isomorphism, which allows functional programs to be extracted from intuitionistic logic deductions [24]. Assumptions in intuitionistic proofs correspond to variables in functional programs. The fact that an assumption in a proof has a scope that determines where it may be used corresponds to the fact that a variable in a program has a syntactic scope in which it is bound.

The same constraint can be applied in logic programming to implement modules using intuitionistic implication and to create local variables using intuitionistic quantifiers [31]. The discipline of scope restricts the use of these assumed facts and values to the appropriate locality.

To exploit these features for practical program synthesis (as in [9, 30]) or logic programming (as in [34]), it is not enough merely to be able to infer automatically if a given formula is an intuitionistic theorem. These tasks call for the automatic derivation of intuitionistic proofs and the analysis of automatic intuitionistic proof search.

In intuitionistic proof theory, we are used to inference rules that enforce this discipline of scope by a discipline of structure. This is what happens for example in the usual sequent calculi for intuitionistic logic, such as that in Fig. 1. Each sequent

$$\Gamma, A \longrightarrow A \qquad\qquad \Gamma, \bot \longrightarrow A$$

$$\frac{\Gamma, A \wedge B, A, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\wedge\!\longrightarrow \qquad \frac{\Gamma \longrightarrow A \quad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\longrightarrow\!\wedge$$

$$\frac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}\longrightarrow\!\vee 1 \qquad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}\longrightarrow\!\vee 2$$

$$\frac{\Gamma, A \vee B, A \longrightarrow C \qquad \Gamma, A \vee B, B \longrightarrow C}{\Gamma, A \vee B \longrightarrow C}\vee\!\longrightarrow$$

$$\frac{\Gamma, A \supset B \longrightarrow A \qquad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\supset\!\longrightarrow \qquad \frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\longrightarrow\!\supset$$

$$\frac{\Gamma, \forall x A, A[t/x] \longrightarrow C}{\Gamma, \forall x A \longrightarrow C}\forall\!\longrightarrow \qquad \frac{\Gamma \longrightarrow A[a/x]}{\Gamma \longrightarrow \forall x A}\longrightarrow\!\forall\dagger$$

$$\frac{\Gamma, \exists x A, A[a/x] \longrightarrow C}{\Gamma, \exists x A \longrightarrow C}\exists\!\longrightarrow\dagger \qquad \frac{\Gamma \longrightarrow A[t/x]}{\Gamma \longrightarrow \exists x A}\longrightarrow\!\exists$$

Fig. 1. A cut-free sequent calculus for intuitionistic logic, LJ. † For ($\longrightarrow\forall$) and ($\exists\longrightarrow$), a must not appear in the conclusion.

contains on the right a single statement C to be proved and on the left a multiset of assumptions Γ that may be used to prove it. Transitions between scope arise when there is a difference in assumptions between the premise sequents of a rule and its result sequent. For example, the ($\longrightarrow\supset$) rule describes the derivation of $A \supset B$ from assumptions Γ in terms of a derivation with different assumptions:

$$\frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\longrightarrow\!\supset$$

In the premise of this inference rule, we have a new scope in which the assumption A may be used, as encoded by the addition of A on the left; but this new combination of assumptions is available only to prove B, as encoded by the lone B on the right. Other rules simply describe inferences that may be performed within particular scopes; the assumptions made available in the premise sequents are the same as those available for the result sequent. ($\longrightarrow\wedge$) is an example:

$$\frac{\Gamma \longrightarrow A \qquad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\longrightarrow\wedge$$

In this method of assigning scope, the position of a rule-application in a proof determines the scope of any new assumptions or new goals for proof that the rule-application introduces. Viewing the proof as a tree, the path from the root to the site of the rule-application contains blocks of inferences performed in a common scope punctuated by inferences that change scope. The sequence of rules that change scope on this path establishes the scope in force when the rule applies. We can therefore refer to systems like that of Fig. 1 as structurally scoped sequent calculi.

Sometimes, a rule application must take narrow scope in a proof, to respect the scope of assumptions in which it depends. When the proof is presented in a structurally scoped calculus, the rule application must appear above the rules that introduce that scope. For example, consider a proof of $(C \supset B \vee A) \supset (C \supset A \vee B)$; the disjunction $B \vee A$ depends on the assumption of C. The proof is shown in (1); ellipsis (...) in sequents indicates that certain assumed formulas have been suppressed for compactness and clarity.

$$\dfrac{\left\{\dfrac{\left\{\dots,C \longrightarrow C \qquad \dfrac{\dfrac{\dots,A \longrightarrow A}{\dots,A \longrightarrow A\lor B}{\longrightarrow\lor 1} \quad \dfrac{\dots,B \longrightarrow B}{\dots,B \longrightarrow A\lor B}{\longrightarrow\lor 2}}{\dots,B\lor A \longrightarrow A\lor B}{\lor\longrightarrow}\right.}{C\supset B\lor A, C \longrightarrow A\lor B}{\supset\longrightarrow}\right.}{C\supset B\lor A \longrightarrow C\supset A\lor B}{\longrightarrow\supset}$$
$$\dfrac{\qquad}{\longrightarrow (C\supset B\lor A)\supset(C\supset A\lor B)}{\longrightarrow\supset} \qquad (1)$$

We identify three scopes in this proof, corresponding to a sequent in which no assumptions are available (the root), a sequent in which the assumption of $C\supset B\lor A$ is available (inside the outer brace), and a sequent in which assumptions of C and $C\supset B\lor A$ are available (inside the inner brace). (In not regarding ($\lor\longrightarrow$) as a change of scope, we anticipate the results of Section 3.2.) In the innermost scope, the proof proceeds by case analysis, by applying ($\supset\longrightarrow$) and ($\lor\longrightarrow$) rules to the assumption $C\supset B\lor A$. We could attempt to apply these two rules in either of the scopes that make that assumption available, but no proof could be built if the rules were applied in the outer scope. In the outer scope, C is not available, so the leftmost subproof would involve the impossible goal of showing $C\supset B\lor A\longrightarrow C$.

In other cases, a rule application must take wide scope, because the indefinite information it encodes must be resolved before nested assumptions can be made. Again, in the structurally scoped calculus, this constrains the position in the proof tree at which the rule application occurs. For example, in proving $B\lor A\supset(C\supset A)\lor(C\supset B)$, the alternatives for the disjunction $B\lor A$ determine which implication should be proved, $C\supset A$ or $C\supset B$. The proof appears in (2).

$$\left\{\dfrac{\dfrac{\dfrac{\{B\lor A,B,C \longrightarrow B}{B\lor A,B \longrightarrow C\supset B}{\longrightarrow\supset}}{B\lor A,B \longrightarrow (C\supset A)\lor(C\supset B)}{\longrightarrow\lor 2} \quad \dfrac{\dfrac{\{B\lor A,A,C \longrightarrow A}{B\lor A,A \longrightarrow C\supset A}{\longrightarrow\supset}}{B\lor A,A \longrightarrow (C\supset A)\lor(C\supset B)}{\longrightarrow\lor 1}}{B\lor A \longrightarrow (C\supset A)\lor(C\supset B)}{\lor\longrightarrow}\right.$$
$$\dfrac{\qquad}{\longrightarrow B\lor A\supset(C\supset A)\lor(C\supset B)}{\longrightarrow\supset} \qquad (2)$$

We identify four scopes in this proof. There is the root where no assumptions are available, an inner scope in which $B\lor A$ is available, and two further scopes, where C is available, once in conjunction with the B case, and once with the A case. Consider the left C scope. Once C is assumed, the structural discipline of scope forces all subsequent reasoning to contribute to the proof of B. Now, we might have attempted to apply the ($\lor\longrightarrow$) rule here, instead of lower, since the assumption of $B\lor A$ remains available here. Since the disjunction $B\lor A$ must contribute not only to that proof but to the proof of $C\supset A$ (and A), the proof could not be completed in this case.

1.2. Search and structure

Sequent calculus can be seen as a method for formalizing the process of proof search in natural deduction. Natural deduction systems express the Curry–Howard isomorphism most concisely: natural deduction proofs correspond to λ-terms both in syntax and in normalization [38]. However, natural deduction raises difficulties for describing proof search strategies. Natural deduction involves two kinds of rules, introduction and elimination rules, that should be used in different circumstances in proof search. Elimination rules should be used to decompose assumptions, introduction rules to assemble conclusions. To use them otherwise requires the interpreter to guess a needed formula from among all possible formulas of the logic. This distinction is made explicit in sequent systems, which separate assumptions and conclusions on different sides of the sequent $\longrightarrow$, and use different rules to decompose the logical connectives on either side.

Sequent systems therefore provide a straightforward framework for proof search. In this framework, the structure of a proof corresponds to the order in which rules should be applied during proof search. The algorithm to search for proofs is simply to build sequent proofs from the root up, repeatedly extending an unfinished branch of the proof by applying a sequent rule that extends the branch. The choice of which rule to apply is nondeterministic; we might apply some finite lookahead to help identify rules that make progress toward completing the proof, but in general we must backtrack among alternative choices for extending the proof. This algorithm for sequent search serves as a jumping off point for further optimizations, including tableau [41] and matrix proof methods [3, 5].

1.3. Scope, search and structure: a conflict

Following these two intuitions, position in a sequent calculus proof identifies both the scope of a rule application and the time at which the rule application must be considered in proof search. The dual roles of position are in conflict. From the perspective of proof search, we would like to apply a rule only when we recognize that it is needed. However, as examples (1) and (2) show, we must consider applying a rule in each of the scopes possible for it. In the structurally scoped calculus, such scopes correspond to positions in the proof – positions that may represent earlier stages in proof search than the stage when the need for the rule is recognized.

For example, consider a variant of the theorem of (2):

$$B \vee A, B \wedge C \supset F, A \wedge C \supset E \longrightarrow (C \supset E) \vee (C \supset F)$$

Again, to prove this it is necessary to apply $(\vee \longrightarrow)$ at wide scope, before any assumption of C, because B and A contribute to the proofs of different implications. But now these contributions are indirect and can be identified only on the basis of a chain of inferences performed in the nested scope. For example, B combines with C to establish the conclusion F by $(\supset\longrightarrow)$. Recognizing such indirect connections can be as hard as constructing the proof itself. In first-order intuitionistic logic in particular, there is no bound on the length of inference chain in a nested scope that may be required to link the result of rule applied in a wide scope to a needed conclusion. In general then, automated methods must be prepared to apply a rule before they know whether the application will even be needed! The regime for imposing scope on proofs means that proofs can no longer be constructed in a goal-directed manner. This is a severe problem in practice, where policies for avoiding or guessing rule orderings in particular situations are typically required [45].

The difficulty is exacerbated because it is impossible in general to apply all possible rules in an outer scope before moving in to a nested scope. The decision to apply the rule to change scope must be undertaken when other possible inferences in the outer scope remain. Should proof search fail subsequent to this decision, we must reconsider applying some of these possible inferences. This means backtracking to a stage when the proof contained an open branch in this scope – so it means discarding (then perhaps repeating) all search attempted since changing scope.

2. Overview

Considerations of search invite us to decouple scope in intuitionistic proofs from position. We shall see in this paper how we can accomplish this by making the discipline of scope explicit, so that the scopes of terms, formulas, and rule applications are represented overtly by terms in the proof. We can refer to proof systems so obtained as explicitly scoped sequent calculi for intuitionistic logic.

By allowing a rule to be applied in a given scope at any point in proof search, explicitly scoped calculi eliminate the difficulties observed above. Explicit scoping allows rules at wide-scope to be selected locally on the basis of an immediate contribution to the proof (possibly at nested scope) and to be added to the proof without revising deductions at nested scope that have been performed already. However, *despite the potentially unorthodox order in which they are built, explicitly scoped proofs correspond directly to ordinary natural deduction proofs.* Explicitly scoped sequent calculi therefore offer a framework for automatically deriving λ-terms and for regulating the combination of modular information in logic programming proof search.

The central results of this paper substantiate these observations. First, in Section 3, we present simple and direct arguments based on permutabilities of inferences that establish a constructive correspondence between proofs in an explicitly scoped system and proofs in the structurally scoped one. We obtain a lifted version of this calculus in Section 5.1 using a standard construction [29]; unification in the lifted calculus constrains the scope of rule applications dynamically in the course of proof search. Then, we outline in Section 5.2, how λ-terms can be extracted from the lifted, explicitly scoped sequent deductions by adapting the techniques proposed by [15].

These results strengthen existing semantic techniques by giving them a constructive, proof-theoretic foundation. In this overview, we motivate the distinction in two ways: by contrasting the intuitions underlying semantic and proof-theoretic derivations of explicitly scoped calculi, and by reviewing a parallel distinction in first-order classical logic between Herbrand's theorem and the Skolem–Herbrand–Gödel theorem.

We have found this alternative syntactic representation of scope extremely useful in practice, in part because of the further results it enables and for its applications to logic programming and program synthesis. For example, invariants of the sequent calculus can be used to devise efficient algorithms for constraining scopes [43]. Moreover, new logical fragments can be shown to have uniform proofs in this system, giving a logical and syntactic characterization of logic programming languages with modules and indefinite information; see Section 5.1. (Further work along these lines is currently in progress.)

2.1. Semantic vs. proof-theoretic intuitions

Any semantics for intuitionistic logic allows us to reason classically about intuitionistic provability, as follows. The semantics specifies a class of models, where each model contains some set of points at which objects exist and at which relations hold. The semantics also describes how formulas are evaluated for truth and falsehood with respect to these points according to compositional rules that can be expressed using classical formulas. (For a survey of intuitionistic semantics see [46, ch. 2; 47, Ch. 13].)

Given the semantics, a demonstration that a formula is valid can proceed by translation: each formula is labeled with a term that represents its point of evaluation and is decomposed in keeping with the rules for its semantic evaluation according to the inference rules of classical logic. In classical logic, propositional rules can be applied in any order with the same effect. Thus by labeling formulas and using classical inferences, we can eliminate the interdependence of scope and structure that complicates proof search for structurally scoped systems like LJ.

In particular, in a Kripke model for intuitionistic logic [28], the points of evaluation are called possible worlds. Conjunction, disjunction and existential quantification are interpreted classically at the world of evaluation. Implication and universal quantification must hold not only at the world of evaluation but for all worlds accessible from the world of evaluation under a transitive, reflexive accessibility relation.

In Kripke semantics, each world can be identified by a sequence of transitions of accessibility required to reach that world. This suggests using terms representing such sequences as labels in translation theorem proving. In the earliest such system, Fitting represents transitions of accessibility as integers and paths of accessibility as integer strings [16, 17]. Smullyan uses variables instead of integers to represent transitions and thus obtains a closer correspondence with classical deduction [42]. Wallen shows how theorem-proving techniques for classical logic, such as matrix proof methods [3, 5] and structure sharing [8], can be applied to Smullyan's system [50]. After further study, these systems can now be regarded as instances of more general techniques of semantics-based translation [35] and labeled deductive systems [11].

These works provide new inference systems and semantic demonstrations that these systems allow the same theorems to be proved as a structurally scoped intuitionistic sequent calculus. But they leave open the question of how to extract a proof in the structurally scoped system from a theorem derived in the new system. One aspect of this problem has been addressed independently in [51], which describes an algorithm for unfolding the matrix proofs of [50] directly into more standard sequent proofs. This procedure is outlined informally with pseudo-code in [52] and extended to related logics in [53]. Indeed, the standard interpretation of $\bot$, as a formula true in no world of any Kripke model, could make this extraction genuinely problematic (see Section 3.3.1).

Our results provide constructive correspondences between path-based explicitly scoped proofs and structurally scoped proofs. In addressing this question, the present result provides a proof-theoretic strengthening of existing work. In fact, together with the soundness and completeness theorems for classical logic, the present result constitutes an alternative demonstration of the soundness and completeness of LJ proofs under a variant of Kripke semantics: the *fallible* semantics, where $\bot$ may be true at selective worlds in a model, provided all other atomic formulas are true there [48].

The strengthening of results corresponds to a strengthening in the interpretation of the labels of formulas. In path-based translation, the elements of a term μ correspond to transitions between possible worlds, and the association between a term μ and a statement p means that p is to be evaluated in the semantics at a world represented by μ. In virtue of our new constructive correspondence, the elements of a term μ correspond directly to rule applications that effect a change of scope in structurally scoped proof. As a sequence, μ describes the sequence of scope-changing rule applications that apply along some path in the proof tree. Labeling p with μ indicates that the rule that introduces p should appear in the structurally scoped proof at the scoped location identified by μ. By rearranging a labeled proof so that the position of inferences in the tree matches the positions named by their labels, we can transform an explicitly scoped proof into a structurally scoped proof. Thus, instead of appealing to semantic intuitions, we can see that the labeling terms are a purely syntactic notation that allows intuitionistic deductions to be built in an incremental way.

2.2. Herbrand's theorem as a parallel

A parallel with these results can be found in Herbrand's theorem for classical logic, which offers a device for reasoning about the right scope for applications of quantifier rules independent of the order in which those rules appear in the proof. Lincoln and Shankar [29] offer a demonstration of the generality of this way of looking at Herbrand's theorem.

The problem is similar to the one just described: The quantifier rules in Fig. 1 involve a link between structure and scope in deductions. The $(\longrightarrow\forall)$ and $(\exists\longrightarrow)$ rules impose a requirement that the eigenvariable substituted for the bound variable must not occur elsewhere in the sequent to which the rule applies:

$$\frac{\Gamma, \exists x A, A[a/x] \longrightarrow C}{\Gamma, \exists x A \longrightarrow C}\,\exists\longrightarrow.$$

Because of this condition, we may regard the subproof above the rule as representing the scope of the eigenvariable. Above this rule, the other quantifier rules ($\forall \rightarrow$) and ($\rightarrow \exists$) may perform substitutions into formulas so as to include the new variable a. Such rules then cannot be permuted below the introduction of a, without violating the eigenvariable condition.

As extended in [29, 40], Herbrand's theorem describes alternative sequent rules for quantifiers that substitute complex terms for bound variables instead of eigenvariables. These terms, called Skolem or Herbrand terms, are representations of eigenvariables; they have the form $f(t_1, \ldots, t_n)$ where f is a symbol associated with an occurrence of a quantifier and $t_1, \ldots, t_n$ is a sequence of terms specified by the logic. This sequence is designed to include as subterms representations of all the eigenvariables that would have to appear in the sequent when the quantifier rule applied – no matter what rearrangements to the proof were performed. In classical logic, the only obstacle to such rearrangement is the impossibility of applying a rule to a subformula lower than any inference involving the formula that contains it. The sequence $t_1, \ldots, t_n$ therefore lists the instantiations made as part of deriving the quantified formula to which the rule applies. In structurally scoped calculi for intuitionistic and linear logic, additional terms are required to reflect the different scopes of quantifiers at different positions in the proof.

The structure of Herbrand terms induces a partial ordering on rules. If a variable is instantiated to t at rule L and a variable is instantiated to a term that properly contains t at rule H, then L should occur lower than H. And among rules that instantiate a variable with a term t, the rule that constructs t as a representation of an eigenvariable should occur lowest. The proof of the theorem shows how to rearrange the proof so that the position of quantifier rules respects this ordering. At this point, the Herbrand terms can be replaced by variables that satisfy eigenvariable conditions when necessary.

This result links instantiations of variables and the possibility of reordering inferences: it chooses instantiations in a way that eliminates the need for backtracking among orderings of quantifier inferences. It therefore offers a greater potential for cutting down proof search than a result describing correct instantiations to skeletons of proofs in which the order of rules is fixed, as in [49]. Herbrand's theorem can also be contrasted with a weaker result, the Skolem–Herbrand–Gödel theorem. The latter theorem shows how a problem of first-order provability can be related to a set of problems of propositional provability by using instantiations with functional terms. This theorem has been extended beyond classical logic in [18], using an extension to the logical syntax. Again, while this result offers a way to derive theorems, it does not directly offer a way to obtain proofs.

3. Scoping by position: A sequent presentation

This section describes and verifies a first explicitly scoped sequent calculus for intuitionistic logic. The calculus is based on two intuitions: first, that the scope transitions of structurally scoped intuitionistic proofs are associated precisely with the rules for implication and universal quantification; and second, that the scope of a rule application is given by the sequence of scope-changing rules that occur on the path from the root of the proof to the site of the rule application. Our strategy is simply to name each rule application that creates a scope, using a fresh variable (following to the first intuition), and to label each formula in the proof with a string of names recording the scope of the rule that introduced it (following the second).

For example, suppose the $(\longrightarrow\supset)$ rule applies to a formula $A \supset B$ labeled by μ. In the structurally scoped proof, this rule makes a transition from the scope where $A \supset B$ is introduced, namely μ, the scope where to a new, nested scope, which we name $\mu\alpha$. So the rule makes available an assumption of A labeled $\mu\alpha$ and a conclusion of B, also labeled $\mu\alpha$. Because scope in this rule is made explicit, there is no need for a further structural mechanism to enforce the intuitionistic connection of A and B.

Dually, suppose the $(\supset\longrightarrow)$ rule applies to a formula $A \supset B$ labeled by μ. The assumption of $A \supset B$ is made at a scope named by μ, but this assumption once made can persist into nested scopes. So if we derive A with a label ν that has μ as a prefix – corresponding in any scope nested within μ – we can conclude B in scope ν. For similar reasons, at leaves of the proof, where we match an assumption with an identical conclusion, the label of the assumption must be a prefix of the label of the conclusion.

The structure of the section is as follows. In Section 3.1, we make some observations about the sequent calculi we will be studying. We adopt a treatment of structure in sequents that makes permutation of inference particularly easy to describe. Then, in Section 3.2, we substantiate the claim that the scopes of intuitionistic logic proofs are associated with exactly the rules for implication and universal quantifiers. The section introduces a sequent calculus in which the right of a sequent is a multiset of formulas and the sequent rules for disjunction and existential quantifiers match those for classical logic. The new calculus offers a simple opportunity to introduce the technique of using permutations of inference to establish the correspondence between proofs in different systems.

Section 3.3 formally describes this explicitly scoped sequent calculi for intuitionistic logic. Section 3.4 gives transformations between proofs in this system and proofs in the structurally scoped system. These transformations exploit the intuitions behind the annotations in a straightforward way. In particular, to transform a structurally scoped proof to an explicitly scoped proof, we simply label the occurrences of formulas in it according to the scoped positions specified by the structure of the proof. Meanwhile, to transform an explicitly scoped proof into a structurally scoped proof, we rearrange the inferences in the explicitly scoped proof so that the positions of rule applications match the labels of their formulas.

3.1. Preliminaries

In the system of Fig. 1, a sequent is written $\Gamma \longrightarrow A$, where Γ is a finite multiset of formulas and A is a single formula. A *derivation* is a tree of sequents derived from initial (or axiom) sequents according to the rules of Fig. 1. We will also call derivations *proofs*, provided no confusion with meta-level argumentation about derivations (another kind of proof) might result. The root of a derivation is called its *end sequent*. If a rule applies to a formula A occurrence in the end sequent of derivation $\mathscr{D}$, we call A the *principal formula* of the rule application, and we call the designated occurrences of the immediate subformulas of A in the immediate subderivations of $\mathscr{D}$ the *side formulas* of the rule application. For quantifier rules, the variable a introduced is the *eigenvariable* of the rule.

The sequent calculus of Fig. 1 is given as *G3a* in [27] and as $\mathscr{GK}_i^{\supset,\wedge,\vee,\forall,\exists,\bot}$ in [20]. It reflects a particular approach to the treatment of structure in sequents, which we will adopt throughout this paper. The calculus dispenses with the rule of *contraction*:

$$\frac{\Gamma,A,A \longrightarrow \Delta}{\Gamma,A \longrightarrow \Delta}\,C$$

in favor of the preservation of principal formulas of rules in subderivations. This automatic duplication of formulas streamlines and localizes the representation of reuse of premises (without this, in converting sequent proofs to natural deductions, intermediate results will first be weakened, then contracted); and in fact, duplication falls out of a natural logical specification of the sequent system (as in [15]). The calculus also dispenses with *weakening*:

$$\frac{\Gamma \longrightarrow \Delta}{\Gamma,A \longrightarrow \Delta}\,W$$

because the axiom rule allows any finite multiset of formulas on the left in addition to the formula that agrees. Instead of repeatedly weakening the end sequent of a derivation $\mathscr{D}$ by formulas Λ using a structural rule, we can define a derivation $\Lambda + \mathscr{D}$ obtained from $\mathscr{D}$ by replacing the left multiset Γ of every sequent in $\mathscr{D}$ by the multiset union of Λ and Γ. As long as no eigenvariable of $\mathscr{D}$ occurs free in Λ, $\Lambda + \mathscr{D}$ is also a correct derivation.

Because the sequent calculus avoids structural rules, it is simple to describe interchanges of logical rules in this calculus. Adjacent logical inferences are applied in succession, so that a higher inference H is applied at the root of the immediate subderivation of a lower inference L. (No structural rules intervene.) If a side formula of L is not the principal formula of H, we may attempt to replace the derivation of the end sequent of L by a new derivation of the same end sequent with H at the root, followed immediately by L, capped by subderivations copied from the original derivation (but possibly weakened). Performing such a replacement constitutes an interchange of rules L and H and demonstrates the permutability of L and H; see [26]. Such replacement is not always possible because of structural conditions L and H impose; in that case the inferences are impermutable.

Since structure is treated implicitly, it is also possible to think of the left of a sequent as a set rather than a multiset of formulas. While simpler now, a set-based formulation complicates the translation from sequent proofs to natural deduction proofs, because

$$\Gamma, A \longrightarrow A, \Delta$$

$$\frac{\Gamma, A\wedge B, A, B \longrightarrow \Delta}{\Gamma, A\wedge B \longrightarrow \Delta}\wedge\!\rightarrow \qquad \frac{\Gamma \longrightarrow A, A\wedge B, \Delta \qquad \Gamma \longrightarrow B, A\wedge B, \Delta}{\Gamma \longrightarrow A\wedge B, \Delta}\rightarrow\!\wedge$$

$$\frac{\Gamma \longrightarrow A, B, A\vee B, \Delta}{\Gamma \longrightarrow A\vee B, \Delta}\rightarrow\!\vee \qquad \frac{\Gamma, A\vee B, A \longrightarrow \Delta \qquad \Gamma, A\vee B, B \longrightarrow \Delta}{\Gamma, A\vee B \longrightarrow \Delta}\vee\!\rightarrow$$

$$\frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A\supset B, \Delta}\rightarrow\!\supset \qquad \frac{\Gamma, A\supset B \longrightarrow A, \Delta \qquad \Gamma, A\supset B, B \longrightarrow \Delta}{\Gamma, A\supset B \longrightarrow \Delta}\supset\!\rightarrow$$

$$\frac{\Gamma, \forall x A, A[t/x] \longrightarrow \Delta}{\Gamma, \forall x A \longrightarrow \Delta}\forall\!\rightarrow \qquad \frac{\Gamma \longrightarrow A[a/x]}{\Gamma \longrightarrow \forall x A, \Delta}\rightarrow\!\forall\dagger$$

$$\frac{\Gamma, \exists x A, A[a/x] \longrightarrow \Delta}{\Gamma, \exists x A \longrightarrow \Delta}\exists\!\rightarrow\dagger \qquad \frac{\Gamma \longrightarrow A[t/x], \exists x A, \Delta}{\Gamma \longrightarrow \exists x A, \Delta}\rightarrow\!\exists$$

Fig. 2. A cut free sequent calculus for minimal logic, LMM. † For $(\rightarrow\forall)$ and $(\exists\rightarrow)$, a must not appear in the conclusion.

the translation calls for several occurrences of the same formula to appear, labeled with distinct proof terms.

3.2. Refining the structural discipline of scope

In justifying explicitly scoped calculi, we will use not LJ but a somewhat less familiar sequent system, LMM, given in Fig. 2. The use of LMM instead of LJ is a convenience which makes transparent the way that scoped regions are created in intuitionistic sequent proofs, exactly at $(\rightarrow\supset)$ and $(\rightarrow\forall)$ rules. This transparency makes the correctness of the system presented in Fig. 2 easier to see and to show.

LMM is a sequent calculus presentation of minimal logic – the fragment of intuitionistic logic without negation or an absurdity rule.[1] Following [16], this formalization localizes the specifically intuitionistic character of the system in the $(\rightarrow\supset)$ and $(\rightarrow\forall)$ rules. (The particular presentation above restricts the system $\mathcal{GKT}_i^{\supset,\wedge,\vee,\forall,\exists,\perp}$ of [20] to minimal logic.) In LMM, unlike in LJ, the right of a sequent is a multiset of formulas, and the same structural conventions apply on the right and the left. In particular, as before, weakening is built into initial sequents, while contraction is built into inference figures.

LMM exploits these multiple conclusions to give the same sequent rules for most connectives that the connectives have in classical logic. (The sequent calculus for classical logic, LK, also has multiple formulas on the right in sequents.) For example, instead of LJ's two right rules for disjunction, LMM has a single $(\rightarrow\vee)$ rule:

$$\frac{\Gamma \longrightarrow A, B, A\vee B, \Delta}{\Gamma \longrightarrow A\vee B, \Delta}\rightarrow\vee$$

[1] Minimal logic shares with intuitionistic logic the properties of scope relevant for logic programming and program synthesis. Intuitionistic negation can be simulated in minimal logic by translating all goal subformulas A into $A\vee\perp$. In fact, as discussed in more detail in Section 3.3.1, other treatments of intuitionistic negation are problematic for the representation of scope in proofs – in translation methods, these treatments may open up the possibility of reasoning about intuitionistic semantics in an essentially classical way.

This rule leaves both disjuncts available and thereby allows the choice of which disjunct is to be proved to be delayed.

The difference with classical sequent calculus lies in the $(\longrightarrow\supset)$ and $(\longrightarrow\forall)$ rules, which apply to deductions where only their side formula appears on the right in the end sequent, for example:

$$\frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B, \Delta} \longrightarrow\supset$$

These are the rules where assumptions are made; discarding the right formulas isolates the subderivation in which the assumption is used, and thereby ensures that the use of any assumption respects its scope. Alternatively, this restriction introduces impermutabilities into the logic by eliminating the possibility of delaying the resolution of disjunctive possibilities until above these rules. Note that since formulas are discarded, to construct a derivation from $\mathscr{D}$ that weakens by A on the right (written $\mathscr{D} + A$), we must add A to the right of just those sequents in $\mathscr{D}$ that *do not lie above* an application of $(\longrightarrow \supset)$ or $(\longrightarrow\forall)$.

The correctness of LMM is typically shown by a simple argument that shows how LMM proofs can be recursively translated to proofs in LJ in which cuts may appear. Then, the cut-elimination theorem can be used to reduce these proofs to cut-free proofs (cf. [20]). We can also show the correctness of LMM by permuting inferences.

Lemma 1. *Every LMM proof $\mathscr{D}$ with end-sequent $\Gamma \longrightarrow A$ (for a single right formula A) can be transformed into an LJ proof by permuting inferences and then "cleaning up" the right sequents.*

Proof. First we describe the cleaning up. Let $\mathscr{D}'$ be an LMM proof, and suppose that every subproof with end sequent $\Gamma \longrightarrow \Delta$ that ends in a left rule involves a singleton Δ. Then $\mathscr{D}'$ can be transformed to LJ by the following translation.

At the axiom, we translate $\Gamma, A \longrightarrow A, \Delta$ by $\Gamma, A \longrightarrow A$. Now, consider a subproof ending with an inference R of $(\longrightarrow\wedge)$, $(\longrightarrow \vee)$, $(\longrightarrow\exists)$ and $(\supset\longrightarrow)$. Each immediate subderivation ends in $\Gamma \longrightarrow \Delta$, and translation gives an LJ proof with end-sequent $\Gamma \longrightarrow A$ for some $A \in \Delta$. If A is not a side formula of R – in any subderivation for right rules, or in the left subderivation for $(\supset\longrightarrow)$, which establishes the antecedent – omit R from the translation: The translation of the subderivation is the needed result. Otherwise construct the translation of the derivation by applying R to the translated subderivations, replacing occurrences of $(\longrightarrow\vee)$ with either $(\longrightarrow\vee 1)$ or $(\longrightarrow\vee 2)$ as appropriate.

Proofs ending in any of the remaining rules are composed of immediate subderivations $\Gamma \longrightarrow A$ – where the subderivations derive a single formula on the left (common to both subderivations, if applicable). Thus, we can obtain an overall translation by translating these subderivations and applying the corresponding LJ rule to the results.

Now, suppose inferences are ordered in $\mathscr{D}'$ in the following way. If a left rule occurs immediately above a right rule R, R it is either $(\longrightarrow\supset)$ or $(\longrightarrow\forall)$; and if a left rule L occurs immediately above a $(\supset\longrightarrow)$ inference R, L is in the right subderivation of R.

Given this ordering, if the end sequent of $\mathscr{D}'$ has a singleton on the right (or ends in a right rule), then every subderivation of $\mathscr{D}'$ that ends by deriving $\Gamma \longrightarrow \Delta$ by a left rule, has a singleton Δ.

We permute the inferences in $\mathscr{D}$ so that they are ordered this way in two steps. First, we reorder left rules that occur above problematic right inferences; then we reorder left rules that occur on the wrong side above ($\supset\rightarrow$) inferences.

In the first step, we observe that inferences other than ($\rightarrow\supset$) and ($\rightarrow\forall$) fall into connected blocks in $\mathscr{D}$ in which ($\rightarrow\supset$) and ($\rightarrow\forall$) do not occur. Whenever a left inference L occurs above a right inference R within a common block, there are no obstacles to interchanging the inferences, cf. [26]. Observe that the principal formula of L cannot be a side formula of R: otherwise R is ($\rightarrow\supset$) and L and R are not in a common block. Moreover, R does not impose a novelty condition on an eigenvariable substituted at L. Otherwise R is ($\rightarrow\forall$) and L and R are not in a common block. The interchanged derivations are constructed in one of four patterns, depending on the number of premises of R and L. We exemplify each pattern.

As an example where R and L each have one premise, we have the transformation below.

$$\dfrac{\dfrac{\Gamma, A\wedge B, A, B \longrightarrow C, D, C\vee D, \Delta}{\Gamma, A\wedge B \longrightarrow C, D, C\vee D, \Delta}\,\wedge\!\rightarrow}{\Gamma, A\wedge B \longrightarrow C\vee D, \Delta}\,\rightarrow\!\vee \quad\Rightarrow\quad \dfrac{\dfrac{\Gamma, A\wedge B, A, B \longrightarrow C, D, C\vee D, \Delta}{\Gamma, A\wedge B, A, B \longrightarrow C\vee D, \Delta}\,\rightarrow\!\vee}{\Gamma, A\wedge B \longrightarrow C\vee D, \Delta}\,\wedge\!\rightarrow$$

A case where L has two premises and R one is this:

$$\dfrac{\dfrac{\Gamma, A\vee B, A \longrightarrow C, D, C\vee D, \Delta \quad \Gamma, A\vee B, B \longrightarrow C, D, C\vee D, \Delta}{\Gamma, A\vee B \longrightarrow C, D, C\vee D, \Delta}\,\vee\!\rightarrow}{\Gamma, A\vee B \longrightarrow C\vee D, \Delta}\,\rightarrow\!\vee \quad\Rightarrow$$

$$\dfrac{\dfrac{\Gamma, A\vee B, A \longrightarrow C, D, C\vee D, \Delta}{\Gamma, A\vee B, A \longrightarrow C\vee D, \Delta}\,\rightarrow\!\vee \quad \dfrac{\Gamma, A\vee B, B \longrightarrow C, D, C\vee D, \Delta}{\Gamma, A\vee B, B \longrightarrow C\vee D, \Delta}\,\rightarrow\!\vee}{\Gamma, A\vee B \longrightarrow C\vee D, \Delta}\,\vee\!\rightarrow$$

The other cases require weakening of derivations. If L has one premise and R two, we have for example:

$$\dfrac{\dfrac{\Gamma, A\wedge B, A, B \longrightarrow C, C\wedge D, \Delta}{\Gamma, A\wedge B \longrightarrow C, C\wedge D, \Delta}\,\wedge\!\rightarrow \quad \begin{array}{c}\mathscr{D}\\ \Gamma, A\wedge B \longrightarrow D, C\wedge D, \Delta\end{array}}{\Gamma, A\wedge B \longrightarrow C\wedge D, \Delta}\,\rightarrow\!\wedge \quad\Rightarrow$$

$$\dfrac{\dfrac{\Gamma, A\wedge B, A, B \longrightarrow C, C\wedge D, \Delta \quad \begin{array}{c}A, B + \mathscr{D}\\ \Gamma, A\wedge B, A, B \longrightarrow D, C\wedge D, \Delta\end{array}}{\Gamma, A\wedge B, A, B \longrightarrow C\wedge D, \Delta}\,\rightarrow\!\wedge}{\Gamma, A\wedge B \longrightarrow C\wedge D, \Delta}\,\wedge\!\rightarrow$$

Finally, if L and R both have two premises, we have for example:

$$\dfrac{\dfrac{\Gamma,A\vee B,A\longrightarrow C,C\wedge D,\Delta \qquad \Gamma,A\vee B,B\longrightarrow C,C\wedge D,\Delta}{\Gamma,A\vee B\longrightarrow C,C\wedge D,\Delta}\vee\!\longrightarrow \qquad \dfrac{\mathscr{D}}{\Gamma,A\vee B\longrightarrow D,C\wedge D,\Delta}}{\Gamma,A\vee B\longrightarrow C\wedge D,\Delta}\longrightarrow\!\wedge \tag{3}$$

We transform it to:

$$\dfrac{\mathscr{D}_1 \qquad \mathscr{D}_2}{\Gamma,A\vee B\longrightarrow C\wedge D,\Delta}\vee\!\longrightarrow$$

where

$$\mathscr{D}_1=\left\{\dfrac{\Gamma,A\vee B,A\longrightarrow C,C\wedge D,\Delta \qquad \dfrac{A+\mathscr{D}}{\Gamma,A\vee B,A\longrightarrow D,C\wedge D,\Delta}}{\Gamma,A\vee B,A\longrightarrow C\wedge D,\Delta}\longrightarrow\!\wedge\right.$$

$$\mathscr{D}_2=\left\{\dfrac{\Gamma,A\vee B,B\longrightarrow C,C\wedge D,\Delta \qquad \dfrac{B+\mathscr{D}}{\Gamma,A\vee B,B\longrightarrow D,C\wedge D,\Delta}}{\Gamma,A\vee B,B\longrightarrow C\wedge D,\Delta}\longrightarrow\!\wedge\right.$$

These permutations can be repeated so that in each block all applications of left rules appear closer to the root than any application of a right rule. Because of the possible duplication of subderivations at interchanges, we must perform the permutations in the right order to prove termination. As in [26], we induct on *degree*, the number of right rules with a left rule above them in the same block. We can always decrease the degree by one as follows: we find the highest such right rule R and permute it above all higher left rules. This sequence of permutations proceeds by induction on *grade*, the number of left rules above R in the same block. Find the lowest such left rule (it must be adjacent to R). Permute it down, decreasing the grade by one in each subderivation.

In the second step, we apply further permutations to this proof, so that whenever ($\supset\longrightarrow$) applies, it is never the case that a left rule is applied at the conclusion of the left subderivation. The structure of this argument is analogous to the previous one. Again, we can reduce by one the number of ($\supset\longrightarrow$) inferences in the proof with left rules concluding their left subderivation (*bad* ($\supset\longrightarrow$) inferences) by fixing any bad ($\supset\longrightarrow$) inference R which has no other bad ($\supset\longrightarrow$) inferences above it. The fix replaces the subderivation ending with the bad inference by another in which no inferences are bad. It is constructed by induction on the number of left rules applied consecutively above R. We reduce this by one at each step by permuting the lowest rule L down. L cannot apply to the side formula of R, which is on the right, nor will it introduce an eigenvariable violation, since R does not introduce an eigenvariable. So the only potentially problematic case is when L is also ($\supset\longrightarrow$): permuting L down cannot make L bad. But we know that L itself is not a bad inference, because R was chosen highest.

That means the left subderivation of L ends in an axiom or a right rule, and all higher left rules are good. We can therefore observe that the subderivation of L is equivalent to an LJ derivation $\mathcal{E}$ ending

$$\Gamma \longrightarrow C$$

We can adapt $\mathcal{E}$ to construct a new proof

$$\cfrac{\begin{array}{c}\mathcal{E}+\Delta\\ \Gamma \longrightarrow C,\Delta\end{array} \qquad \cfrac{\Gamma,D \longrightarrow A,\Delta \qquad \Gamma,D,B \longrightarrow \Delta}{\Gamma,D \longrightarrow \Delta}\,R}{\Gamma \longrightarrow \Delta}\,L$$

in which R appears only above the right subderivation of L.

In constructing this final derivation, we only interchange left rules (within common blocks), so we keep all the rules in good order. We thus obtain a permuted proof $\mathcal{D}'$ corresponding to an LJ proof. □

LMM eliminates impermutabilities associated with the requirement that $(\longrightarrow\vee)$ select one of the two disjuncts to be proven once and for all. For example, the LJ proof (4) requires the $(\vee\longrightarrow)$ rule to apply lower than the $(\longrightarrow\vee)$ rule.

$$\cfrac{\cfrac{B\vee A,B \longrightarrow B}{B\vee A,B \longrightarrow A\vee B}\,{\longrightarrow}\vee \qquad \cfrac{B\vee A,A \longrightarrow A}{B\vee A,A \longrightarrow A\vee B}\,{\longrightarrow}\vee}{B\vee A \longrightarrow A\vee B}\,\vee{\longrightarrow} \tag{4}$$

These rules may be applied in any order in LMM, as the derivation in (5) witnesses.

$$\cfrac{\cfrac{B\vee A,B \longrightarrow A,B \qquad B\vee A,A \longrightarrow A,B}{B\vee A \longrightarrow A,B}\,\vee{\longrightarrow}}{B\vee A \longrightarrow A\vee B}\,{\longrightarrow}\vee \tag{5}$$

However, not all impermutabilities of LJ are gone. Recasting the proof (2) in LMM yields the proof (6):

$$\cfrac{\left\{\cfrac{\cfrac{\cfrac{\{B\vee A,B,C \longrightarrow B}{B\vee A,B \longrightarrow C\supset A,C\supset B}\,{\longrightarrow}\supset \qquad \cfrac{\{B\vee A,A,C \longrightarrow A}{B\vee A,A \longrightarrow C\supset A,C\supset B}\,{\longrightarrow}\supset}{B\vee A \longrightarrow (C\supset A),(C\supset B)}\,\vee{\longrightarrow}}{B\vee A \longrightarrow (C\supset A)\vee(C\supset B)}\,{\longrightarrow}\vee}{}\right.}{\longrightarrow B\vee A\supset(C\supset A)\vee(C\supset B)}\,{\longrightarrow}\supset \tag{6}$$

Exploiting the new permutability, we can delay $(\vee\longrightarrow)$ until after $(\longrightarrow\vee)$. Nevertheless, because the $(\longrightarrow\supset)$ rules discard the alternative formulas on the right, it remains impossible to permute the application of $(\vee\longrightarrow)$ above the $(\longrightarrow\supset)$ rules in LMM.

These remaining LMM impermutabilities establish the scopes in the proof which isolate the consequences of assumptions. Since these scopes are an essential feature of intuitionistic logic, a different tack is required to devise a proof system without these impermutabilities. It is to this problem that we now turn.

3.3. Path annotations

Having isolated the intuitionistic discipline of scope in the structure of $(\longrightarrow\supset)$ and $(\longrightarrow\forall)$ rule, we will now make that discipline of scope explicit.

To represent scopes, we use strings built from a distinguished infinite alphabet of annotations to label terms and formulas. By convention, letters from the beginning of the Greek alphabet (α, β, etc.) represent eigenvariables that may appear in annotations; letters from the middle (μ, ν, etc.) represent strings of such eigenvariables.

We adapt sequents to describe the scope of first-order terms and formulas as follows. Each sequent has the form

$$\Sigma \triangleright \Gamma \longrightarrow \Delta$$

Each formula occurrence in Γ and Δ is labeled with an annotation term that specifies the scope of the formula. (These terms are written with superscripts.) The scope of first-order terms is specified by the indexing context Σ. Σ is a list of pairs $x:\mu$ assigning an annotation to each first-order variable that appears free in the sequent. The scope of a compound term is determined by this assignment to free variables:

Definition 1. t is a Σ-term of index μ if and only if for every free variable x that occurs in t, Σ assigns $x:\nu$ and ν is a prefix of μ.

By imposing appropriate manipulations to these annotations, we obtain a proof system that creates and matches scopes without discarding formulas from sequents. For example, $(\longrightarrow\supset)$ creates a new scope by introducing a new annotation variable α that cannot appear in the end sequent; the antecedent is made and the consequent derived in the new scope.

$$\frac{\Sigma \triangleright \Gamma, A^{\mu\alpha} \longrightarrow B^{\mu\alpha}, A \supset B^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow A \supset B^{\mu}, \Delta}\longrightarrow\supset$$

In contrast to LMM, the rule preserves all the right formulas from the end sequent in the subderivation. Likewise, $(\longrightarrow\forall)$ introduces a new scope by a transition α, assumes a new first-order eigenvariable a restricted to the new scope, and puts its side-formula there:

$$\frac{\Sigma, a:\mu\alpha \triangleright \Gamma \longrightarrow A[a/x]^{\mu\alpha}, \forall x A^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow \forall x A^{\mu}, \Delta}\longrightarrow\forall$$

The corresponding left rules offer the possibility of a change in scope. At $(\supset\longrightarrow)$, given the scope μ of the principal formula, we may consider deriving the antecedent and introducing the consequent at any longer string $\mu\nu$:

$$\frac{\Sigma \triangleright \Gamma, A \supset B^{\mu} \longrightarrow A^{\mu\nu}, \Delta \qquad \Sigma \triangleright \Gamma, A \supset B^{\mu}, B^{\mu\nu} \longrightarrow \Delta}{\Sigma \triangleright \Gamma, A \supset B^{\mu} \longrightarrow \Delta}\supset\longrightarrow$$

$$\Sigma \rhd \Gamma, A^{\mu} \longrightarrow A^{\mu\nu}, \Delta$$

$$\frac{\Sigma \rhd \Gamma, A \wedge B^{\mu}, A^{\mu}, B^{\mu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, A \wedge B^{\mu} \longrightarrow \Delta}\wedge\longrightarrow$$

$$\frac{\Sigma \rhd \Gamma \longrightarrow A^{\mu}, A \wedge B^{\mu}, \Delta \qquad \Sigma \rhd \Gamma \longrightarrow B^{\mu}, A \wedge B^{\mu}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A \wedge B^{\mu}, \Delta}\longrightarrow\wedge$$

$$\frac{\Sigma \rhd \Gamma, A \vee B^{\mu}, A^{\mu} \longrightarrow \Delta \qquad \Sigma \rhd \Gamma, A \vee B^{\mu}, B^{\mu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, A \vee B^{\mu} \longrightarrow \Delta}\vee\longrightarrow$$

$$\frac{\Sigma \rhd \Gamma \longrightarrow A^{\mu}, B^{\mu}, A \vee B^{\mu}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A \vee B^{\mu}, \Delta}\longrightarrow\vee$$

$$\frac{\Sigma \rhd \Gamma, A \supset B^{\mu} \longrightarrow A^{\mu\nu}, \Delta \qquad \Sigma \rhd \Gamma, A \supset B^{\mu}, B^{\mu\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, A \supset B^{\mu} \longrightarrow \Delta}\supset\longrightarrow$$

$$\frac{\Sigma \rhd \Gamma, A^{\mu\alpha} \longrightarrow B^{\mu\alpha}, A \supset B^{\mu}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A \supset B^{\mu}, \Delta}\longrightarrow\supset\dagger$$

$$\frac{\Sigma \rhd \Gamma, \forall x A^{\mu}, A[t/x]^{\mu\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, \forall x A^{\mu} \longrightarrow \Delta}\forall\longrightarrow\ddagger$$

$$\frac{\Sigma, a : \mu\alpha \rhd \Gamma \longrightarrow A[a/x]^{\mu\alpha}, \forall x A^{\mu}, \Delta}{\Sigma \rhd \Gamma \longrightarrow \forall x A^{\mu}, \Delta}\longrightarrow\forall\dagger$$

$$\frac{\Sigma, a : \mu \rhd \Gamma, \exists x A^{\mu}, A[a/x]^{\mu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, \exists x A^{\mu} \longrightarrow \Delta}\exists\longrightarrow\dagger$$

$$\frac{\Sigma \rhd \Gamma \longrightarrow A[t/x]^{\mu}, \exists x A^{\mu}, \Delta}{\Sigma \rhd \Gamma \longrightarrow \exists x A^{\mu}, \Delta}\longrightarrow\exists\ddagger$$

Fig. 3. Explicitly scoped, cut-free sequent calculus for minimal logic, LMP. † For ($\longrightarrow\forall$), ($\exists\longrightarrow$) and ($\longrightarrow\supset$), a and α must not appear in the conclusion. ‡ For ($\forall\longrightarrow$), there is a proviso that t be a Σ-term of index $\mu\nu$; for ($\longrightarrow\exists$) that t be a Σ-term of index μ.

Recall that this is in keeping with the structural discipline on the left in LJ and LMM, which preserve formulas on the left across transitions into nested scopes. For ($\forall\longrightarrow$), the transition to a nested scope may make available the first-order term which instantiates the bound variable.

$$\frac{\Sigma \rhd \Gamma, \forall x A^{\mu}, A[t/x]^{\mu\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, \forall x A^{\mu} \longrightarrow \Delta}\forall\longrightarrow$$

The rule has a proviso that the term t substituted for the variable x must be a Σ-term of index $\mu\nu$, to ensure that variables are only instantiated to appropriately defined terms. (The ($\longrightarrow\exists$) rule imposes an analogous constraint.)

The rule for initial sequents is now:

$$\Sigma \rhd \Gamma, A^{\mu} \longrightarrow A^{\mu\nu}, \Delta$$

The remaining rules mirror their classical and LMM counterparts. The full system is given in Fig. 3 as a calculus named LMP because the annotations denote paths to positions in the proof.

Consider the proofs of (1) and (2) in LMP. Making the scopes of the LMM proof (6) explicit with annotations gives the proof (7):

$$\dfrac{\left\{\dfrac{\dfrac{\dfrac{\{\rhd B\vee A^{\alpha},B^{\alpha},C^{\alpha\delta}\longrightarrow B^{\alpha\delta},\ldots}{\rhd B\vee A^{\alpha},B^{\alpha}\longrightarrow C\supset A^{\alpha},C\supset B^{\alpha},\ldots}\,{\scriptstyle\to\supset}\qquad\dfrac{\{\rhd B\vee A^{\alpha},A^{\alpha},C^{\alpha\beta}\longrightarrow A^{\alpha\beta},\ldots}{\rhd B\vee A^{\alpha},A^{\alpha}\longrightarrow C\supset A^{\alpha},C\supset B^{\alpha},\ldots}\,{\scriptstyle\to\supset}}{\rhd B\vee A^{\alpha}\longrightarrow(C\supset A)^{\alpha},(C\supset B)^{\alpha},\ldots}\,{\scriptstyle\vee\to}}{\rhd B\vee A^{\alpha}\longrightarrow(C\supset A)\vee(C\supset B)^{\alpha},\ldots}\,{\scriptstyle\to\vee}\right.}{\rhd\longrightarrow B\vee A\supset(C\supset A)\vee(C\supset B)}\,{\scriptstyle\to\supset}\tag{7}$$

This proof has the same structure as (6), and the same scopes are created. Now these scopes are also named: α names the scope introduced by the lower $(\to\supset)$ rule, $\alpha\beta$ and $\alpha\delta$ name the scopes introduced by the different assumptions of C.

As indicated by the ellipses, there is no dereliction of formulas on the right in (7). This allows rules to be permuted above $(\to\supset)$ and $(\to\forall)$. Eq. (8) illustrates this by moving the $(\to\supset)$ rules down below the application of $(\to\vee)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\rhd B\vee A^{\alpha},B^{\alpha},C^{\alpha\beta},C^{\alpha\delta}\longrightarrow B^{\alpha\delta},\ldots\qquad\rhd B\vee A^{\alpha},A^{\alpha},C^{\alpha\beta},C^{\alpha\delta}\longrightarrow A^{\alpha\beta},\ldots}{\rhd B\vee A^{\alpha},C^{\alpha\beta},C^{\alpha\delta}\longrightarrow A^{\alpha\beta},B^{\alpha\delta},\ldots}\,{\scriptstyle\vee\to}}{\rhd B\vee A^{\alpha},C^{\alpha\beta}\longrightarrow A^{\alpha\beta},(C\supset B)^{\alpha}}\,{\scriptstyle\to\supset}}{\rhd B\vee A^{\alpha}\longrightarrow(C\supset A)^{\alpha},(C\supset B)^{\alpha},\ldots}\,{\scriptstyle\to\supset}}{\rhd B\vee A^{\alpha}\longrightarrow(C\supset A)\vee(C\supset B)^{\alpha},\ldots}\,{\scriptstyle\to\vee}}{\rhd\longrightarrow B\vee A\supset(C\supset A)\vee(C\supset B)}\,{\scriptstyle\to\supset}\tag{8}$$

Because of these permutations, we can no longer isolate subproofs of (8) as recording all and only the inferences performed in a particular scope.

Meanwhile, an LMP proof corresponding to the LJ proof (1) appears in (9).

$$\dfrac{\dfrac{\dfrac{\rhd\ldots,C^{\alpha\beta}\longrightarrow C^{\alpha\beta}\qquad\dfrac{\dfrac{\rhd\ldots,A^{\alpha\beta}\longrightarrow A^{\alpha\beta}}{\rhd\ldots,A^{\alpha\beta}\longrightarrow A\vee B^{\alpha\beta}}\,{\scriptstyle\to\vee}\quad\dfrac{\rhd\ldots,B^{\alpha\beta}\longrightarrow B^{\alpha\beta}}{\rhd\ldots,B^{\alpha\beta}\longrightarrow A\vee B^{\alpha\beta}}\,{\scriptstyle\to\vee}}{\rhd\ldots,B\vee A^{\alpha\beta}\longrightarrow A\vee B^{\alpha\beta}}\,{\scriptstyle\vee\to}}{\rhd C\supset B\vee A^{\alpha},C^{\alpha\beta}\longrightarrow A\vee B^{\alpha\beta}}\,{\scriptstyle\supset\to}}{\rhd C\supset B\vee A^{\alpha}\longrightarrow C\supset A\vee B^{\alpha}}\,{\scriptstyle\to\supset}}{\rhd\longrightarrow(C\supset B\vee A)\supset(C\supset A\vee B)}\,{\scriptstyle\to\supset}\tag{9}$$

Note that the $(\supset\to)$ rule application involves a change of scope. The antecedent C of the conditional can only be established at scope extending $\alpha\beta$, because the axiom that establishes it uses the assumption of C at $\alpha\beta$. Thus, the conclusion $B\vee A$ of the conditional is established only at scope $\alpha\beta$. Because the $(\supset\to)$ rule refers to the annotation β, the $(\supset\to)$ rule cannot be permuted below the preceding $(\to\supset)$ rule. When the $(\to\supset)$ rule applies, β cannot appear on the sequent.

There is thus an asymmetry in LMP. It is possible to permute rules higher in the proof, even though the rules do not make use of the assumptions introduced there. But it is impossible to permute rules lower in the proof than the introduction of assumptions that they do use. This asymmetry will be eliminated by adapting Herbrand's theorem to LMP, in Section 5.1.

This asymmetry is the basis of the syntactic proof of soundness for LMP. Among the rules that occur above an $(\supset\longrightarrow)$ or $(\forall\longrightarrow)$ rule in an LMP proof, there will be all the rules that do depend on the assumption being made – those that should be there according to the structural regime of scope – as well as some that do not belong because they do not depend on the assumption being made. All those that do not belong can simply be permuted down to the scope where they do belong. The proof resulting from these permutations essentially matches LMM figures.

3.3.1. *Labels, semantics, and negation*

The annotations of LMP reflect dual intuitions. We have emphasized how annotations represent of the introduction of formulas and terms at different syntactic scopes in an LMM proof. The other intuition derives from the semantic interpretation of minimal logic formulas in Kripke models for modal logics [28]. For minimal logic these intuitions coincide, but for negation there is a possible discrepancy.

According to the semantic intuition, the labels corresponds to the points in the model at which formulas are true or false and at which individuals exist. In Kripke models, these points are worlds related by a transitive and reflexive binary relation R of accessibility. Each annotation represents a *path of accessibility* from the real world, reached by the empty path, to some other possible world. Paths of accessibility are a natural alternative to accessibility relations. Given R, we can construct a set of transitions such that whenever wRw', there is a transition α such that $w\alpha = w'$, and vice versa; see [35]. Here, the paths are strings because accessibility is transitive and reflexive: there is a single step of accessibility corresponding to each pair of steps, or to no step at all.

A further constraint on intuitionistic Kripke models is that atomic formulas true at a world w remain true at all worlds accessible from w. Similarly, an individual that exists at world w continues to exist at all worlds accessible from w. These constraints account for why the left annotation on initial sequents is to be a prefix of the right annotation, and for why the index of a substituted term is to be a prefix of the annotation of the formula into which it is substituted.

The remaining sequent rules implement classical reasoning over the recursive clauses defining truth of formulas. Implication is a good example. $A\supset B$ is true at a world μ exactly when for all transitions x to a world accessible from μ, if A is true at μx then B is true at μx. Regarding C^σ as an abbreviation of C is true at world σ, the classical inferences governing this definition are:

$$\cfrac{\cfrac{\Gamma, A^{\mu\alpha} \longrightarrow B^{\mu\alpha}, A^{\mu\alpha}\supset B^{\mu\alpha}, \forall x(A^{\mu x}\supset B^{\mu x}), \Delta}{\Gamma \longrightarrow A^{\mu\alpha}\supset B^{\mu\alpha}, \forall x(A^{\mu x}\supset B^{\mu x}), \Delta}\longrightarrow\supset}{\Gamma \longrightarrow \forall x(A^{\mu x}\supset B^{\mu x}), \Delta}\longrightarrow\forall$$

The ($\longrightarrow\supset$) inference rule of LMP imitates this exactly, replacing $\forall x(A^{\mu x} \supset B^{\mu x})$ by its equivalent $A \supset B^{\mu}$ and harmlessly omitting the intermediate step and the intermediate side formula.

The use of semantics for intuitionistic and modal deduction has been explored extensively, but has rarely been investigated from a purely proof-theoretic point of view. Inference rules for intuitionistic logic inspired by classical inferences are first given in [42]; except for negation the rules are analogous to those here – apart from Smullyan's use of the condensed format of tableaux with uniform notation. (Smullyan's system is, in turn, a refinement of Fitting's use of integer prefixes in tableau deduction [16, 17].) Wallen studies translation deduction in [50]; although the underlying mechanism mirrors Smullyan's, Wallen's presentation of it incorporates not only uniform notation and translation, but also matrix method search, unification, and structure-sharing. This complicates the task of applying Wallen's methods to other strategies for representation and proof search (for example the uniform proof search needed in abstract logic programming languages [32]). Both authors relate their systems directly to Kripke models, leaving open how similar systems might be used for proof construction and program synthesis. Subsequent research [1, 4, 13, 25, 35, 36] has explored the general use of similar techniques across a variety of modal systems, but has continued to present semantic proofs of correctness and to emphasize the use of particular theorem-proving techniques for first-order classical logic, particularly resolution.

Negation offers a venue in which to distinguish proof-theoretic intuitions from these semantic ones. In LMM, sequent rules for intuitionistic negation can be given directly, as below:

$$\frac{\Gamma \longrightarrow A, \Delta}{\Gamma, \neg A \longrightarrow \Delta}\neg\longrightarrow \qquad \frac{\Gamma, A \longrightarrow}{\Gamma \longrightarrow \neg A, \Delta}\longrightarrow\neg$$

Alternatively, $\neg A$ can be rendered $A \supset \bot$, where $\bot$ is a distinguished proposition with the following behavior:

$$\frac{}{\Gamma, \bot \longrightarrow \Delta}\bot$$

Wallen [50] uses the first formulation. His translation corresponds to the use of the two sequent rules below.

$$\frac{\Sigma \rhd \Gamma \longrightarrow A^{\mu\nu}, \Delta}{\Sigma \rhd \Gamma, \neg A^{\mu} \longrightarrow \Delta}\neg\longrightarrow \qquad \frac{\Sigma \rhd \Gamma, A^{\mu\alpha} \longrightarrow \Delta}{\Sigma \rhd \Gamma \longrightarrow \neg A^{\mu}, \Delta}\longrightarrow\neg$$

These rules correspond to the usual truth conditions for $\bot$ in Kripke models: at no world is $\bot$ true.

In the presence of these rules, the only proof of some sequent

$$\Sigma \rhd \Gamma, \Gamma^{*} \longrightarrow \Delta$$

may come from proving A and $\neg A$ from formulas in Γ^{*}. The same problem arises if the $\bot$ rule is realized with the formulation:

$$\frac{}{\Sigma \rhd \Gamma, \bot^{\mu} \longrightarrow \Delta}\bot$$

These possibilities disrupt the proof-theoretic invariants which we will use to establish the correspondence between LMP and LMM.

One way to describe what goes wrong is this. These two presentations of negation are constructed so as to avoid making the decision of which formula in a sequent is being derived by contradiction. Such lack of commitment is generally advantageous in search – this paper in fact is concerned with delaying the similar choice of the scope in which rules are used. For reasoning about scope, however, this treatment of negation is problematic because a complete proof need never resolve this ambiguity. Such proofs lack a crucial piece of information necessary to reconstruct an intuitionistic natural deduction. For example, consider the proof below:

$$\dfrac{\dfrac{\dfrac{\dfrac{A^{\alpha} \longrightarrow A^{\alpha\beta}, C^{\alpha}, B^{\alpha\beta}}{A^{\alpha}, \neg A^{\alpha\beta} \longrightarrow C^{\alpha}, B^{\alpha\beta}}\neg\longrightarrow}{A^{\alpha} \longrightarrow C^{\alpha}, \neg A \supset B^{\alpha}}\longrightarrow\supset}{A^{\alpha} \longrightarrow (C \vee (\neg A \supset B))^{\alpha}}\longrightarrow\vee}{\longrightarrow A \supset C \vee (\neg A \supset B)}\longrightarrow\supset$$

From root up, we first decompose formulas on the right, and then obtain an initial sequent by applying $(\neg \longrightarrow)$; the inference extends the annotation $\alpha\beta$ of the principal formula by the empty string (but any other extension would also give a proof). This can only correspond to a natural deduction in which the contradiction of A and $\neg A$ is used *to infer* B. But nothing about the sequent proof indicates this: for all this proof says, the contradiction could be used to show C.

From a proof-theoretic perspective, the rule below is more appropriate:

$$\dfrac{}{\Sigma \rhd \Gamma, \bot^{\mu} \longrightarrow A^{\mu\nu}, \Delta}\bot$$

It maintains scope in negation. It corresponds not to ordinary Kripke semantics but to fallible Kripke semantics [48]. This semantics allows $\bot$ to be true at some worlds, provided that every other atomic formula is also true at those worlds. Fallible semantics was developed to enable constructive completeness proofs for intuitionistic logic; so it is not surprising that it comes up again here where constructive reasoning about proofs is also required.

This proof theory and its fallible semantics is quite close to minimal logic, which treats $\bot$ as an ordinary proposition. For example, the rule can be simulated straightforwardly by recursively translating each goal formula A (i.e. any positive subformula of a formula on the right of the sequent, or any negative subformula on the left) into the disjunction $\bot \vee A$, and thereafter treating $\bot$ as an ordinary proposition. In view of this translation, the analysis in the remainder of this paper will be concerned with minimal logic only.

3.4. *Proof-theoretic results*

This section provides a direct, proof-theoretic justification of LMP: We establish that the annotations are nothing more than a proof-theoretic device for indicating scope in an intuitionistic deduction. Labels on formulas ensure syntactically that a classical proof system – with arbitrary permutabilities of rules, subject only to eigenvariable conditions – respects intuitionistic information-flow: that facts assumed as part of proving some formula only contribute to establishing that formula.

3.4.1. Some basic results

We begin with some invariants on the form and function of annotations in LMP proofs. The main point is to validate the basic properties of the design of LMP – that annotation variables may be regarded as naming individual inferences that change scope (Lemma 3) and that all the scopes of inferences in the proof lie in a tree whose nodes are uniquely labeled by these names (Lemma 4).

As a preliminary, we obtain a first result that suggests intuitively how annotations restrict information-flow in proofs. Recall that a left formula can only be used in the axiom rule of LMP when it is annotated with a prefix of some right formula. In fact, induction shows that a left formula cannot be used anywhere in an LMP proof unless it is annotated with a prefix of some right formula in the end sequent. Thus, when nested scopes are represented by longer annotations, annotations will ensure that assumptions introduced inside a nested scope cannot be used outside.

Lemma 2 (Irrelevance). *Let $\mathscr{D}$ be an LMP proof of height h of*

$$\Sigma \rhd \Gamma, \Gamma^* \longrightarrow \Delta$$

where for every formula A^μ in Γ^*, there is no formula B^ν in Δ such that μ is a prefix of ν. Then $\mathscr{D}$ can be transformed into a proof with height no more than h of

$$\Sigma \rhd \Gamma \longrightarrow \Delta.$$

Proof. By induction on the structure of proofs. At axiom links, the labels on Γ^* will not match those of the key right formula $A^{\mu\nu}$ of the axiom, so the left formula A^μ must be from Γ.

Supposing the claim true for proofs of height n or less, consider proofs of height $n+1$. The rules $(\vee\longrightarrow)$, $(\wedge\longrightarrow)$, $(\exists\longrightarrow)$, $(\longrightarrow\vee)$, $(\longrightarrow\wedge)$, and $(\longrightarrow\exists)$ do not alter annotations, so extending the induction hypothesis is straightforward. For example, suppose $\mathscr{D}$ ends in

$$\frac{\Sigma \rhd \Gamma, \Gamma^*, A\wedge B^\mu, A^\mu, B^\mu \longrightarrow \Delta}{\Sigma \rhd \Gamma, \Gamma^*, A\wedge B^\mu \longrightarrow \Delta}\wedge\longrightarrow$$

Applying the induction hypothesis gives a proof with one of the following two conclusions, depending on whether μ is a prefix of the annotation of any formula in Δ:

$$\Sigma \rhd \Gamma \longrightarrow \Delta \quad \text{or} \quad \Sigma \rhd \Gamma, A\wedge B^\mu, A^\mu, B^\mu \longrightarrow \Delta.$$

In the first case this derivation suffices; in the second, the needed derivation is constructed by finally applying $(\wedge\longrightarrow)$ again.

Even though they may extend annotations, ($\forall\rightarrow$), ($\rightarrow\forall$) and ($\rightarrow\supset$) are not much harder. Thus, suppose $\mathscr{D}$ ends in

$$\frac{\Sigma, a:\mu\alpha \rhd \Gamma, \Gamma^* \longrightarrow A[a/x]^{\mu\alpha}, \forall x A^{\mu}, \Delta}{\Sigma \rhd \Gamma, \Gamma^* \longrightarrow \forall x A^{\mu}, \Delta}\rightarrow\forall$$

Observe that if no formula in Γ^* is annotated with a prefix of μ, no formula in Γ^* is annotated with a prefix of $\mu\alpha$: by the eigenvariable condition, α does not occur in Γ^*. Accordingly, the induction hypothesis applies to the immediate subderivation to give a derivation of

$$\Sigma, a:\mu\alpha \rhd \Gamma \longrightarrow A[a/x]^{\mu\alpha}, \forall x A^{\mu}, \Delta.$$

Apply ($\rightarrow\forall$) to this.

Finally, suppose $\mathscr{D}$ ends in

$$\frac{\Sigma \rhd \Gamma, \Gamma^*, A\supset B^{\mu} \longrightarrow A^{\mu\nu}, \Delta \qquad \Sigma \rhd \Gamma, \Gamma^*, A\supset B^{\mu}, B^{\mu\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, \Gamma^*, A\supset B^{\mu} \longrightarrow \Delta}\supset\rightarrow.$$

If $\mu\nu$ is not a prefix of any Δ annotation, then the induction hypothesis applies to the right subderivation to give a proof of

$$\Sigma \rhd \Gamma' \longrightarrow \Delta.$$

where Γ' includes Γ and, if appropriate, $A\supset B^{\mu}$. This suffices. Otherwise, $\mu\nu$ is a prefix of some annotation of Δ, so if no formula in Γ^* is annotated with a prefix of any Δ annotation, then the same is true of $A^{\mu\nu}, \Delta$. Thus, induction gives proofs:

$$\Sigma \rhd \Gamma, A\supset B^{\mu}, B^{\mu\nu} \longrightarrow \Delta \quad \text{and} \quad \Sigma \rhd \Gamma, A\supset B^{\mu} \longrightarrow A^{\mu\nu}, \Delta.$$

Combine these by ($\supset\rightarrow$). □

As in other sequent systems, we can rename eigenvariables in a proof so that no two rules in a proof introduce the same one.

Definition 2 (*Pure variable proof*). A proof tree in LMP is a pure-variable proof tree if and only for every symbol $\underline{a}$ occurring as the eigenvariable in an application of the rules ($\rightarrow\forall$), ($\rightarrow\supset$) or ($\exists\rightarrow$), $\underline{a}$ does not occur both free and bound in any formula in the proof tree, and $\underline{a}$ only occurs in the subtree rooted at the sequent constituting the premise of the rule. (Note: Here $\underline{a}$ ranges over eigenvariables introduced in terms or on annotations.)

Lemma 3 (Pure variable proofs). *Any LMP deduction with end sequent $\Sigma \rhd \Gamma \rightarrow \Delta$ can be converted to a pure-variable proof of the same sequent, simply by replacing occurrences of variables with new variables* (*cf.* [19, p. 312]).

Proof (*sketch*). The proof can be adapted from that for other sequent systems (see for example [19, pp. 274–276, 312, 313]). One first shows that given an LMP derivation

$\mathscr{D}$ with end sequent $\Sigma \triangleright \Gamma \longrightarrow \Delta$ in which the variable b does not occur bound and the variable a does not occur at all, the derivation $\mathscr{D}[a/b]$ obtained by substituting a for every occurrence of b in $\mathscr{D}$ is an LMP derivation of $\Sigma[a/b] \triangleright \Gamma[a/b] \longrightarrow \Delta[a/b]$. This is a straightforward induction on the height of proofs, in which the indexing contexts and annotations add no additional complexity. The main result is established by induction on the number of applications of $(\longrightarrow\forall)$, $(\longrightarrow\supset)$, and $(\exists\longrightarrow)$ in the derivation. One applies the induction hypothesis to obtain pure variable proofs corresponding to subderivations above the applications of these rules closest to the root, then exploits the substitution property to ensure that the eigenvariables in different subderivations are distinct. □

According to renaming, any eigenvariable is introduced exactly once onto a right formula, and according to irrelevance, annotations on left formulas can match the annotations of right formulas without loss of generality. Together, this means that each annotation eigenvariable always appears after the same sequence of other eigenvariables: it is always used the same way, to represent the same scope.

Definition 3 (*Unique prefix property*). A set of annotations has the unique prefix property if and only if for any pair $\mu\alpha v$ and $\mu'\alpha v'$ in the set, $\mu = \mu'$.

Lemma 4 (Tree annotations). *For any LMP derivation $\mathscr{D}$ of height h with end sequent*

$$\Sigma \triangleright \Gamma \longrightarrow \Delta$$

where the annotations in the end sequent have the unique prefix property, there is a derivation $\mathscr{D}'$ of height no larger than h with end sequent

$$\Sigma \triangleright \Gamma' \longrightarrow \Delta$$

where the formulas in Γ' are a subset of those that occur in Γ and the annotations appearing throughout $\mathscr{D}'$ have the unique prefix property.

Proof. By Lemma 3, we may assume $\mathscr{D}$ is a pure variable proof. By Lemma 2, we can obtain a shorter derivation $\mathscr{D}'$ from $\mathscr{D}$ in which the annotation of left formulas are always prefixes of the annotations of right formulas. By induction, we can show that any derivation $\mathscr{D}'$ so constructed, whose end sequent has the unique prefix property, has the unique prefix property. For initial sequents, this is immediate. So suppose some rule application results in a sequent with the unique prefix property. If it is a left rule, the unique prefix property must extend to the immediate subderivations: left rules will not result in new prefixes because they would be irrelevant. Likewise, although immediate subderivations for right rules may incorporate additional prefixes, these prefixes must involve fresh symbols. Thus, the unique prefix property also extends to immediate subderivations of right rules. Hence, the induction hypothesis applies to show that subderivations have the unique prefix property in their entirety. Then the fact that the proof is a pure variable proof means eigenvariables are introduced once only, so that no different prefixes are introduced for the same eigenvariable in the separate subderivations. Thus the whole proof has the unique prefix property. □

The fact that eigenvariables on annotations in deductions have unique prefixes is significant both for demonstrating that the annotation mechanism is correct and for constructing algorithms to work on annotations. As we shall see presently, it allows the annotations in LMP proofs to be put in correspondence with positions in LMM proofs. In [4, 13, 37], the unique prefix property is used to construct specialized equational unification algorithms for equations in translation theorem-proving; in [43] it plays a role in deriving constraint algorithms for translation theorem-proving.

3.4.2. LMP is complete

We first consider how any LMM proof can be converted into an LMP proof, by adding appropriate annotations throughout. The crux of the argument is how annotations are extended by instantiation with appropriate terms in applications of $(\supset\rightarrow)$ and $(\forall\rightarrow)$. Here is the idea. In LMM, the logical scope of a rule application R is given by its position in the proof. This position is identified by the sequence of $(\rightarrow\supset)$ and $(\rightarrow\forall)$ rules that occur on the path from the root to R. In particular, *all* rule applications above an application of $(\rightarrow\supset)$ or $(\rightarrow\forall)$ in LMM fall into its scope. In LMP, scope is given by the annotation that labels a formula – but the eigenvariables in this annotation correspond one-to-one with the rule applications in the LMM proof. Thus, in translating applications of $(\supset\rightarrow)$ and $(\forall\rightarrow)$ from the LMM proof, the annotations will be extended to the full LMP annotation that corresponds to their scoped location in the LMM deduction.

Some notation describing the addition describing the addition of annotations to sequents will facilitate the presentation of this result. For any annotation string μ and any multiset of formulas Δ, Δ^{μ} will denote the sequent consisting of a formula occurrence A^{μ} for each formula occurrence A of Δ. If Δ is the right hand of a sequent in the LMM proof, in the scope in the proof associated with the LMP annotation μ, Δ^{μ} will be the right hand of the corresponding sequent in the LMP proof. Meanwhile, for any map η associating a (possibly different) annotation string with each formula occurrence in a sequent Γ, Γ^{η} will denote the sequent containing a formula occurrence $A^{\eta(A)}$ for each formula occurrence A of Γ. A left side of a sequent Γ in an LMM proof may correspond to any Γ^{η} in the corresponding LMP proof. The alternatives reflect ways of adding formulas on the left in the different scopes in which the μ scope is nested in the LMM proof.

Completeness is now stated as follows:

Theorem 1 (Completeness). *Let us be given any LMM deduction with end sequent $\Gamma \longrightarrow \Delta$. Then for any annotation string μ, any function η assigning prefixes of μ to formulas in Γ, and any indexing context Σ assigning prefixes of μ to the free variables of Γ and Δ, there is an LMP deduction of*

$$\Sigma \rhd \Gamma^{\eta} \longrightarrow \Delta^{\mu}.$$

Proof. By induction on the structure of derivations in LMM. For LMM axioms, an LMP axiom can be constructed using the fact that η assigns to A on the left some prefix σ of the annotation of A on the right, μ.

The cases for $\vee, \wedge$ and $(\exists \longrightarrow)$ are straightforward. Given an LMM derivation ending in the application of one of these rules, apply the induction hypothesis to the immediate subderivations: this gives LMP deductions to which the corresponding LMP rule applies. For example, for $(\longrightarrow \vee)$ the LMM derivation ends:

$$\frac{\Gamma \longrightarrow A, B, A \vee B, \Delta}{\Gamma \longrightarrow A \vee B, \Delta} \longrightarrow \vee$$

The corresponding LMP proof ends:

$$\frac{\Sigma \rhd \Gamma^{\eta} \longrightarrow A^{\mu}, B^{\mu}, A \vee B^{\mu}, \Delta^{\mu}}{\Sigma \rhd \Gamma^{\eta} \longrightarrow A \vee B^{\mu}, \Delta^{\mu}} \longrightarrow \vee$$

The subderivation is obtained by the induction hypothesis from the subderivation of the LMM derivation.

For the rules $(\supset \longrightarrow)$ and $(\forall \longrightarrow)$, we ensure that the annotation of right formulas μ appears as the annotation of side formulas in applications of these rules in constructing the LMP proof. Thus, suppose $\mathscr{D}$ ends in $(\supset \longrightarrow)$, as follows:

$$\frac{\Gamma, A \supset B \longrightarrow A, \Delta \qquad \Gamma, A \supset B, B \longrightarrow \Delta}{\Gamma, A \supset B \longrightarrow \Delta} \supset \longrightarrow$$

Apply the induction hypothesis to the first subderivation with μ and η, and to the second derivation with μ and a function η' exactly like η except that $\eta'(B) = \mu$. This gives derivations of

$$\Sigma \rhd \Gamma^{\eta}, A \supset B^{\eta(A \supset B)} \longrightarrow A^{\mu}, \Delta^{\mu} \quad \text{and} \quad \Sigma \rhd \Gamma^{\eta}, A \supset B^{\eta(A \supset B)}, B^{\mu} \longrightarrow \Delta^{\mu}$$

Since $\eta(A \supset B)$ is a prefix of μ, these two derivations can be combined by LMP $(\supset \longrightarrow)$ to yield the needed overall derivation.

Similarly, suppose $\mathscr{D}$ ends in $(\forall \longrightarrow)$, as follows:

$$\frac{\Gamma, \forall x A, A[t/x] \longrightarrow \Delta}{\Gamma, \forall x A \longrightarrow \Delta} \forall \longrightarrow$$

Apply the induction hypothesis to the subderivation, with μ and the function η' exactly like η except that $\eta'(A[t/x]) = \mu$. This gives a derivation of:

$$\Sigma \rhd \Gamma^{\eta}, \forall x A^{\eta(\forall x A)}, A[t/x]^{\mu} \longrightarrow \Delta^{\mu}.$$

Because every free variable in Σ is decorated with a prefix of μ, any term t must be a Σ-term of index μ. So the side condition on substitutions for $(\forall \longrightarrow)$ in LMP is satisfied; applying the rule gives the needed derivation.

Likewise, for the case of an LMM derivation $\mathscr{D}$ ending in $(\longrightarrow \exists)$, free variables are all associated with prefixes of μ by Σ, so any term t is a Σ-term of index μ. But as right formulas, both $\exists xA$ and $A[t/x]$ are to be annotated with μ. Thus the LMP $(\rightarrow\exists)$ rule applies to the derivation obtained by the induction hypothesis from the immediate subderivation of $\mathscr{D}$. The resulting derivation ends:

$$\frac{\Sigma \triangleright \Gamma^{\eta} \longrightarrow A[t/x]^{\mu}, \exists xA^{\mu}, \Delta^{\mu}}{\Sigma \triangleright \Gamma^{\eta} \longrightarrow \exists xA^{\mu}, \Delta^{\mu}} \rightarrow\exists$$

Finally, for the rules $(\rightarrow\supset)$ and $(\rightarrow\forall)$, observe that Δ is eliminated when the annotation of the principal formula is extended. (This is why it suffices to consider translations in which all right formulas receive the same annotation.) Specifically, suppose the LMM deduction ends in

$$\frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B, \Delta} \rightarrow\supset$$

Apply the induction hypothesis to the immediate subderivation, with $\mu' = \mu\alpha$ for some new α, and η' like η except $\eta'(A) = \mu'$. This gives a derivation $\mathscr{D}$ ending in:

$$\Sigma \triangleright \Gamma^{\eta}, A^{\mu\alpha} \longrightarrow B^{\mu\alpha}.$$

This can be weakened by the formulas in Δ^{μ}, so as then to derive the needed:

$$\frac{\begin{array}{c}\mathscr{D} + \Delta^{\mu}\\ \Sigma^{\eta} \triangleright \Gamma^{\eta}, A^{\mu\alpha} \longrightarrow B^{\mu\alpha}, \Delta^{\mu}\end{array}}{\Sigma^{\eta} \triangleright \Gamma^{\eta} \longrightarrow A \supset B^{\mu}, \Delta^{\mu}} \rightarrow\supset$$

Likewise, if the LMM deduction ends in

$$\frac{\Gamma \longrightarrow A[a/x]}{\Gamma \longrightarrow \forall xA, \Delta} \rightarrow\forall$$

apply the induction hypothesis to the immediate subderivation, with $\mu' = \mu\alpha$ for some new α, and Σ' extending Σ by the assignment $a : \mu\alpha$. This gives a derivation $\mathscr{D}$. Construct in LMP the needed derivation:

$$\frac{\begin{array}{c}\mathscr{D} + \Delta^{\mu}\\ \Sigma^{\eta}, a : \mu\alpha \triangleright \Gamma^{\eta} \longrightarrow A[t/x]^{\mu\alpha}, \Delta^{\mu}\end{array}}{\Sigma^{\eta} \triangleright \Gamma^{\eta} \longrightarrow \forall xA^{\mu}, \Delta^{\mu}} \rightarrow\forall \qquad \square$$

3.4.3. LMP is sound

Inversely, every LMP proof can be transformed into an LMM proof. Because of the different ways scopes are represented in the two systems, the transformation involves reorganizing the proof so that right formulas with compatible annotations end up together in the same scoped region of the proof. In particular, the transformation gives a way of reordering applications of proof rules so that only what must appear above any application of $(\rightarrow\supset)$ and $(\rightarrow\forall)$ actually does appear there. When applications appear in this order, an invariant of annotations can be exploited to demonstrate that the subproof above each $(\rightarrow\supset)$ and $(\rightarrow\forall)$ rule constitutes a proof of its right-side formula. Thus the transformed proof instantiates only LMM inference figures. Overall, the method recalls Schellinx's proof-theoretic justification of the embedding of intuitionistic logic into linear logic [39].

In considering just a *single* application P of $(\rightarrow\supset)$ or $(\rightarrow\forall)$, there is a simple characterization of which other rule applications permute below P.

Lemma 5 (Permutability). *Let $\mathscr{D}$ be an LMP derivation containing an application P of $(\rightarrow\supset)$ or $(\rightarrow\forall)$, which introduces the eigenvariable α in the annotation. Let R denote any other rule application in $\mathscr{D}$ above P, and let μ be the annotation of the side formula of R. Then R permutes with P if and only if μ does not contain α.*

Proof. If μ contains α, then permuting R below P will violate the eigenvariable condition for α. This takes care of the *only if* case.

For the *if* case, consider first which pairs of rules might fail to permute. There are only five possibilities: $(\supset\rightarrow)/(\rightarrow\supset)$, $(\supset\rightarrow)/(\rightarrow\forall)$, $(\forall\rightarrow)/(\rightarrow\forall)$, $(\forall\rightarrow)/(\exists\rightarrow)$ and $(\rightarrow\exists)/(\exists\rightarrow)$. They arise only when eigenvariable conditions will be violated by the permuting of rules. (The proof transformations needed to achieve the permutation in all other cases are just the transformations from [26] illustrated in the proof of Lemma 1.) For each of these nonpermuting examples, it can be shown that the annotation introduced by the lower rule is a prefix of the annotation introduced by the higher rule. For impermutabilities $(\supset\rightarrow)/(\rightarrow\supset)$ and $(\supset\rightarrow)/(\rightarrow\forall)$, the appearance in the annotation of the symbol α introduced lower is the source of the impermutability. For quantifier impermutabilities, the source of the impermutability might be the appearance of the lower eigenvariable, a, in the substitution term, t. Nevertheless, in these cases, the side condition that the term t be a Σ-term of appropriate index applies. This ensures that the annotation of the lower rule appears on the annotation of the higher rule, in virtue of its association with the eigenvariable in the indexing context. Incidentally, also as a result of this side condition, the configuration that would give rise to a $(\rightarrow\exists)/(\rightarrow\forall)$ impermutability (which would otherwise be expected) cannot be given a legal annotation.

Given these observations, the claim can be established by an induction on the structure of LMP derivations. The measure for the induction is the number of applications of rules above P and below R. The base case we have just shown. Assume that when an application is no more than n steps above P, it permutes below P just in case the annotations of its side formulas do not contain α. Consider the case of a rule application, R, $n+1$ steps above P. Let Q be the rule application immediately below R. If R permutes below Q do so: this reduces by one the distance between R and P, and permits the application of the induction hypothesis. If R does not permute below Q, consider whether Q permutes below P. If not, by the induction hypothesis the side formulas of Q must contain α. But then, by the observations of the preceding paragraph show that R, which does not permute below Q, must contain α as well. Otherwise, the induction hypothesis applies to Q and P, to give a deduction in which R is only n rule applications above P (because Q has been permuted below P). Apply the induction hypothesis to this derivation. □

The following consequence of the permutability lemma is at the heart of the soundness of LMP.

Lemma 6. *Let $\mathscr{D}$ be a deduction in LMP of*

$$\Sigma \rhd \Gamma \longrightarrow C, \Delta$$

such that C is of the form $\forall x A$ or $A \supset B$; C is the principal formula of the lowest rule application P in $\mathscr{D}$ where the eigenvariable α is introduced on annotations; and no higher rule application in $\mathscr{D}$ permutes below P. Then either

$$\Sigma \rhd \Gamma \longrightarrow \Delta$$

is an axiom or we can construct a deduction $\mathscr{D}'$ from $\mathscr{D}$ which shows

$$\Sigma \rhd \Gamma \longrightarrow C.$$

Proof. Right rules change annotations only by adding eigenvariables. Because eigenvariables may be considered distinct (by obtaining a pure variable proof as in Lemma 3), and because addition of another eigenvariable to an annotation indicates incompatibility with α (by the unique prefix property, as in Lemma 4), we may assume that no descendants of Δ are labeled with annotations that contain α in $\mathscr{D}$. Now, consider an axiom link in $\mathscr{D}$

$$\Sigma' \rhd \Gamma', A^{\mu} \longrightarrow A^{\mu\nu}, \Delta'$$

where $\mu\nu$ does not contain α. Then neither A^{μ} nor $A^{\mu\nu}$ is a side formula of any rule in $\mathscr{D}$. Such a rule by Lemma 5 on permutability could permute below P. Therefore, A^{μ} must be already in Γ and $A^{\mu\nu}$ in Δ: we start from an axiom.

Otherwise, every application of an axiom in $\mathscr{D}$ involves a right formula annotated with a string containing α. So each right formula is a descendant of C, not of any formula in Δ. Meanwhile, each left formula in an axiom is either annotated with a string that contains α – in which case it too is a descendant of C – or with one that does not – in which case it is in fact a formula in Γ. Accordingly, we can obtain a proof of C by erasing all descendants of Δ everywhere in $\mathscr{D}$. □

In order to turn this result into a method of converting LMP proofs into LMM proofs, we need to show that we can permute the rules in the proof so that *all* implications and universals in the proof lie simultaneously under only the necessary rules. This will ensure that each such rule involves a proof of its side formula, and hence that the proof can be described by LMM rules.

Theorem 2 (Soundness). *From any LMP deduction $\mathscr{D}$ a deduction $\mathscr{D}'$ can be constructed with the following property. For any subderivation $\mathscr{D}''$ of $\mathscr{D}'$ ending in $(\longrightarrow\supset)$*

or $(\rightarrow\forall)$, *no rule applied above in* $\mathscr{D}''$ *can be permuted below. By applying Lemma* 6 *recursively to* $\mathscr{D}'$, *we obtain an LMM proof.*

Proof. The proof is by induction on the number of times rule applications of $(\rightarrow\supset)$ and $(\rightarrow\forall)$ occur on any path from a leaf of the derivation to the conclusion. The base case, when the derivation contains no such formulas, is immediate.

Suppose the claim holds for proofs where $(\rightarrow\forall)$ or $(\rightarrow\supset)$ formulas occur at most n times from a leaf to the root. In any derivation where these rules occur at most $n+1$ times, it suffices to replace the subderivations ending in $(\rightarrow\supset)$ and $(\rightarrow\forall)$ with appropriately reordered variants. Accordingly, we consider a derivation $\mathscr{D}$ ending in a rule application R with principal formula C at

$$\Sigma \rhd \Gamma \longrightarrow C, \Delta$$

with C a universal or an implication with depth $n+1$. Let α be the annotation eigenvariable introduced.

Let $\mathscr{D}_1$ denote the derivation obtained by first applying the induction hypothesis to the immediate subderivation, and then permuting below R all rules that permute below *every* application of $(\rightarrow\supset)$ and $(\rightarrow\forall)$ in the proof. The possibility of such permutations can be established as in Lemma 1 by a double induction first on the degree – the number of inferences which refer to an annotation eigenvariable introduced at a $(\rightarrow\supset)$ or $(\rightarrow\forall)$ rule in the proof but with a rule above them that does not – and then considering the highest inference with something above it that should be below it, on the number of such inferences above it (the grade).

In $\mathscr{D}_1$, R may be applied to C several times with modified end sequents, because of these permutations:

$$\Sigma_1 \rhd \Gamma_1 \longrightarrow C, \Delta_1$$

We can eliminate each as follows. Suppose there is some subproof of $\mathscr{D}_1$ above it, with $\forall$ and $\supset$ depth n, that looks like this:

$$\Sigma' \rhd \Gamma' \longrightarrow D^{\mu}, \Delta$$

with D a universal or implication, where μ does not contain α. Applying Lemma 6 and Lemma 2 of irrelevance gives a proof where Γ' is restricted only to formulas containing μ only – not α. Permutability dictates that all these formulas are therefore elements of Γ_1. Likewise, D^{μ} is an element of Δ_1. Accordingly, we can use (a weakened version of) this shorter proof instead of the proof involving C in $\mathscr{D}_1$.

Otherwise, in every such formula D^{μ} above R, μ contains α. This entails that the rule applications above R are precisely those that cannot be permuted below R. The preceding result now applies to show that either

$$\Sigma_1 \rhd \Gamma_1 \longrightarrow \Delta_1 \quad \text{or} \quad \Sigma_1 \rhd \Gamma_1 \longrightarrow C.$$

Substituting whichever proof exists into $\mathscr{D}_1$ gives the needed derivation. □

4. A broader view of deduction and explicit scope

This section puts LMP in context with alternative theorem-proving work. As discussed in Section 4.4, the use of LMP seems generally compatible with a variety of complementary techniques for improving proof search based on observations about size and branching of proofs [12, 14, 23]; some effort is required to state these observations in a common language.

The bulk of this section, Sections 4.1–4.3, is devoted to contrasting LMP with other labeled systems from a proof-theoretic point of view. LMP offers an explicit representation of scope in intuitionistic proofs based on the position of rules in the proof. Another strategy is to adopt an explicit representation of scope based on the content of sequents. With scoping based on content, labels represent the packets of assumptions from which a formula in the proof must be derived. The idea has been explored semantically in [11] and proof theoretically in [6, 7].

For example, the labeling of ($\longrightarrow\supset$) by content goes as follows. According to the labeling scheme, assume that the principal formula ($A \supset B$) already depends on some packet of assumptions σ. The rule introduces a new assumption A, which therefore gets a new atomic label, say α. A new conclusion B must then be derived using the combination of σ and α, written $\sigma \circ \alpha$. This labeling is implemented by using the following sequent rule (with an eigenvariable condition for α):

$$\frac{\Sigma \rhd \Gamma, A^{\alpha} \longrightarrow B^{\sigma\circ\alpha}, A \supset B^{\sigma}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A \supset B^{\sigma}, \Delta} \longrightarrow\supset$$

Systems based on position and systems based on content have very different intuitions behind them. It might therefore be suspected that the systems have very different behavior – and that labeling by content might be easier to prove correct because of its closer correspondence with the ordinary sequent rules. The reality is more subtle. In fact, some compromise of intuitions is required to turn scoping by content into an efficient theorem-proving method. These compromises leave the method in a surprisingly close formal correspondence to scoping by position. Scoping by position may even be regarded as the basic method on which others are variants.

To back this up, we observe first in Section 4.1 that our encoding of positions as strings with LMP was not the only possibility. The same inference rules and proof techniques apply if we encode positions using sets. We then show in Section 4.2 how the labels of [11] can be justified by interpreting them as abbreviations of these sets. The labels of [6, 7] offer a different kind of abbreviation for these sets, as outlined in Section 4.3.

4.1. Set structure for annotations

In this section, we adapt the rules of LMP by treating annotations as sets or multisets rather than strings. Like the string labels in LMP, these labels name the rules applied on a path to a particular scoped position in a structurally scoped proof. Unlike string labels, however, these labels provide only partial descriptions of positions in structurally scoped proofs. They do not specify the order in which rules are to apply. These new labelings are possible because this order can be reconstructed from the labeling of rules. Rules that change labels represent points of transition in the structurally scoped proof, between the position represented by one set and the position represented by an augmentation of it. We shall see that labels obtained by sequences of such augmentations in an explicitly scoped proof continue to correspond uniquely to a tree of (structurally scoped) rule applications.

To describe the modification to use sets or multisets is straightforward. To foreshadow the connection with [11], we introduce the notation $\mu \circ \nu$ to represent the combination of scopes μ and ν. To define string labels, we take $\circ$ to be an associative concatenation with identity ε:

$$\mu \circ \varepsilon = \varepsilon \circ \mu = \mu \quad \mu \circ (\nu \circ \rho) = (\mu \circ \nu) \circ \rho$$

To obtain multisets, we add commutativity to these equations:

$$\mu \circ \nu = \nu \circ \mu$$

Sets are also governed by the equation of idempotence:

$$\mu \circ \nu \circ \nu = \mu \circ \nu$$

Once we have written out the system using this $\circ$ notation, as in Fig. 4, we can use the same rules to describe labeled deduction for different equational theories. $(\forall \longrightarrow)$ illustrates this. The rule is:

$$\frac{\Sigma \rhd \Gamma, \forall x A^{\mu}, A[t/x]^{\mu \circ \nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, \forall x A^{\mu} \longrightarrow \Delta} \forall\longrightarrow$$

Depending on the equational theory in force, the extended scope $\mu \circ \nu$ represents either concatenation of μ and ν, with string labels; or the multiset union of μ and ν, with multiset labels; or the set union of μ and ν, with set labels. The side condition that t respect the scope $\mu \circ \nu$ is also rephrased in neutral language:

Definition 4. t is a Σ-term of index μ if and only if for every free variable x that occurs in t, Σ assigns $x : \mu'$ and for some ρ, $\mu' \circ \rho = \mu$.

We will refer to the system in Fig. 4 with set or multiset labels as LMS.

Like string annotations, set and multiset annotations can be motivated both from the syntax of intuitionistic proofs and as an encoding of intuitionistic semantics. In particular, set annotations implement the topological semantics of intuitionistic logic, where formulas are evaluated with respect to the open sets of a topological space; see [47, Ch. 13]. To account for set annotations, we interpret annotations as representing open sets in these models and interpret $\circ$ as set intersection.

$$\Sigma \triangleright \Gamma, A^{\mu} \longrightarrow A^{\mu\circ\nu}, \Delta$$

$$\frac{\Sigma \triangleright \Gamma, A \wedge B^{\mu}, A^{\mu}, B^{\mu} \longrightarrow \Delta}{\Sigma \triangleright \Gamma, A \wedge B^{\mu} \longrightarrow \Delta}\wedge\rightarrow$$

$$\frac{\Sigma \triangleright \Gamma \longrightarrow A^{\mu}, A \wedge B^{\mu}, \Delta \qquad \Sigma \triangleright \Gamma \longrightarrow B^{\mu}, A \wedge B^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow A \wedge B^{\mu}, \Delta}\rightarrow\wedge$$

$$\frac{\Sigma \triangleright \Gamma, A \vee B^{\mu}, A^{\mu} \longrightarrow \Delta \qquad \Sigma \triangleright \Gamma, A \vee B^{\mu}, B^{\mu} \longrightarrow \Delta}{\Sigma \triangleright \Gamma, A \vee B^{\mu} \longrightarrow \Delta}\vee\rightarrow$$

$$\frac{\Sigma \triangleright \Gamma \longrightarrow A^{\mu}, B^{\mu}, A \vee B^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow A \vee B^{\mu}, \Delta}\rightarrow\vee$$

$$\frac{\Sigma \triangleright \Gamma, A \supset B^{\mu} \longrightarrow A^{\mu\circ\nu}, \Delta \qquad \Sigma \triangleright \Gamma, A \supset B^{\mu}, B^{\mu\circ\nu} \longrightarrow \Delta}{\Sigma \triangleright \Gamma, A \supset B^{\mu} \longrightarrow \Delta}\supset\rightarrow$$

$$\frac{\Sigma \triangleright \Gamma, A^{\mu\circ\alpha} \longrightarrow B^{\mu\circ\alpha}, A \supset B^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow A \supset B^{\mu}, \Delta}\rightarrow\supset^{\dagger}$$

$$\frac{\Sigma \triangleright \Gamma, \forall x A^{\mu}, A[t/x]^{\mu\circ\nu} \longrightarrow \Delta}{\Sigma \triangleright \Gamma, (\forall x A)^{\mu} \longrightarrow \Delta}\forall\rightarrow^{\ddagger}$$

$$\frac{\Sigma, a : \mu \circ \alpha \triangleright \Gamma \longrightarrow A[a/x]^{\mu\circ\alpha}, \forall x A^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow \forall x A^{\mu}, \Delta}\rightarrow\forall^{\dagger}$$

$$\frac{\Sigma, a : \mu \triangleright \Gamma, \exists x A^{\mu}, A[a/x]^{\mu} \longrightarrow \Delta}{\Sigma \triangleright \Gamma, \exists x A^{\mu} \longrightarrow \Delta}\exists\rightarrow^{\dagger}$$

$$\frac{\Sigma \triangleright \Gamma \longrightarrow A[t/x]^{\mu}, \exists x A^{\mu}, \Delta}{\Sigma \triangleright \Gamma \longrightarrow \exists x A^{\mu}, \Delta}\rightarrow\exists^{\ddagger}$$

Fig. 4. A more general explicitly scoped cut-free sequent calculus for minimal logic, *LMS*. † For ($\rightarrow\forall$), ($\exists\rightarrow$) and ($\rightarrow\supset$), a and α must not appear in the conclusion. ‡ For ($\forall\rightarrow$), there is a proviso that t be a Σ-term of index $\mu \circ \nu$; for ($\rightarrow\exists$) that t be a Σ-term of index μ.

The labeling of initial sequents matches the truth conditions of atomic formulas in topological models, as follows. Each atomic formula A is assigned an open set $[\![A]\!]$ as its semantic value. A is true in a topological model at an open set q exactly when $q \subseteq [\![A]\!]$. An assumption that A is true at q thus allows the conclusion that A is true at any $q \cap r$, as given in the rule for initial sequents.

$A \supset B$ is true in a topological model at q exactly when B is true at all open subsets of q where A is true. This condition is equivalent to the condition that for any open set r, if A is true at $q \cap r$ then so is B. This corresponds to the proof rule for implication.

Topological semantics gives classical semantics to conjunction, as reflected in the LMS proof rule. The general definition for the semantics of disjunction in topological models is complicated, and involves decomposing a set of evaluation into a union of other sets. Rather than encoding this definition explicitly, LMS uses the classical proof rule for disjunction to simulate it.

A direct proof of correctness for LMS can be constructed exactly along the lines of Sections 3.4.1–3.4.3. An outline of this argument follows. The key difference is the statement and proof of the tree annotation lemma, since the unique prefix property cannot be defined in LMS (and a "unique subset property" would be false).

As before, we get an irrelevance result for LMS; it is shown by the same induction given in Section 3.4.1, with *subset* substituted for *prefix*. We also get a theorem that supplies pure variable proofs for LMS.

The tree annotation result must now go as follows.

Lemma 7 (Tree annotations). *Let $\mathscr{D}$ be an LMS derivation of height h with end sequent*

$$\Sigma \rhd \Gamma \longrightarrow \Delta$$

and suppose there is a pair of symbols α and β such that no annotation in the end sequent contains both α and β. Then there is a derivation $\mathscr{D}'$ of height no larger than h with end sequent

$$\Sigma \rhd \Gamma' \longrightarrow \Delta$$

where the formulas in Γ' are a subset of those that occur in Γ and no annotation in $\mathscr{D}'$ contains both α and β.

Proof. We assume $\mathscr{D}$ is a pure variable proof. By irrelevance, we can obtain a shorter derivation $\mathscr{D}'$ from $\mathscr{D}$ in which the annotation of left formulas are always subsets of the annotations of right formulas. By induction, we can show that, in any $\mathscr{D}'$ with this property, no annotation contains both α and β unless the annotation of some formula in the end sequent does. This is immediate at axioms. Suppose the claim is true of derivations of height n, and consider a derivation of height $n+1$ in whose end sequent no annotation contains both α and β. If the derivation ends in a left rule, no annotation will contain both α and β in the end sequent of its immediate subderivations: these combinations could not occur on the right, so a left rule introducing such combinations would be irrelevant. Likewise, although immediate subderivations for right rules may include larger sets of annotations, these sets will extend existing sets by fresh symbols different from α and β by the pure variable property. Thus, the absence of annotations combining α and β extends to the end sequents of immediate subderivations of right rules. Hence, the induction hypothesis applies to show that α and β never occur as elements of the same annotation throughout the whole proof. □

The proof of completeness given in Section 3.4.2 now carries over directly to LMS: here, too, the scoped geometry of LMM proofs determines what scope is needed in rule applications that extend annotations. In LMS, right formulas can be annotated with any set μ, and left formulas with arbitrary subsets of μ:

Theorem 3 (Completeness). *Let D be an LMM deduction with end sequent $\Gamma \longrightarrow \Delta$. Then for any set annotation μ, any function η assigning subsets of μ to formulas in Γ, and any indexing context Σ assigning subsets of μ to the free term variables of Γ and Δ, there is an LMS deduction of*

$$\Sigma \rhd \Gamma^{\eta} \longrightarrow \Delta^{\mu}.$$

Likewise, a proof of soundness for LMS can be formulated in terms of two lemmas as in Section 3.4.3. As with LMP, we use the Σ-term condition which relates substituted terms and formula annotations – formulated in terms of subsets not prefixes – to give a permutability lemma for LMS:

Lemma 8 (Permutability). *Let $\mathscr{D}$ be an LMS derivation containing an application P of $(\rightarrow\supset)$ or $(\rightarrow\forall)$, which introduces the eigenvariable α in the annotation. Let R denote any other rule application in $\mathscr{D}$ above P, and let μ be the annotation of the side formula of R. Then R permutes with P if and only if μ does not contain α.*

This result extends to an intermediate lemma:

Lemma 9. *Let $\mathscr{D}$ be a deduction in LMS of*

$$\Sigma \rhd \Gamma \longrightarrow C, \Delta$$

such that C is of the form $\forall xA$ or $A \supset B$; C is the principal formula of the lowest rule application P in $\mathscr{D}$ where the eigenvariable α is introduced on annotations; and no higher rule application in $\mathscr{D}$ permutes below P. Then either

$$\Sigma \rhd \Gamma \longrightarrow \Delta \text{ is an axiom}$$

or we can construct an LMS deduction $\mathscr{D}'$ from $\mathscr{D}$ which shows

$$\Sigma \rhd \Gamma \longrightarrow C.$$

This is because LMS shares the three critical properties used before to constrain axiom links in the deduction $\mathscr{D}$. First, the right sequent rules are formulated to change annotations by only adding eigenvariables. Second, separate variables must be distinct (by the pure variable proof lemma). Third, no formulas derived from Δ formulas in $\mathscr{D}$ are labeled with annotations that contain α. This third property is now an indirect consequence of the new Lemma 7 on tree annotations. Consider any rule application R whose principal formula is a descendant of a Δ formula with its original annotation, and which causes the addition of another eigenvariable β to this annotation on its side formulas (with Δ on the right, this is the only way annotations of descendants of Δ might change). Because of the eigenvariable condition, no annotation in the end sequent of the immediate subderivations of R contains both α and β: Lemma 7 applies to show that no formula in the entire deduction is labeled with both α and β.

Applying this result recursively, as earlier, demonstrates the completeness of LMS.

4.2. Abbreviating set annotations

Consider a labeled system with set annotations, just as in the previous subsection, except with the following revised rules for $\supset$ and $\forall$:

$$\frac{\Sigma \rhd \Gamma, A^{\alpha} \longrightarrow B^{\sigma\circ\alpha}, A \supset B^{\sigma}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A \supset B^{\sigma}, \Delta} \rightarrow\supset$$

$$\frac{\Sigma \rhd \Gamma, A \supset B^{\mu} \longrightarrow A^{\nu}, \Delta \qquad \Sigma \rhd \Gamma, A \supset B^{\mu}, B^{\mu\circ\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, A \supset B^{\mu} \longrightarrow \Delta} \supset\rightarrow$$

$$\frac{\Sigma, a : \alpha \rhd \Gamma \longrightarrow A[a/x]^{\mu\circ\alpha}, \forall x A^{\mu}, \Delta}{\Sigma \rhd \Gamma \longrightarrow \forall x A^{\mu}, \Delta} \rightarrow\forall,\ a,\ \alpha \text{ new}$$

$$\frac{\Sigma \rhd \Gamma, \forall x A^{\mu}, A[t/x]^{\mu\circ\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, \forall x A^{\mu} \longrightarrow \Delta} \forall\rightarrow,\ t\ \Sigma\text{-term of index } \nu$$

This calculus also describes intuitionistic proofs. Because the new ($\rightarrow\supset$) and ($\rightarrow\forall$) rules encode a transition from scope σ to scope $\sigma \circ \alpha$ on their right-side formulas, the rules continue to describe the scoped location at which α is introduced. This allows the representation of this scope to be abbreviated to α on the left-side formula of the ($\rightarrow\supset$) rule, and in the indexing of the new term a of the ($\rightarrow\forall$) rule. Because of this abbreviation, we call this system LMA.

The arguments presented previously can be easily adapted to establish that LMA is a correct calculus for the syntactic representation of scope in intuitionistic proofs. We can show LMA complete by the argument of Theorems 1 and 3. For, LMA retains the ($\supset\rightarrow$) and ($\forall\rightarrow$) rules that allow formulas to be labeled with the current scope as structural scope is made explicit; as the argument requires, the labels of left formulas remain subsets of the labels of right formulas, even with the new right rules.

We can show LMA sound by transforming its labels to match the labels of LMS. This transformation simply folds out the abbreviation undertaken at ($\rightarrow\supset$) and ($\rightarrow\forall$) rules.

Theorem 4. *Every LMA derivation can be transformed into an LMS derivation, by a change in labeling.*

Proof. The transformation involves a partial mapping δ taking atomic annotation symbols α to sets of annotation symbols $\delta(\alpha)$. The translation $\delta(\mu)$ of an LMA annotation term μ will be (the $\circ$-concatenation of) the union of $\delta(\alpha)$ for all α in μ. We ensure that $\delta(\alpha) = \delta(\delta(\alpha))$ and $\alpha \in \delta(\alpha)$.

We translate an LMA sequent using such a map δ to relabel left formulas and indexing contexts; Γ^{δ} abbreviates the multiset of formulas with an occurrence of $A^{\delta(\mu)}$ for each occurrence of A^{μ} in Γ, and similarly Σ^{δ}. We use a function η to relabel right formulas: For each occurrence A^{μ} on the right $\eta(A^{\mu})$ is a formula A^{ν} with $\delta(\mu) \subseteq \nu$; Δ^{η} denotes the action of η on the formulas in Δ. Given any such δ and η, and an LMA derivation with end sequent $\Sigma \rhd \Gamma \longrightarrow \Delta$, we can construct an LMS derivation of $\Sigma^{\delta} \rhd \Gamma^{\delta} \longrightarrow \Delta^{\eta}$. The proof is by induction on the structure of LMA derivations.

The conditions on η and δ preserve the correctness of axiom and substitution rules because if $\mu \subseteq \nu$ then $\eta(\mu) \subseteq \eta(\nu)$ and if also $\delta(A^{\nu}) = A^{\nu'}$ then $\eta(\mu) \subseteq \nu'$. The remainder of the construction is to derive a labeling of the side formulas of the inference rules so as to satisfy LMS figures and allow the induction hypothesis to be applied.

For $(\rightarrow\supset)$ and $(\rightarrow\forall)$, this goes as follows. The inference introduces an annotation eigenvariable α on a principal formula whose revised annotation is μ. We relabel the subderivation by η' where $\eta'(\alpha) = \mu \circ \alpha$ and otherwise η' agrees with η (assuming without loss of generality because α is new that $\eta(\alpha)$ is undefined). For the new right-side formula $A^{\nu\circ\alpha}$ we set $\delta(A^{\nu\circ\alpha}) = A^{\mu\circ\alpha}$. The result instantiates LMS figures.

For $(\supset\rightarrow)$ and $(\forall\rightarrow)$, the inference involves annotations μ, ν and $\mu \circ \nu$. In both cases, we extend δ to relabel the side formula $B^{\mu\circ\nu}$ as $B^{\eta(\mu\circ\nu)}$. To complete the relabeling, for $(\supset\rightarrow)$ we can extend δ to δ' in the left subderivation so that the side formula A^{ν} gets $\delta'(A^{\nu}) = A^{\eta(\mu\circ\nu)}$ in place of ν. Meanwhile, for $(\forall\rightarrow)$, we simply observe that a Σ-term of index ν must be a Σ^{η}-term of index $\eta(\nu)$ and hence index $\eta(\mu \circ \nu)$.

The correspondence for remaining inference rules is immediate. □

LMA represents another take on topological semantics. The LMA implication rule corresponds to the model-theoretic condition that $A \supset B$ is true at an open set q exactly when B is true at $q \cap r$ for any open set r where A is true. This is yet another equivalent to the conditions on implication presented earlier.

D'Agostino and Gabbay [11] describe a labeled deductive system for propositional intuitionistic logic without disjunction whose labeling matches LMA. They motivate this system neither as an abbreviated representations of paths through proofs nor as a translation method for topological models, but rather as a direct encoding of intuitionistic sequent rules. For example, in $(\rightarrow\supset)$, μ represents the sequent from which $A \supset B$ is to be proved, and then $\mu \circ \alpha$ represents the sequent from which B is to be proved. The assumption of A at α indicates that α represents the formula A. In fact, d'Agostino and Gabbay observe that this intuition allows the eigenvariable condition on $(\rightarrow\supset)$ to be simplified. All assumptions of A are identical, so it suffices to have a single *A-characteristic* atomic label α used exactly when A is assumed.

We have just seen one way to extend the labeling of [11] to capture full intuitionistic logic and to obtain a proof system with the advantages for proof search of classical logic. Such an extension does not seem compatible with the intuition d'Agostino and Gabbay propose, however. Consider applying d'Agostino and Gabbay's intuition to disjunction. Encoding the basic LJ $(\vee \rightarrow)$ sequent rule directly might give something like this:

$$\frac{\Sigma \rhd \Gamma, A \vee B^{\nu}, A^{\alpha} \longrightarrow C^{\mu\circ\alpha}, C^{\mu}, \Delta \qquad \Sigma \rhd \Gamma, A \vee B^{\nu}, B^{\beta} \longrightarrow C^{\mu\circ\beta}, C^{\mu}, \Delta}{\Sigma \rhd \Gamma, A \vee B^{\nu} \longrightarrow C^{\mu}, \Delta} \vee \rightarrow$$

Its application would be subject to conditions that ν be a subset of μ, that α be A-characteristic, and that β be B-characteristic. Because the rule affects multiple formulas in the end sequent simultaneously, it is obviously a significant departure from ordinary sequent-calculus with an uncertain impact on proof search.

On the other hand, we might attempt to exploit classical reasoning for disjunction, as in LMM (and LMA):

$$\frac{\Sigma \rhd \Gamma, A \vee B^{\nu}, A^{\nu} \longrightarrow \Delta \qquad \Sigma \rhd \Gamma, A \vee B^{\nu}, B^{\nu} \longrightarrow \Delta}{\Sigma \rhd \Gamma, A \vee B^{\nu} \longrightarrow \Delta} \vee \longrightarrow$$

This actually yields incorrect results if we only require atomic labels of assumptions to be characteristic of those assumptions. For example, $A \supset B \vee C$ does not entail $(A \supset B) \vee (A \supset C)$ intuitionistically. Yet we have the following labeled proof in which α is A-characteristic:

$$\cfrac{\cfrac{\cfrac{\cfrac{\ldots, A^{\alpha} \longrightarrow A^{\alpha}, \ldots \qquad \cfrac{\ldots, B^{\alpha} \longrightarrow B^{\alpha}, C^{\alpha}, \ldots \qquad \ldots, C^{\alpha} \longrightarrow B^{\alpha}, C^{\alpha}, \ldots}{\ldots, B \vee C^{\alpha} \longrightarrow B^{\alpha}, C^{\alpha}, \ldots} \vee \longrightarrow}{A^{\alpha}, A^{\alpha}, A \supset B \vee C \longrightarrow B^{\alpha}, C^{\alpha}, \ldots} \supset \longrightarrow}{A^{\alpha}, A \supset B \vee C \longrightarrow B^{\alpha}, A \supset C, \ldots} \longrightarrow \supset}{A \supset B \vee C \longrightarrow A \supset B, A \supset C, \ldots} \longrightarrow \supset}{A \supset B \vee C \longrightarrow (A \supset B) \vee (A \supset C)} \longrightarrow \vee \tag{10}$$

Thus, it is crucial for extending the proposal of [11] that assumptions have fresh labels regardless of the content of those assumptions. This underscores that the proof-theoretic meaning of labels is to record the identity of the inferences at which assumptions are made. This proof-theoretic meaning underlies the simple syntactic proof of correctness for LMA building on those for LMS and LMP. This contrasts with the infinitary and nonconstructive proof of correctness in [11], which is ultimately semantic in nature.

4.3. *λ-terms as set annotations*

Bittel [6, 7] defines a proof-theoretic method for eliminating impermutabilities in intuitionistic logic which uses labels inspired by λ-terms. Bittel's proposal encodes assumptions from which a conclusion is derived by the free variables in the λ-term that labels the conclusion. Thus, for example, Bittel's rule for $(\longrightarrow\supset)$ can be written as the following sequent rule:

$$\frac{\Gamma, x : A \longrightarrow M : B, \Delta}{\Gamma \longrightarrow \lambda x.M : A \supset B, \Delta} \longrightarrow \supset$$

The rule has a side condition that x not appear free in the end sequent.

We can distill the mechanism behind Bittel's system by representing only the free variables and discarding the remainder of the λ-term. This reveals quite a close connection with LMA. If the set of free variables of $\lambda x.M$ is μ, and the set of free variables of x is α, then the set of free variables of M is $\mu \circ \alpha$ – provided x occurs in M. More precisely, if the set of free variables of M is some set v, then the set of free variables of $\lambda x.M$ is $v - \alpha$. So Bittel's $(\longrightarrow\supset)$ rule can be reconstructed:

$$\frac{\Gamma, A^{\alpha} \longrightarrow B^{v}, \Delta}{\Gamma \longrightarrow A \supset B^{v-\alpha}, \Delta} \longrightarrow\supset,\ \alpha \text{ new}$$

The difference between Bittel's system and LMA lies in the fact that Bittel's system is designed to represent *exactly* the assumptions from which a conclusion is derived. LMA allows a formula to be labeled with the names of assumptions that do not directly contribute to its proof. This difference is visible already in the $(\longrightarrow\supset)$ rule; it recurs in Bittel's axiom rule and in the other right rules of Bittel's system. Bittel's axiom identifies the labels of assumption and conclusion:

$$\Sigma \rhd \Gamma, A^{\mu} \longrightarrow A^{\mu}, \Delta$$

Other right rules are modified to label their principal formula with the union of the annotations of their side formulas. For example, for $(\longrightarrow\wedge)$, we need:

$$\frac{\Sigma \rhd \Gamma \longrightarrow A^{\mu}, A \wedge B^{\mu\circ v}, \Delta \qquad \Sigma \rhd \Gamma \longrightarrow B^{v}, A \wedge B^{\mu\circ v}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A \wedge B^{\mu\circ v}, \Delta} \longrightarrow \wedge$$

For technical reasons, an explicit rule of contraction on the right is also required.

$$\frac{\Sigma \rhd \Gamma \longrightarrow A^{\mu}, A^{v}, \Delta}{\Sigma \rhd \Gamma \longrightarrow A^{\mu\circ v}, \Delta} C$$

(In the λ-calculus, this rule corresponds to a new term constructor for implicit case analysis.) Let LMT denote the system with the left rules of LMA plus right and axiom rules modified in this way; the T records the original status of annotations as terms.

To show LMT sound, we can give an inductive construction. We start with an LMM proof with end sequent $\Gamma \longrightarrow \Delta$, a function η labeling formula occurrences in Γ, and an indexing context Σ. The construction produces a labeling η' of formula occurrences in Δ and a proof in LMT of $\Sigma \rhd \Gamma^{\eta} \longrightarrow \Delta^{\eta'}$. At axioms, η' sends the right linked formula A to the η image of its left match, and sends the remaining formulas to the empty set. The case of $(\longrightarrow\wedge)$ is representative of inductive steps in this construction: we apply the induction hypothesis to obtain proofs of

$$\Sigma \rhd \Gamma^{\eta} \longrightarrow A^{\mu}, A \wedge B^{\mu'}, \Delta^{\delta} \quad \text{and} \quad \Sigma \rhd \Gamma^{\eta} \longrightarrow A^{v}, A \wedge B^{v'}, \Delta^{\delta'}$$

By weakening these derivations and applying the contraction rule as necessary, we can obtain derivations of

$$\Sigma \rhd \Gamma^{\eta} \longrightarrow A^{\mu}, A \wedge B^{\mu\circ v}, \Delta^{\eta'} \quad \text{and} \quad \Sigma \rhd \Gamma^{\eta} \longrightarrow A^{v}, A \wedge B^{\mu\circ v}, \Delta^{\eta'}$$

where $\eta'(C) = \delta(C) \circ \delta'(C)$ except for the principal and side formulas of the $(\longrightarrow\wedge)$ inference. These derivations can be composed using the LMT $(\longrightarrow\wedge)$ inference.

To show this complete, we can transform its labeling into the labeling of LMA. We need only change the labels of right formulas; we inductively associate each right formula A^{μ} with the appropriate new annotation v with $\mu \subseteq v$, in the obvious way.

4.4. Intuitionistic proof search: some comparisons

In Section 1.3, we motivated one difficulty in intuitionistic proof search, caused by the need to order rule applications to reflect scope. We have now seen several syntactic methods that recast proof rules so as to reduce the impact of order using explicit labeling of scope. In Section 5.1 we shall see how to extend explicitly scoped systems using Herbrand terms and unification so as to eliminate the impact of order altogether. This represents a solution to a major problem in automatic derivation of intuitionistic proofs. It is far from the only problematic feature of intuitionistic proof search, however. Related work that addresses these other problems must still be adapted to the present framework. We sketch the issues involved in this section.

The first problem concerns bounding the size of proofs. Propositional intuitionistic logic can be shown decidable, and decision algorithms for it devised, using such bounds. In [14], Dyckhoff presents a sequent calculus from which these bounds follow naturally; this system limits the number of times a formula is decomposed along each path in a structurally scoped proof to one, by eliminating contraction and avoiding preservation of principal formulas of inferences. As the following argument suggests, what underlies Dyckhoff's results is the fact that

$$A \text{ intuitionistically entails } B \supset C \equiv (A \supset B) \supset C. \qquad (11)$$

In any one intuitionistic scope, as in classical propositional logic, there is nothing to be gained from decomposing a formula more than once. This in itself does not guarantee that intuitionistic propositional proofs have bounded size, however, because premises of the form $(A \supset B) \supset C$ can be instantiated in any scope μ to create a new scope $\mu\alpha$. Fact (11) says that this is necessary only once in each scoped path: since A holds at scope $\mu\alpha$, then for every scope $\mu\alpha\nu$, $A \supset B$ is true there exactly if B is true there. Dyckhoff's calculus encodes this directly using a structural discipline of scope. Articulated explicitly as here, it should also be possible to incorporate this constraint into an explicitly scoped system, and thereby obtain an explicitly scoped decision procedure for intuitionistic logic. The explicitly scoped system would retain a possibility of goal-oriented proof search and hence a possibility of faster failure than proof search in Dyckhoff's calculus.

A second problem is to ensure that the calculus gives proofs a compact form. Sequent calculus proofs without cut are often required to include redundant subtrees. This point is emphasized in [10] where it is shown that cut-free classical propositional sequent proofs cannot polynomially simulate truth tables. These redundancies are addressed by matrix methods of proof [3, 5] and tableaux with analytic cuts [12]. Labeled methods have been extended to these frameworks in [11, 50] – but on a semantic rather than a proof-theoretic basis. Because cut-elimination can be proved purely syntactically, we can now anticipate finding a proof-theoretic basis for these techniques, in light of the present work, and then adapting these techniques where appropriate to synthesize proofs in intuitionistic natural deduction and analyze intuitionistic proof search.

A third problem, given that inferences can now be applied in any order, is to select a perspicuous or efficient order for applying them. Algorithms for proof search with particular orders of inference can be given as refinements of the full sequent calculi. For example, in [23], a restriction on uses of ($\supset\rightarrow$) is exploited to obtain a bijection between normal simply typed λ-terms and sequent proofs. Logic programming, meanwhile, can be presented in terms of sequent calculi in which right rules must be applied before left rules whenever possible [32]. This order of rule application is embodied in restricted *focusing* sequent calculi in [2, 33]; the proofs obtained are called *uniform*. These disciplines for restricting proof search have so far been formulated in structurally scoped sequent calculi; thus while they can reduce branching in proof search, they cannot by themselves enable proof search for full intuitionistic logic to proceed in a goal-directed manner as in an explicitly scoped calculus. The intuitions behind these refinements remain applicable in explicitly scoped systems, however. For example, we return to the question of the application of these techniques to explicitly scoped calculi for intuitionistic logic in Section 5.1.

There remains the problem of selecting the right discipline of explicit scope for a particular application, from the four we have seen: LMP, LMS, LMA, LMT. One ground for comparison is the complexity of reasoning with the appropriate equational theory of terms. Bittel's LMT might seem best on these grounds: Bittel shows that LMT can be implemented using annotations with free structure and ordinary unification. This implementation is problematic, however, in that it prevents the system from lifting compactly to use unification as do the others. In fact, Bittel uses a lifting procedure that does not eliminate quantifier impermutabilities but allows substitutions to be made appropriate to the particular order in which quantifier rules are used – a strategy analogous to [49]. Meanwhile, for the strings of LMP, the situation is in fact much better than it might appear. Stone [43] shows that the string equations resulting from an LMP proof can be solved in polynomial time using a constraint algorithm that avoids the need to backtrack among alternative equational unifiers. There is thus some reason to think that LMP is not only the most basic system, but also the most efficient one.

5. Proof-theoretic extensions and applications

In Sections 3 and 4, we considered a variety of systems that allow intuitionistic proofs to be constructed in a more liberal order than a typical, structurally scoped calculus. We motivated the need for such systems in Section 1 with two applications that depend on the scope discipline of intuitionistic proofs: the analysis of automatic proof search for logic programming and the automatic synthesis of functional programs. This section returns to these applications and establishes their connection to the results of Sections 3 and 4, using some additional proof-theoretic results. We begin in Section 5.1 by presenting a Herbrand theorem for LMP and sketching its relevance for logic programming. We continue in Section 5.2 with an algorithm to extract λ-terms from lifted LMP deductions, and sketch its role in program synthesis.

5.1. Lifting LMP

Recall from Section 3.3 that LMP represents only a halfway point in the development of a calculus in which rules may appear in any order. LMP allows propositional rules that could occur low in a structurally scoped proof to be delayed to a higher point; but it does not always allow propositional rules that could occur high in a structurally scoped proof to be advanced to a lower point – this was illustrated in proofs (8) and (9). This difference is a result of the eigenvariable condition on annotations and first-order terms imposed by $(\rightarrow\supset)$, $(\rightarrow\forall)$ and $(\exists\rightarrow)$ figures.

But in Section 2.2, we have already observed that there is a general syntactic device for eliminating these eigenvariable conditions: the use of Herbrand terms in place of eigenvariables [29]. Herbrand terms are representations of eigenvariables whose constituency, not position in the proof, specifies an appropriate order in which eigenvariables are to be introduced. At the same time as it eliminates the quantifier impermutabilities in a calculus, the use of Herbrand terms allows instantiations of quantifiers to be delayed until sufficient information becomes available. A variable is used in place of a substituted first-order term and its value is determined using unification.

This section develops and applies a sequent calculus refinement of LMP that uses Herbrand terms and unification. We call this new calculus LMU. The calculus implements dynamic Skolemization in the style of [29]. Dynamic Skolemization annotates formulas in proofs with the information needed to construct any Herbrand terms when quantifier-like rules apply, instead of rewriting formulas to a special functional form containing Herbrand terms before proof search begins. Dynamic Skolemization is appropriate because any functional form for intuitionistic logic would have to go beyond the ordinary syntax of formulas – for example to encode the intuitionistic difference between $A\supset\exists xB(x)$ and $\exists x(A\supset B(x))$. (But see [18] for one way to do this.)

In our notation for dynamic Skolemization, each formula is subscripted by a list of terms H that must occur in the sequent before any inference could decompose that formula. The list of terms is maintained so as to include the instantiations made in deriving the formula (since it would be impossible to arrive at the formula without making those instantiations). This suffices for LMP because it has only eigenvariable impermutabilities. However, in general, the list must also be updated to include additional terms based on the propositional impermutabilities of the logic. For example, [29] describes the additional bookkeeping required to maintain these terms for first-order linear logic [22] with its panoply of impermutabilities [2, 21, 44].

To maintain the list H, rules that formerly involved a free choice of terms are revised. Logic variables are substituted for bound variables, and a record of the use of the logic variable is made by appending it to the label H on the formula. The values of logic variables are later constrained at axioms by unification. For example, we have for $(\supset\rightarrow)$:

$$\frac{\ldots\rhd\Gamma,A\supset B^{\mu}_{H}\longrightarrow A^{\mu x}_{H,x},\Delta\qquad\ldots\rhd\Gamma,A\supset B^{\mu}_{H},B^{\mu x}_{H,x},\longrightarrow\Delta}{\ldots\rhd\Gamma,A\supset B^{\mu}_{H}\longrightarrow\Delta}\supset\rightarrow$$

(The ellipses anticipate further revisions required under dynamic Skolemization.)

Having maintained the list H, dynamic Skolemization reformulates the remaining quantifier-like rules so as to build an appropriate term. This is done by combining this list H with a name for the principal connective of the inference. The name, which we indicate by a subscript, is viewed as a function symbol and H as its argument. Thus, the action of $(\longrightarrow\supset)$ in LMU can be schematized thus:

$$\frac{\ldots \rhd \Gamma, A_H^{\mu(gH)} \longrightarrow B_H^{\mu(gH)}, A \supset_g B_H^{\mu}, \Delta}{\ldots \rhd \Gamma \longrightarrow A \supset_g B_H^{\mu}, \Delta} \longrightarrow \supset$$

The function symbol associated with the implication is g, the list of instantiations H, and the new Herbrand term representing an arbitrary transition of accessibility is gH. The use of a name associated with the symbol is a slight departure from [29]; they use a unique name for each rule application. The difference involves adding to their proof a simple step to eliminate redundant eigenvariable introductions by exploiting the preservation of principal and side formulas on sequents in LMP.

Again following [29], we make the role of unification in assigning values to logic variables an explicit part of the sequent calculus. Now, given the use of Herbrand terms to eliminate scope, the lifetime of a variable can extend beyond the subproof where it is introduced. Each sequent therefore includes an input substitution U, which encodes the constraints in force and the unifications performed up to the point in proof search where the sequent arises; and an output substitution V, which encodes the constraints in force and the unifications performed up to the point in proof search where the proof of that sequent has been completed. Effectively, V specializes U so as to respect the constraints imposed by axioms in the proof of the sequent.

A distinction between input and output indexing contexts on sequents is required for the same reason. Each sequent includes a input specification Σ of the scopes of terms already introduced when the sequent is first encountered in proof search, and an output indexing Θ that specifies not only these scopes but also the scopes of terms introduced as part of proving the sequent. Overall, sequents are written

$$U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow \Delta$$

to reflect their dependence on inputs U and Σ and production of outputs V and Θ.

As an illustration of how these new labels of sequents combine with dynamic Skolemization, we can give the example of $(\exists\longrightarrow)$ and $(\longrightarrow\exists)$ in full. First $(\exists\longrightarrow)$:

$$\frac{U; \Sigma, gH : \mu / V; \Theta \rhd \Gamma, \exists_g x A_H^{\mu}, A[gH/x]_H^{\mu} \longrightarrow \Delta}{U; \Sigma / V; \Theta \rhd \Gamma, \exists_g x A_H^{\mu} \longrightarrow \Delta} \exists\longrightarrow$$

$(\exists\longrightarrow)$ introduces a new Herbrand term gH determined from the list of instantiations H and the name g of the quantifier occurrence. This new Herbrand term is confined to the scope μ with which the principal formula is labeled; the input indexing to the subderivation therefore includes the specification $gH : \mu$.

Second, $(\rightarrow \exists)$:

$$\frac{U;\Sigma,y:\mu/V;\Theta \rhd \Gamma \longrightarrow A[y/x]^{\mu}_{H,y}, \exists x A^{\mu}_{H}, \Delta}{U;\Sigma/V;\Theta \rhd \Gamma \longrightarrow \exists x A^{\mu}_{H}, \Delta} \rightarrow \exists$$

Here, y is a fresh logic variable. Note that because unification determines the value of y, it is the unification step and not the $(\rightarrow \exists)$ rule that must determine whether the value assigned to y respects the scope μ of the formula where y is introduced. Accordingly, the rule adds an indexing $y:\mu$ to Σ.

This leaves only the revision of initial sequents left to be explained. We have

$$U;\Sigma/V;\Sigma \rhd \Gamma, B^{\mu}_{H} \longrightarrow A^{\nu}_{I}, \Delta$$

where V is any most general unifier more specific than U and having the following properties:
1. $V(A)$ and $V(B)$ are identical as terms;
2. $V(\nu)$ is identical as a string to $V(\mu x)$ for some fresh logic variable x; and
3. for any term variable y – where Σ assigns σ to y, and some Herbrand function-application hY is a subterm of $V(y)$ – the following holds: for any term hZ – associated by Σ with some annotation ρ – such that $V(hZ) = V(hY)$, $V(\rho)$ is a prefix of $V(\sigma)$.

Computation of V calls for string unification; nevertheless annotation equations are sufficiently simple that the existence of a solution to a set of annotation equations in polynomial time for many search strategies [43].

LMU is summarized in Fig. 5. The construction of LMU instantiates the general procedure for the construction of optimized sequent calculi described in [29]. The proof of correctness given in [29] applies immediately to LMU, once we establish the correctness of implementing the indexing check by enforcing condition (3) on substitutions. To establish this, we first observe that Lincoln and Shankar's correspondence between proofs in the optimized calculus and proofs in the ground calculus is quite close. In particular, the introductions of Herbrand function applications equal to hY under the output substitution V of an LMU proof correspond to some rule applications introducing the eigenvariable a in an LMP proof. Meanwhile, the assignment of t to y by V in the optimized proof indicates that the term corresponding to t should be substituted at the rule application in the ground proof corresponding to the introduction of y.

Because of this regularity, condition (3) ensures that the term substituted in the ground proof at the site corresponding to y is in fact a term of appropriate index. This is just because Herbrand function applications appear in $V(y)$ at the positions corresponding to any eigenvariables that appear in the ground term. Now, Herbrand function applications may appear in $V(y)$ *without* corresponding to eigenvariables: they may appear nested inside arguments to other Herbrand function applications. However, these additional checks will not rule out any unifiers of appropriate index. For suppose Σ contains $hY:\rho$ and $gZ:\varsigma$ where $V(gZ)$ is a subterm of $V(hY)$. $V(u)$ must contain $V(gZ)$ for some variable u free on the formula from which h is introduced. By condition (3),

$$U; \Sigma / MGU(MGU(U, A, B), \mu x, \nu); \Sigma \rhd \Gamma, B^{\mu}_{H} \longrightarrow A^{\nu}_{I}, \Delta^{*}$$

$$\frac{U; \Sigma / V; \Theta \rhd \Gamma, A \wedge B^{\mu}_{H}, A^{\mu}_{H}, B^{\mu}_{H} \longrightarrow \Delta}{U; \Sigma / V; \Theta \rhd \Gamma, A \wedge B^{\mu}_{H} \longrightarrow \Delta} \wedge\rightarrow$$

$$\frac{U; \Sigma / W; \Phi \rhd \Gamma \longrightarrow A^{\mu}_{H}, A \wedge B^{\mu}_{H}, \Delta \qquad W; \Phi / V; \Theta \rhd \Gamma \longrightarrow B^{\mu}_{H}, A \wedge B^{\mu}_{H}, \Delta}{U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow A \wedge B^{\mu}_{H}, \Delta} \rightarrow\wedge$$

$$\frac{U; \Sigma / W; \Phi \rhd \Gamma, A \vee B^{\mu}_{H}, A^{\mu}_{H} \longrightarrow \Delta \qquad W; \Phi / V; \Theta \rhd \Gamma, A \vee B^{\mu}_{H}, B^{\mu}_{H} \longrightarrow \Delta}{U; \Sigma / V; \Theta \rhd \Gamma, A \vee B^{\mu}_{H} \longrightarrow \Delta} \vee\rightarrow$$

$$\frac{U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow A^{\mu}_{H}, B^{\mu}_{H}, A \vee B^{\mu}_{H}, \Delta}{U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow A \vee B^{\mu}_{H}, \Delta} \rightarrow\vee$$

$$\frac{U; \Sigma / W; \Phi \rhd \Gamma, A \supset B^{\mu}_{H} \longrightarrow A^{\mu x}_{H,x}, \Delta \qquad W; \Phi / V; \Theta \rhd \Gamma, A \supset B^{\mu}_{H}, B^{\mu x}_{H,x} \longrightarrow \Delta}{U; \Sigma / V; \Theta \rhd \Gamma, A \supset B^{\mu}_{H} \longrightarrow \Delta} \supset\rightarrow$$

$$\frac{U; \Sigma / V; \Theta \rhd \Gamma, A^{\mu(gH)}_{H} \longrightarrow B^{\mu(gH)}_{H}, A \supset_{g} B^{\mu}_{H}, \Delta}{U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow A \supset_{g} B^{\mu}_{H}, \Delta} \rightarrow\supset$$

$$\frac{U; \Sigma, y : \mu z / V; \Theta \rhd \Gamma, \forall x A^{\mu}_{H}, A[y/x]^{\mu z}_{H,y,z} \longrightarrow \Delta}{U; \Sigma / V; \Theta \rhd \Gamma, \forall x A^{\mu}_{H} \longrightarrow \Delta} \forall\rightarrow$$

$$\frac{U; \Sigma, gH : \mu(hH) / V; \Theta \rhd \Gamma \longrightarrow A[gH/x]^{\mu(hH)}_{H}, \forall_{gh} x A^{\mu}_{H}, \Delta}{U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow \forall_{gh} x A^{\mu}_{H}, \Delta} \rightarrow\forall$$

$$\frac{U; \Sigma, gH : \mu / V; \Theta \rhd \Gamma, \exists_{g} x A^{\mu}_{H}, A[gH/x]^{\mu}_{H} \longrightarrow \Delta}{U; \Sigma / V; \Theta \rhd \Gamma, \exists_{g} x A^{\mu}_{H} \longrightarrow \Delta} \exists\rightarrow$$

$$\frac{U; \Sigma, y : \mu / V; \Theta \rhd \Gamma \longrightarrow A[y/x]^{\mu}_{H,y}, \exists x A^{\mu}_{H}, \Delta}{U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow \exists x A^{\mu}_{H}, \Delta} \rightarrow\exists$$

Fig. 5. LMU, a unification-based presentation of LMP. The only proviso is (*) that *MGU* must supply a unifier (with occurs check) at axioms that assigns labels to quantifiers and terms in accordance with Σ.

the annotation $V(\varsigma)$ is a prefix of the annotation $V(\mu)$ associated with u. But the logical rules ensure that μ is a prefix of ρ. Hence, $V(\varsigma)$ must be a prefix of $V(\rho)$.

Theorem 5 (Correctness). *Let Γ and Δ be any multisets of labeled formulas which do not contain Herbrand functions or variables. Then there is a deduction in LMP of*

$$\Sigma \rhd \Gamma \longrightarrow \Delta$$

if and only if there is a deduction in LMU of

$$U; \Sigma / V; \Theta \rhd \Gamma \longrightarrow \Delta$$

for some U, V, and Θ.

Proof (*As outlined in Lincoln and Shankar* [29, pp. 284–287]). The *only if* direction is established by an induction on proofs in the ground system (in this case, LMP) which shows that the rule ordering of the ground proof and the substitutions made in the ground proof describe an analogous proof in the lifted system (in this case, LMU). The *if* direction is established by showing that the bindings of values to variables, any propositional impermutabilities inherited from the ground system (LMU has none), a condition that formulas precede their subformulas, and transitivity will induce a strict partial order on the rule applications in any proof in the lifted system. An induction on the structure of lifted proofs establishes that permutations of inferences convert any lifted proof to another proof in which the ordering of rule applications matches this induced partial order.

Substituting eigenvariables for Herbrand terms and appropriate values for logic variables gives a new proof where eigenvariable conditions are satisfied – with one exception due to the use of names for symbols rather than inferences in LMU. We may have cases where one $(\longrightarrow\supset)$, $(\longrightarrow\forall)$ or $(\exists\longrightarrow)$ rule introduces the same variable as a lower one (such occurrences are unordered by Lincoln and Shankar's conditions). Such cases are dispatched as follows. The principal and side formulas of the two rule applications must be identical. Because of the preservation of formulas in sequents in LMU and LMP, the side formulas of the lower application are available in the sequent when the higher rule applies. Therefore, we can exploit the admissibility of contraction and simply eliminate the higher inference. □

LMU has completely free permutabilities; reasoning about the order of introduction of quantifiers and implications is factored into the occur-check in unification. To illustrate this, we return to the LMP proof of $(C\supset B\vee A)\supset(C\supset A\vee B)$ given in (9). The corresponding proof in LMU is obtained simply by using variables in place of ground instantiations at $(\supset\longrightarrow)$ – assuming a labeling $(C\supset B\vee A)\supset_\alpha(C\supset_\beta A\vee B)$. It appears in (12).

$$\dfrac{\dfrac{\dfrac{;/\vartheta;\,\rhd\ldots,C^{\alpha\beta}\longrightarrow C^{\alpha x}\qquad \dfrac{\dfrac{\vartheta;/\vartheta;\,\rhd\ldots,A^{\alpha x}\longrightarrow A^{\alpha\beta}}{\vartheta;/\vartheta;\,\rhd\ldots,A^{\alpha x}\longrightarrow A\vee B^{\alpha\beta}}\longrightarrow\vee\qquad\dfrac{\vartheta;/\vartheta;\,\rhd\ldots,B^{\alpha x}\longrightarrow B^{\alpha\beta}}{\vartheta;/\vartheta;\,\rhd\ldots,B^{\alpha x}\longrightarrow A\vee B^{\alpha\beta}}\longrightarrow\vee}{\vartheta;/\vartheta;\,\rhd\ldots,B\vee A^{\alpha x}\longrightarrow A\vee B^{\alpha\beta}}\vee\longrightarrow}{;/\vartheta;\,\rhd C\supset B\vee A^{\alpha},C^{\alpha\beta}\longrightarrow A\vee B^{\alpha\beta}}\supset\longrightarrow}{\dfrac{;/\vartheta;\,\rhd C\supset B\vee A^{\alpha}\longrightarrow C\supset A\vee B^{\alpha}}{;/\vartheta;\,\rhd\longrightarrow(C\supset B\vee A)\supset(C\supset A\vee B)}\longrightarrow\supset}\longrightarrow\supset \tag{12}$$

Here ϑ is a substitution of β for x. Proof (12) has much the same form as (9), but proof (12) represents the fact that $(\supset\longrightarrow)$ outscopes $(\longrightarrow\supset)$ in the binding of x to β, not in the structure of the proof. Thus, a permuted proof in which the $(\supset\longrightarrow)$ rule applies lower, as in (13), is also possible.

$$\dfrac{\dfrac{\dfrac{;/\vartheta;\rhd\ldots C^{\alpha\beta}\longrightarrow C^{\alpha x},\ldots}{;/\vartheta;\rhd\ldots\longrightarrow C^{\alpha x},C\supset A\vee B^{\alpha},\ldots}\longrightarrow\supset\qquad\dfrac{\dfrac{\dfrac{\vartheta;/\vartheta;\rhd\ldots B^{\alpha x}\longrightarrow B^{\alpha\beta}\ldots}{\vartheta;/\vartheta;\rhd\ldots B^{\alpha x}\longrightarrow A\vee B^{\alpha\beta}\ldots}\longrightarrow\vee}{\vartheta;/\vartheta;\rhd\ldots B^{\alpha x}\longrightarrow C\supset A\vee B^{\alpha}\ldots}\longrightarrow\supset\qquad\dfrac{\dfrac{\vartheta;/\vartheta;\rhd\ldots A^{\alpha x}\longrightarrow A^{\alpha\beta}\ldots}{\vartheta;/\vartheta;\rhd\ldots A^{\alpha x}\longrightarrow A\vee B^{\alpha\beta}\ldots}\longrightarrow\vee}{\vartheta;/\vartheta;\rhd\ldots A^{\alpha x}\longrightarrow C\supset A\vee B^{\alpha}\ldots}\longrightarrow\supset}{\vartheta;/\vartheta;\rhd\ldots B\vee A^{\alpha x}\longrightarrow C\supset A\vee B^{\alpha}\ldots}\vee\longrightarrow}{/\vartheta;\rhd C\supset B\vee A^{\alpha}\longrightarrow C\supset A\vee B^{\alpha}\ldots}\supset\longrightarrow}{/\vartheta;\rhd\longrightarrow(C\supset B\vee A)\supset(C\supset A\vee B)}\longrightarrow\supset \tag{13}$$

Because the inferences of LMU proofs may appear in any order, we can choose arbitrary regimes for ordering inferences in LMU proofs without sacrificing completeness. One possible regime, suggested in [32] and extended in [2, 33], is to apply left rules only when we are committed that no right formula will be the principal formula of a higher rule in the proof, and until then to apply right rules. This search strategy provides a general description of the behavior of an interpreter for a logic programming language. It allows connectives in right formulas to be viewed as instructions for search.

This construction applies to any sequent calculus with appropriate permutabilities, not just LMU. For example, we might also apply it after adapting the results of [11, 17, 35, 42, 50] to derive an explicitly scoped sequent calculus of intuitionistic provability by purely semantic methods. However, the syntactic analysis of the proofs obtained plays an important part of describing the logic programming language. For example, as observed in [31], LJ proofs gain a natural modular structure from how they ensure that the assumption of A can only contribute to the proof of B in proving $A \supset B$. We have already used syntactic methods to put LMU proofs in correspondence with LJ proofs. LMU can therefore be used immediately to extend the logical analysis of modules presented in [31] to other logical fragments.

5.2. Extracting λ-terms

As mentioned in Section 1.1, a major motivation for considering intuitionistic deduction is the automatic synthesis of functional programs [9, 30]. In the deductive approach to program synthesis, the input is a specification of the type of a function (including constraints on the relation between its argument and its result). The output is a function that provably has this type. Using the Curry–Howard isomorphism [24], the type of the function can be specified as a formula in intuitionistic logic and the resulting program can be extracted from the proof of the formula. This requires the derivation not only of intuitionistic theorems but also of intuitionistic natural deductions.

LMU's contribution to this research program is to offer advantageous search for proofs that correspond to intuitionistic natural deductions. Now, the proof of correctness of LMU gave a system for permuting LMU inferences first to LMP inferences and then to LMM inferences and finally to LJ inferences. Thus, performing these permutations on an LMP or LMU deduction already gives a way to extract λ-terms. This technique is rather unsatisfactory, however, in so far as the majority of the permutations dictated by the correctness proof will have no impact on the λ-term ultimately obtained. We now consider how to extract λ-terms directly from LMU proofs.

5.2.1. Motivation

Intuitively, a completed LMU proof specifies a collection of intuitionistic inferences labeled with the scope in which each is to be performed.

Thus far, the collection of inferences is represented only by the inferences that the proof contains. The first step in extracting a λ-term from an LMU proof is to make the inferences explicit. We will do this in the style of [15] by labeling formulas with terms

$$\dfrac{\dfrac{\dfrac{/\sigma \rhd \ldots A^{\alpha} \longrightarrow A^{x} \ldots \qquad \sigma/\sigma \rhd \ldots C^{\alpha} \longrightarrow C^{\alpha} \ldots}{/\sigma \rhd \ldots A \supset C,\, A^{\alpha} \longrightarrow C^{\alpha} \ldots} \qquad \dfrac{\sigma/\tau \rhd \ldots B^{\alpha} \longrightarrow B^{y} \ldots \qquad \tau/\tau \rhd \ldots C^{y} \longrightarrow C^{\alpha} \ldots}{\sigma/\tau \rhd \ldots B \supset C,\, B^{\alpha} \longrightarrow C^{\alpha} \ldots}}{/\tau \rhd A \supset C,\, B \supset C,\, (A \vee B)^{\alpha} \longrightarrow C^{\alpha} \ldots}}{/\tau \rhd A \supset C,\, B \supset C \longrightarrow A \vee B \supset C}$$

a. The LMU rendering of a usual intuitionistic logic proof; $\sigma = \alpha/x$; $\tau = \alpha/x, \alpha/y$.

$$\dfrac{\dfrac{\dfrac{\dfrac{/\sigma \rhd \ldots A^{\alpha} \longrightarrow A^{x} \ldots \qquad \sigma/\tau \rhd \ldots B^{\alpha} \longrightarrow B^{y} \ldots}{/\tau \rhd \ldots (A \vee B)^{\alpha} \longrightarrow A^{x},\, B^{y},\, C^{\alpha} \ldots}}{/\tau \rhd \ldots \longrightarrow A^{x},\, B^{x},\, (A \vee B \supset C)} \qquad \dfrac{\tau/\tau \rhd \ldots C^{x} \longrightarrow C^{\alpha} \ldots}{\tau/\tau \rhd \ldots C^{x} \longrightarrow B^{y},\, A \vee B \supset C}}{/\tau \rhd \ldots A \supset C \longrightarrow B^{y},\, A \vee B \supset C} \qquad \dfrac{\tau/\tau \rhd \ldots C^{y} \longrightarrow C^{\alpha} \ldots}{\tau/\tau \rhd \ldots C^{y} \longrightarrow A \vee B \supset C}}{/\tau \rhd A \supset C,\, B \supset C \longrightarrow A \vee B \supset C}$$

b. An LMU proof equal to the first up to permutations; $\sigma = \alpha/x$; $\tau = \alpha/x, \alpha/y$.

Fig. 6. Permutations motivate partiality and a different treatment of scope.

recording the inferences that derive them. In this presentation, formulas on the left may be labeled with complex proof-terms built by applying left rules; this contrasts with presentations such as that in [20] where left formulas are always labeled with variables, and substitutions are performed at the application of left rules. This allows each sequent to record thc inferences performed in each scope.

The second step is to assemble a λ-term from these scoped inferences. With structural scope, it is possible for this assembly to proceed incrementally in lock-step with the construction of the sequent proof. The structure of the proof matches the abstraction and substitution operations that need to be performed in assembling a λ-term (the specifications in [15, 20] do this). However, in LMU, scopes do not always correspond to regions of the proof. We therefore require a recursive traversal of the record of scoped inferences to assemble the final λ-term.

Bittel's method of extracting λ-terms also involves an incremental labeling and a postprocessing traversal [6, 7]. Bittel's traversal procedure is qualitatively quite similar to the one here. Our process, unlike his, exploits an independent discipline of explicit scope to streamline deduction and guide extraction. At the same time, we account for a greater range of ordering of rules and therefore need an additional mechanism to accumulate inferences during the construction of proofs.

We can illustrate the issues involved by the contrast between the two proofs of Fig. 6. If f names $A \supset C$ and g names $B \supset C$, these proofs both derive a function $\lambda u.case(u\ of\ inl(v) \Rightarrow f(v) | inr(v') \Rightarrow g(v'))$. In the proof of Fig. 6(a), this term matches the scoped structure of the proof. The lowest rule is $(\longrightarrow\supset)$ just as the widest scope connective is λ, the next rule is $(\vee \longrightarrow)$ just as the next connective is *case*, etc.

In LMU, it is more complicated. The $(\vee \longrightarrow)$ rule corresponding to the case statement lies at the upper left, while the $(\longrightarrow\supset)$ rule appears three times! The annotations define the scope of connectives: scope is no longer simply a reflection of the structure of the proof tree. To see that scope is still represented, observe that all subproofs contain the annotation α corresponding to the one λ in the resulting term. Because these scope-annotations propagate through unification during proof construction, the synthesis of *case* statements for disjunctions and *casex* statements for existential quantifiers from LMU proofs will be delayed until the proof is complete and the exact scope of connectives is determined.

This process requires a new mechanism for assembling proof-terms from separate subtrees of a proof. In the proof of Fig. 6(b), the right subproofs, even though combined by ($\supset\!\rightarrow$), each partially constrain the deduction associated with the conclusion. One is associated with the inference $inl(v) \Rightarrow f(v)$, the other with the inference $inr(v') \Rightarrow g(v')$. To handle this, the term associated with a formula must be regarded as a *partial* specification of the natural deduction proof of that formula. The need to assemble these partial specifications into complete ones reflects the implicit role of contraction in collapsing repetitions of formulas in sequents after permutations.

5.2.2. *Recording inferences*

A precise simultaneous specification of LMU and intuitionistic natural deduction is as follows. We begin here by describing how the inferences made in an LMU proof are recorded. We continue in Section 5.2.3 by describing how terms are reconstructed from this record.

We will use variables in terms as placeholders for content that cannot be determined until scopes are fixed. Since annotations determine scope in the proof and in the corresponding λ-term, we implement this through two maps that associate distinct variables with annotation eigenvariables. K_α abstracts the term that ultimately will describe the right-side formula of the rule introducing annotation α; v_α will describe the label of the left-side formula. Thus, *every* ($\rightarrow\!\supset$) rule introducing annotation α will be effectively assigned the anonymous proof term $\lambda v_\alpha . K_\alpha$. As described below, separate structures will determine what terms the proof says should actually correspond to these variables.

In ordinary natural deduction, *case* and *casex* statements indicate not only scope, but also the link between a term and the variables that represent its different possible values. Since the construction of *case* statements is delayed, these associations must now be treated explicitly. We will implement them using maps from scope-labeled formulas to variables, and thereby obtain a mnemonic that signals how the relevance of variables may depend on the cases introduced by that formula at that scope. In particular, $fst?(A \vee B^\mu_H)$ and $snd?(A \vee B^\mu_H)$ indicate the variables to be introduced in applying the ($\vee \rightarrow$) rule to a principal formula $A \vee B^\mu_H$. Meanwhile, $unx?(\exists_g A^\mu_H)$ indicates the term variable to be introduced in applying the ($\exists \rightarrow$) rule to a principal formula with label $\exists_g A^\mu_H$. (Since LMU maintains substitutions mapping logic variables to values, it will be necessary to apply such a substitution V to a natural deduction term; in so doing, V must be extended to rename these special variables, so that $V(fst?(M)) = (VM)$, etc.)

Terms for labeling formulas in sequents are constructed according to the following grammar:

$$\begin{aligned} T ::= {} & var \mid fst(T) \mid snd(T) \mid \langle T, T \rangle \mid \\ & inl(T) \mid inr(T) \mid inx(t, T) \\ & (TT^\mu) \mid as\ x\ in\ T \mid (Tt^\mu) \mid type\ v\ in\ T \mid \end{aligned}$$

The superscript on applications indicates the scope of the application. The notation *as v in T* corresponds to $\lambda v . T$ but emphasizes that this construction does not bind v;

and likewise *type* ι *in* T for $\lambda\iota.T$. (The λ, *case*, and *casex* statements that indicate scope come later.)

To encode the partial specification of terms in different parts of the proof, the label of a formula will be a set of terms. Each element of this set will specify a sequence of natural deduction steps which could be appropriately included in the analyses of certain of the cases described by the proof. In describing the labels of formulas, we allow a variable v to abbreviate the singleton $\{v\}$, and we allow $f(F_1,\dots,F_k)$ to abbreviate $\{f(M_1,\dots,M_k)\,|\,M_1 \in F_1,\dots,M_k \in F_k\}$ for constructors and destructors f.

Rules which ordinarily bind variables in proof-terms must be adjusted to accommodate explicit scope. This is done by registering the proof-terms of the inferences on the sequent for later processing. There are two such repositories. First, there is a set T of pairs of annotation terms and proof terms. In each scope, T describes alternative terms that might be constructed, depending on the different cases that must be considered in that scope. T specifications are augmented at $(\rightarrow\supset)$ and $(\rightarrow\forall)$ applications to include the proof terms under which the right-side formula is derived. For example, suppose the sequent rule for $(\rightarrow\supset)$ applies with its side formula B associated with terms F and labeled with scope $\mu\alpha$. Then the rule extends T as derived in the sub-derivation to $T \cup \{(\mu\alpha,M)\,|\,M \in F\}$ – abbreviated $T \cup (\mu\alpha,F)$. The proof-terms for B are otherwise discarded; the principal formula is associated with an anonymous term – *as* v_α *in* $K_\alpha : A \supset B^\mu$.

Second, there is a set C of triples of annotation terms, labeled formulas, and proof terms. C contains a tuple (μ,A,M) if a case analysis of the formula A, depending on the value of M, may be required in scope μ. C specifications are augmented at $(\vee\rightarrow)$ and $(\exists\rightarrow)$ applications, to record the proof terms under which the principal formula is derived. The side formulas are associated with appropriate new variables.

C and T thus indicate how terms are to be reconstructed to replace each K_α variable: cases are introduced corresponding to appropriate elements of C and proof terms are derived in each case corresponding to an appropriate element of T. Now, like substitutions and indexing contexts, the specifications of C and T grow incrementally during a proof, so that input and output values are required on sequents. The overall form of sequents is therefore

$$T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow \Delta$$

(the change in substitution from U to V and the change in indexing from Σ to Θ records the incremental evolution of state as in LMU). The formulas in Γ and Δ are associated with sets of proof terms (in addition to the labeling already needed from LMU). It is convenient also to notate Δ as $L : \Delta$ to indicate that each formula A in Δ is associated with a proof term $L(A)$.

Unlike labels of scope, which allow dependencies that are not used, term labels for right formulas must represent dependencies exactly. Thus, the axiom rule takes the form

$$T; C; U; \Sigma / T; C; MGU(MGU(U,A,B), \mu x, v); \Sigma \triangleright \Gamma, F : B^\mu_H \longrightarrow F : A^\nu_I, \emptyset : \Delta$$

where $\oslash : \Delta$ indicates that Δ is a multiset of formulas each associated with an empty function from cases to terms. Because dependencies are exact, term labels must be merged by sequent rules. For example:

$$\frac{\ldots \triangleright \Gamma \longrightarrow F : A^{\mu}_{H}, G : A \wedge B^{\mu}_{H}, L : \Delta \qquad \ldots \triangleright \Gamma \longrightarrow F' : B^{\mu}_{H}, G' : A \wedge B^{\mu}_{H}, L' : \Delta}{\ldots \triangleright \Gamma \longrightarrow \langle F, F' \rangle \cup G \cup G' : A \wedge B^{\mu}_{H}, L \cup L' : \Delta} \longrightarrow \wedge$$

The notation $L \cup L' : \Delta$ indicates the multiset in which each formula occurrence A of Δ is associated with $L(A) \cup L'(A)$.

Complete rules elaborating LMU sequents with proof-terms are given in Fig. 7. Apart from the nuances about partiality and scope described above, the presentation is essentially identical to that found in [15] and should offer no surprises.

Observe that the various transformations that we have considered so far in this paper can be extended naturally to apply in this calculus. (Each of these regularities can be proved straightforwardly by examination of cases and induction on proofs, if necessary.) For example, if $\mathscr{D}$ is a derivation with proof terms, then we can obtain a weakened derivation $\Lambda + \mathscr{D}$ by adding any multiset Λ of decorated formulas on the left throughout $\mathscr{D}$ and another weakened derivation $\mathscr{D} + \oslash : \Lambda$ obtained by adding on the right throughout $\mathscr{D}$ any multiset Λ in which each formula is assigned the empty set of terms. Moreover, from a derivation ending

$$T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow \Delta$$

we can construct an analogous derivation ending

$$T''; C''; U''; \Sigma'' / T''; C''; U''; \Sigma'' \triangleright \Gamma \longrightarrow \Delta$$

as long as T'', C'', Σ'' contain all the tuples in T', C' and Θ and as long as U'' always equates terms that V equates. These facts allow derivations to be copied and reused: thus, full permutations of inference remain possible in the calculus with proof terms, and, in fact, permutations applied in a proof do not alter its end sequent.

Moreover, appropriate transformations of contraction are available. A proof whose end sequent contains two identical formulas with identical proof-term labels on the left can be simplified to a proof whose end sequent contains a single occurrence of this assumption.

Finally, consider cases where $\mathscr{D}'$ omits part of $\mathscr{D}$, but has identical formulas on left and right in the end sequent to $\mathscr{D}$, with proof-terms labeled identically on the left, and contains only inferences from $\mathscr{D}$. This leaves open that $\mathscr{D}$ has the form:

$$T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow L : \Delta$$

whereas $\mathscr{D}'$ has the form:

$$T; C; U; \Sigma / T''; C''; V'; \Theta' \triangleright \Gamma \longrightarrow L' : \Delta$$

In such a case, $T'' \subseteq T'$, $C'' \subseteq C'$, and $L'(A) \subseteq L(A)$ for all $A \in \Delta$.

$$T; C; U; \Sigma / T; C; MGU(MGU(U, A, B), \mu x, \nu); \Sigma \triangleright \Gamma, F : B^{\mu}_{H} \longrightarrow F : A^{\nu}_{I}, \oslash : \Delta$$

$$\frac{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma,\ F : A \wedge B^{\mu}_{H},\ fst(F) : A^{\mu}_{H};\ snd(F) : B^{\mu}_{H} \longrightarrow \Delta}{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma,\ F : A \wedge B^{\mu}_{H} \longrightarrow \Delta} \wedge\rightarrow$$

$$\frac{T; C; U; \Sigma / T'; C'; W; \Phi \triangleright \Gamma \longrightarrow F : A^{\mu}_{H},\ G : A \wedge B^{\mu}_{H},\ L : \Delta \qquad T'; C'; W; \Phi / T''; C''; V; \Theta \triangleright \Gamma \longrightarrow F' : B^{\mu}_{H},\ G' : A \wedge B^{\mu}_{H},\ L' : \Delta}{T; C; U; \Sigma / T''; C''; V; \Theta \triangleright \Gamma \longrightarrow \langle F, F' \rangle \cup G \cup G' : A \wedge B^{\mu}_{H},\ L \cup L' : \Delta} \rightarrow\wedge$$

$$\frac{T; C; U; \Sigma / T'; C'; W; \Phi \triangleright \Gamma,\ F : A \vee B^{\mu}_{H},\ fst?(A \vee B^{\mu}_{H}) : A^{\mu}_{H} \longrightarrow L : \Delta \qquad T'; C'; W; \Phi / T''; C''; V; \Theta \triangleright \Gamma,\ F : A \vee B^{\mu}_{H},\ snd?(A \vee B^{\mu}_{H}) : B^{\mu}_{H} \longrightarrow L' : \Delta}{T; C; U; \Sigma / T''; C'' \cup (\mu, A \vee B^{\mu}_{H}, F); V; \Theta \triangleright \Gamma,\ F : A \vee B^{\mu}_{H} \longrightarrow L \cup L' : \Delta} \vee\rightarrow$$

$$\frac{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow F : A^{\mu}_{H},\ F' : B^{\mu}_{H},\ G : A \vee B^{\mu}_{H}, \Delta}{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow inl(F) \cup inr(F') \cup G : A \vee B^{\mu}_{H}, \Delta} \rightarrow\vee$$

$$\frac{T; C; U; \Sigma / T'; C'; W; \Phi \triangleright \Gamma,\ F : A \supset B^{\mu}_{H} \longrightarrow G : A^{\mu x}_{H,x},\ L : \Delta \qquad T'; C'; W; \Phi / T''; C''; V; \Theta \triangleright \Gamma,\ F : A \supset B^{\mu}_{H},\ (FG)^{\mu x} : B^{\mu x}_{H,x} \longrightarrow L' : \Delta}{T; C; U; \Sigma / T''; C''; V; \Theta \triangleright \Gamma,\ F : A \supset B^{\mu}_{H} \longrightarrow L \cup L' : \Delta} \supset\rightarrow$$

$$\frac{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma,\ v_{gH} : A^{\mu(gH)}_{H} \longrightarrow F : B^{\mu(gH)}_{H},\ as\ v_{gH}\ in\ K_{gH} : A \supset_{g} B^{\mu}_{H}, \Delta}{T; C; U; \Sigma / T' \cup (\mu(gH), F); C'; V; \Theta \triangleright \Gamma \longrightarrow as\ v_{gH}\ in\ K_{gH} : A \supset_{g} B^{\mu}_{H}, \Delta} \rightarrow\supset$$

$$\frac{T; C; U; \Sigma, y : \mu z / T'; C'; V; \Theta \triangleright \Gamma,\ F : \forall x A^{\mu}_{H},\ (F y^{\mu z}) : A[y/x]^{\mu z}_{H,y,z} \longrightarrow \Delta}{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma,\ F : \forall x A^{\mu}_{H} \longrightarrow \Delta} \forall\rightarrow$$

$$\frac{T; C; U; \Sigma, gH : \mu(hH) / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow F : A[gH/x]^{\mu(hH)}_{H},\ type\ gH\ in\ K_{hH} : \forall_{gh} x A^{\mu}_{H}, \Delta}{T; C; U; \Sigma / T' \cup (\mu(hH), F); C'; V; \Theta \triangleright \Gamma \longrightarrow type\ gH\ in\ K_{hH} : \forall_{gh} x A^{\mu}_{H}, \Delta} \rightarrow\forall$$

$$\frac{T; C; U; \Sigma, gH : \mu / T'; C'; V; \Theta \triangleright \Gamma,\ F : \exists_{g} x A^{\mu}_{H},\ unx?(\exists_{g} x A^{\mu}_{H}) : A[gH/x]^{\mu}_{H} \longrightarrow \Delta}{T; C; U; \Sigma / T'; C' \cup (\mu, \exists_{g} x A^{\mu}_{H}, F); V; \Theta \triangleright \Gamma,\ F : \exists_{g} x A^{\mu}_{H} \longrightarrow \Delta} \exists\rightarrow$$

$$\frac{T; C; U; \Sigma, y : \mu / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow F : A[y/x]^{\mu}_{H,y},\ G : \exists x A^{\mu}_{H}, \Delta}{T; C; U; \Sigma / T'; C'; V; \Theta \triangleright \Gamma \longrightarrow inx(y, F) \cup G : \exists x A^{\mu}_{H}, \Delta} \rightarrow\exists$$

Fig. 7. Assigning proof terms in LMU deductions.

Together with the the theorems of Sections 3.2, 3.4 and 5.1, these regularities ensure that for any labeled proof

$$T; C; U; \Sigma/T'; C'; V; \Theta \rhd \longrightarrow F : A$$

where A does not contain Herbrand terms or variables and is labeled with the empty path, there is proof

$$T; C; U; \Sigma/T''; C''; V; \Theta \rhd \longrightarrow F' : A$$

which contains a correctly ordered sequence of inferences corresponding to an LJ proof and where $T'' \subseteq T'$, $C'' \subseteq C'$ and $F' \subseteq F$.

5.2.3. *Reconstructing terms*

Given specifications T and C and a set δ of variables guaranteed to be bound in the current scope and case, a term $M : A$ is *reconstructed* by a recursive traversal in which subterms of the form K_α are elaborated. The elaboration, *giving cases*, introduces the *case* and *casex* statements needed at scope α, adds the appropriate variable bindings, and recursively reconstructs alternative actions to take at each case. In the recursive invocation, we consider only the subsets of T and C involving tuples with annotation terms that have a proper prefix ending in α; we abbreviate those subsets T_α and C_α. These processes take for granted that the substitution V has applied to T, C and M, and also that eigenvariables have been substituted for Herbrand terms (as performed in showing the correctness of LMU).

The procedures of reconstruction and giving cases are specified nondeterministically, because multiple reconstructions may be possible. These multiple reconstructions arise because LMU allows some inferences to go unused and others to be redundant. Whether such redundant proofs actually are discovered automatically depends on the search strategy; in many cases they will not be.

Definition 5 (*Reconstruction/giving cases*). One term is a reconstruction of another at δ (given T and C) as described by the following cases:

- For any variable x, x is a reconstruction of x at δ if and only if $x \in \delta$.
- For any term M, if N is a reconstruction of M at δ, then $fst(N)$, $snd(N)$, $inl(N)$, $inr(N)$, $inx(t, N)$ and Nt are reconstructions at δ of $fst(M)$, $snd(M)$, $inl(M)$, $inr(M)$, $inx(t, M)$ and Mt^μ, respectively.
- For any terms M and M', if N and N' are reconstructions at δ of M and M', respectively, then $\langle N, N' \rangle$ and NN' are reconstructions at δ of $\langle M, M' \rangle$ and MM'^μ, respectively.
- For any term K_α, if N gives cases at $\delta \cup \{v_\alpha\}$ for K_α (given T and C), then $\lambda v_\alpha N$ is a reconstruction at δ of *as* v_α *in* K_α and $\lambda \iota N$ is a reconstruction at δ of *type* ι *in* K_α.

N gives cases for K_α at δ given T and C according to the following conditions:

- There is some formula A of the form $\exists_g x B^\mu_H$ with a tuple $(\alpha, A, M) \in C$ with $unx?(A) \notin \delta$ and for which there is a reconstruction M' of M at δ (given T and C).

N may be any term of the form:

$$casex(M' \textit{ of } inx(a, unx?(M)) \Rightarrow R)$$

where a is the eigenvariable introduced by the quantifier, and R gives cases for K_α at $\delta \cup \{unx?(M)\}$ (given T and C).

- There is some formula A of the form $B \vee C_H^\mu$ with a tuple $(\alpha, A, M) \in C$ with neither $fst?(A) \in \delta$ nor $snd?(A) \in \delta$ and for which there is a reconstruction M' of M at δ. N may be any term of the form:

  $$case(M' \textit{ of } inl(fst?(A)) \Rightarrow R \mid inr(snd?(A)) \Rightarrow R')$$

  where R gives cases for K_α at $\delta \cup \{fst?(A)\}$ (given T and C), and R' gives cases for K_α at $\delta \cup \{snd?(A)\}$ (given T and C).
- Otherwise, N may be any term obtained by reconstructing at δ any term M for which $(\alpha, M) \in T$, given T_α and C_α.

Observe that if $T' \subseteq T$, $C' \subseteq T$ and there is a reconstruction of M at δ given T' and C', then there is a reconstruction of M at δ given T and C. This must be so if M does not refer to K_α terms, since T and C will not figure in the reconstruction of M. Inductive reasoning then shows that for each case treated during reconstruction of M for T' and C', a case binding the same variables will also be treated during reconstruction of M for T and C; thus, although additional cases may show up in reconstructing M for T and C, after each path the same term in T can be reconstructed as the term reconstructed in the analogous case from T'.

It is also easy to see that the result of reconstruction corresponds to a natural deduction proof.

Theorem 6 (Correctness of extraction). *Consider a proof with end sequent of the form*

$$\oslash; \oslash; U; \Sigma/T; C; V; \Theta \rhd \longrightarrow F : A$$

(Such a proof describes a complete derivation.) Suppose that N is a reconstruction of any term $M \in F$ for T and C at $\oslash$. Then N represents a natural deduction proof of A.

Proof. When any proof-term variable x is reconstructed in a term, it will be bound. For this will happen only when $x \in \delta$, but when we reconstruct starting from $\oslash$, at each recursive invocation δ contains only variables whose binding operators will surround the term being constructed.

Since every variable in N is thus correctly bound, to show that N represents a natural deduction proof of A, it suffices to show that each variable is used with a consistent type throughout N. This follows immediately from the construction and deconstruction of proof-terms in lock-step with formulas in the sequent calculus, and the existence of a unifying substitution matching the types of left and right occurrences of variables. □

We can also establish that a λ term can always be reconstructed according to this scheme:

Theorem 7 (Completeness of extraction). *Given a deduction $\mathscr{D}$ with end sequent*

$$\oslash; \oslash; U; \Sigma/T; C; V; \Theta \rhd \longrightarrow F : A$$

Then for some $M \in F$ there is some reconstruction of M for T and C at $\oslash$.

Proof. As observed in Section 5.2.2, we can transform $\mathscr{D}$ to a deduction corresponding to an LJ inference, which records inferences $T' \subseteq T$ and $C' \subseteq C$ and derives terms $F' \subseteq F$.

As observed above, if there is a reconstruction of any term M in F from T' and C', then there is a reconstruction of M from T and C. So it suffices to consider $\mathscr{D}'$. This has the advantage that reconstruction of $\mathscr{D}'$ can proceed in lock-step with the syntactic structure of the proof.

We show by induction on the structure of $\mathscr{D}'$ the following property of proofs. Let the end sequent of the proof be

$$T; C \ldots /T'; C' \ldots \Gamma \longrightarrow \Delta$$

Let δ denote the proof-term variables free in the labels of Γ formulas and suppose some term in each of those labels can be reconstructed according to T and C at δ. Let $F : A^{\mu}$ be the distinguished formula in Δ such that the proof corresponds to an LJ proof of $\Gamma \longrightarrow F : A^{\mu}$ by the cleaning transformations of Lemma 1. Γ is empty except above $(\longrightarrow \supset)$; so if the proof ends in a left rule, μ has the form $\nu\alpha$. Then if the proof ends in a left rule then giving cases for K_{α} at δ after $\oslash$ given $T' \cup (\mu, F)$ and C' succeeds; and otherwise we have $M \in F$ where reconstructing M at δ given T' and C' succeeds.

Note that the requirement placed on deductions ending in right rules entails that placed on deductions ending in left rules. If reconstructing M at δ given T' and C' succeeds, then no matter what further cases are introduced in elaborating cases at α using $T' \cup (\mu, M)$ and C', we always arrive at a leaf at which the reconstruction of M can be used.

We illustrate the key cases of the induction here. At axioms, the label F is a label of some left formula, which by assumption can be reconstructed.

At $(\supset\longrightarrow)$, the left subderivation ends

$$T; C \ldots /T''; C'' \ldots \Gamma \longrightarrow G : A, \ldots$$

By the construction of $\mathscr{D}'$, this derivation cannot end in a right rule (see Lemma 1); $G : A$ is the side formula of the $(\supset\longrightarrow)$ inference. Therefore, by the induction hypothesis, there is a reconstruction of $G : A$ given T'' and C''. Now the right subderivation ends

$$T''; C''; \ldots /T'; C' \ldots \Gamma, (FG)^{\mu} : B^{\mu}_{H} \longrightarrow F' : A', \ldots$$

We can now conclude that the induction hypothesis applies to this derivation. The labels of Γ continue to have reconstructions given T'' and C'', because they extend T and C. Moreover, since we have reconstructions for elements of F and G given

$$\dfrac{\dfrac{\mathcal{D}_1 \qquad\qquad \mathcal{D}_2}{/;\{(\alpha, A\vee B^{\alpha}, v_\alpha)\};\tau \rhd f: A\supset C,\ g: B\supset C,\ v_\alpha : A\vee B^{\alpha} \longrightarrow f(\mathit{fst?}(v_\alpha))^x \cup g(\mathit{snd?}(v_\alpha))^y : C^{\alpha},\ldots}}{/\{(\alpha, f(\mathit{fst?}(v_\alpha))^x),(\alpha, g(\mathit{snd?}(v_\alpha))^y)\};\{(\alpha, A\vee B^{\alpha}, v_\alpha)\};\tau\rhd f: A\supset C,\ g: B\supset C \longrightarrow \mathit{as}\ v_\alpha\ \mathit{in}\ K_\alpha : A\vee B\supset C}$$

$$\mathcal{D}_1 = \boxed{\dfrac{/\sigma \rhd \ldots \mathit{fst?}(v_\alpha) : A^{\alpha} \longrightarrow \mathit{fst?}(v_\alpha) : A^{x},\ldots \qquad \sigma/\sigma \rhd \ldots, f(\mathit{fst?}(v_\alpha))^x : C^{x} \longrightarrow f(\mathit{fst?}(v_\alpha))^x : C^{\alpha},\ldots}{/\sigma \rhd \ldots, f: A\supset C,\ \mathit{fst?}(v_\alpha) : A^{\alpha} \longrightarrow f(\mathit{fst?}(v_\alpha))^x : C^{\alpha},\ldots}}$$

$$\mathcal{D}_2 = \boxed{\dfrac{\sigma/\tau \rhd \mathit{snd?}(v_\alpha) : B^{\alpha} \longrightarrow \mathit{snd?}(v_\alpha) : B^{y} \qquad \tau/\tau \rhd \ldots, g(\mathit{fst?}(v_\alpha))^y : C^{y} \longrightarrow g(\mathit{snd?}(v_\alpha))^y : C^{\alpha},\ldots}{\sigma/\tau \rhd \ldots, g: B\supset C,\ \mathit{snd?}(v_\alpha) : B^{\alpha} \longrightarrow g(\mathit{snd?}(v_\alpha))^y : C^{\alpha}}}$$

a. Extracting a term from the usual proof by this system; $\sigma = \alpha/x$; $\tau = \alpha/x, \alpha/y$.

$$\dfrac{\dfrac{\mathcal{D}_1 \qquad\qquad \mathcal{D}_2}{/\{(\alpha, f(\mathit{fst?}(v_\alpha))^x)\}; C; \tau \rhd f: A\supset C \longrightarrow \mathit{snd?}(v_\alpha) : B^{\alpha},\ \mathit{as}\ v_\alpha\ \mathit{in}\ K_\alpha : A\vee B\supset C} \qquad \mathcal{D}_3}{/\{(\alpha, f(\mathit{fst?}(v_\alpha))^x),(\alpha, g(\mathit{snd?}(v_\alpha))^y)\};\{(\alpha, A\vee B^{\alpha}, v_\alpha)\};\tau\rhd f: A\supset C,\ g: B\supset C \longrightarrow \mathit{as}\ v_\alpha\ \mathit{in}\ K_\alpha : A\vee B\supset C}$$

$$\mathcal{D}_1 = \boxed{\dfrac{\dfrac{/\sigma \rhd \ldots, \mathit{fst?}(v_\alpha) : A^{\alpha} \longrightarrow \mathit{fst?}(v_\alpha) : A^{\alpha},\ldots \qquad \sigma/\tau \rhd \ldots, \mathit{snd?}(v_\alpha) : B^{\alpha} \longrightarrow \mathit{snd?}(v_\alpha) : B^{\alpha},\ldots}{/;\{\alpha, A\vee B^{\alpha}, v_\alpha\};\tau \rhd \ldots, v_\alpha : A\vee B^{\alpha} \longrightarrow \mathit{fst?}(v_\alpha) : A^{x},\ \mathit{snd?}(v_\alpha) : B^{y},\ \emptyset : C^{\alpha},\ldots}}{/;\{\alpha, A\vee B^{\alpha}, v_\alpha\};\tau \rhd \ldots \longrightarrow \ldots, \mathit{fst?}(v_\alpha) : A^{\alpha},\ \mathit{snd?}(v_\alpha) : B^{\alpha},\ \mathit{as}\ v_\alpha\ \mathit{in}\ K_\alpha : A\vee B\supset C}}$$

$$\mathcal{D}_2 = \boxed{\dfrac{; C; \tau/; C; \tau \rhd \ldots, v_\alpha : A\vee B^{\alpha}, f(\mathit{fst?}(v_\alpha))^x : C^{x} \longrightarrow f(\mathit{fst?}(v_\alpha))^x : C^{\alpha},\ldots}{; C; \tau/\{(\alpha, f(\mathit{fst?}(v_\alpha))^x)\}; C; \tau \rhd f(\mathit{fst?}(v_\alpha))^x : C^{x} \longrightarrow \mathit{as}\ v_\alpha\ \mathit{in}\ K_\alpha : A\vee B\supset C,\ldots}}$$

$$\mathcal{D}_3 = \boxed{\dfrac{T_1; C; \tau/T_1; C; \tau \rhd v_\alpha : A\vee B^{\alpha}, g(\mathit{fst?}(v_\alpha))^y : C^{y} \longrightarrow g(\mathit{snd?}(v_\alpha))^y : C^{\alpha},\ldots}{T_1; C; \tau/\{(\alpha, f(\mathit{fst?}(v_\alpha))^x),(\alpha, g(\mathit{snd?}(v_\alpha))^y)\}; C; \tau \rhd \ldots g(\mathit{snd?}(v_\alpha))^y : C^{y} \longrightarrow \mathit{as}\ v_\alpha\ \mathit{in}\ K_\alpha : A\vee B\supset C,\ldots}}$$

b. The permuted proof. $\sigma = \alpha/x$; $\tau = \alpha/x, \alpha/y$.

Fig. 8. Extracting λ terms from the proofs of Figure 6.

T'' and C'', we must also have a reconstruction for an element of $(FG)^{\mu}$ there. The induction hypothesis therefore supplies a reconstruction or case analysis of F'. Since this is a left rule, this suffices.

At $(\vee\longrightarrow)$, we have two subderivations:

$$T; C \ldots /T''; C'' \ldots \Gamma, \mathit{fst?}(A\vee B^{\mu}_{H}) : A^{\mu}_{H} \longrightarrow F : A'^{\nu\alpha}_{H'},\ldots$$

$$T''; C'' \ldots /T'; C' \ldots \Gamma, \mathit{snd?}(A\vee B^{\mu}_{H}) : B^{\mu}_{H} \longrightarrow F' : A'^{\nu\alpha}_{H'},\ldots$$

The induction hypothesis applies to both. Further, the condition on Γ ensures that some element of the label G of the principal formula of the rule can be reconstructed given T and C – and thus given T' and C'. We must show that we can give cases for K_α given $T' \cup (\nu\alpha, F\cup F')$ and $C' \cup (\mu, A\vee B^{\mu}_{H}, G)$. Giving cases begins at δ (corresponding to the free variables end sequent) by treating the case for G. This leaves the subproblems of giving cases for $\delta\cup\{\mathit{fst?}(A\vee B^{\mu}_{H})\}$ and $\delta\cup\{\mathit{snd?}(A\vee B^{\mu}_{H})\}$ – that this is feasible is guaranteed by the induction hypothesis.

At $(\longrightarrow\supset)$, the subderivation ends:

$$T; C \ldots /T''; C'' \ldots \Gamma, v_\alpha : A^{\mu\alpha}_{H} \longrightarrow M : B^{\mu\alpha}_{H},\ldots$$

The induction hypothesis applies immediately; this shows (at least) that there is a case analysis of K_α given $T''\cup(\mu\alpha, M)$ and C'' at $\delta\cup\{v_\alpha\}$. But if this is possible, then we can reconstruct *as* v_α *in* K_α at δ given these records. This is just what we need to establish for the overall derivation. □

Fig. 8 shows the application of this system to the proofs of Fig. 6, and illustrates this result. In both cases, we are left with the problem of reconstructing the term *as* v_α *in* K_α given $T = \{(\alpha, f(fst?(v_\alpha))^\alpha), (\alpha, g(snd?(v_\alpha))^\alpha)\}$ and $C = \{(\alpha, A \vee B^\alpha, v_\alpha)\}$ (with δ providing for assumed variables f and g). We give cases for K_α by finding the case v_α, which reconstructs to itself, and reconstructing the terms $f(fst?(v_\alpha))^x$ and $g(snd?(v_\alpha))^y$ for the two outcomes of v_α. Letting $v = fst?(v_\alpha)$ and $v' = snd?(v_\alpha)$, we arrive, as expected, at $\lambda v_\alpha\, case(v_\alpha\ of\ inl(v) \Rightarrow f(v) \mid inr(v') \Rightarrow g(v'))$.

6. Conclusion

This paper has considered an alternative proof system for intuitionistic logic, and justified it by a syntactic argument. Although inspired by translation proof methods, this is a distinct, more direct result. In fact, together with the soundness and completeness theorems for classical logic, this result effectively amounts to an alternative demonstration of the soundness and completeness of (fallible) Kripke semantics for characterizing LJ proofs. Further, its proof-theoretic formulation makes possible new applications of translation methods in logic programming and program synthesis.

More generally, this work shows one way to construct efficient inference procedures by developing syntactic abstractions for scope and information-flow in proofs. This new strategy contrasts with the strategy of [11] of labeling formulas based on the content of the sequents from which they are to be proved. We use symbols to name the inferences in the proof that represent a change of scope and strings of symbols to encode the scoped position of other inferences in the proof. By imposing constraints on these symbols that mirror to the constraints imposed in a structural discipline of scope, we arrive at system of explicit scope that allows rules to be used in any order. Because terms represent positions in the proof, the terms themselves describe where the inferences belong according to the original structure of proofs.

We see then that such procedures can have extremely simple statements and extremely natural justifications. Moreover, as the development of a family of systems including LMA and LMT shows, such abstractions are not tied directly to any one semantics and can build on structure already implicit in a proof system. This raises the prospect of applying this idea to other systems, particularly linear logic [22], even in the absence of compelling classical semantics. The task remains daunting since there must be at least three kinds of scope transition in linear logic, corresponding to the splitting of context at $(\rightarrow \otimes)$, the copying of context at $(\rightarrow \vee)$, and the modalization of context at $(\rightarrow !)$. These scope transitions interact in complicated ways, as underscored by permutability studies [2, 21, 44]. We leave this problem to future research.

Acknowledgements

Thanks to Mark Steedman and an anonymous reviewer for extensive comments and Jean Gallier and Dale Miller for helpful discussion. This work was supported by an

NSF graduate fellowship, and an IRCS graduate fellowship, as well as NSF grant IRI95-04372, ARPA grant N6601-94-C6043, and ARO grant DAAH05-94-G0426.

References

[1] S.J. Aitken, H. Reichgelt, N. Shadbolt, Resolution theorem proving in reified modal logics, J. Automat. Reason. 12 (1) (1994) 103–129.

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

[3] P.B. Andrews, Theorem proving via general matings, J. Assoc. Comput. Mach. 28 (2) (1981) 193–214.

[4] Y. Auffray, P. Enjalbert, Modal theorem proving: an equational viewpoint, J. Logic Comput. 2 (3) (1992) 247–295.

[5] W. Bibel, Automated Theorem Proving, Vieweg, Braunschweig, 1982.

[6] O. Bittel, Tableau-based theorem proving and synthesis of λ-terms in the intuitionistic logic, in: Logics in AI, Lecture Notes in Computer Science, vol. 633, Springer, Berlin, 1992, pp. 262–278.

[7] O. Bittel, A tableau-based theorem proving method for intuitionistic logic, in: Automated Deduction in Nonstandard Logics: Papers from the 1993 Fall Symp., AAAI Press Technical Report FS-93-01, 1993, pp. 9–16.

[8] R.S. Boyer, J.S. Moore, The sharing of structure in theorem-proving programs, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence 7, Edinburgh University Press, 1972, pp. 101–116.

[9] R.L. Constable et al., Implementing Mathematics with the Nuprl Proof Development System, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[10] M. D'Agostino, Are tableaux and improvement on truth-tables, J. Logic Language Inform. 1 (1992) 235–252.

[11] M. D'Agostino, D.M. Gabbay, A generalization of analytic deduction via labelled deductive systems. part I: Basic substructural logics, J. Automat. Reason. 13 (1994) 243–281.

[12] M. D'Agostino, M. Mondadori, The taming of the cut. classical refutations with analytic cut, J. Logic Comput. 4 (1994) 285–319.

[13] F. Debart, P. Enjalbert, M. Lescot, Multimodal logic programming using equational and order-sorted logic, Theoret. Comput. Sci. 105 (1992) 141–166.

[14] R. Dyckhoff, Contraction-free sequent calculi for intuitionistic logic, J. Symbolic Logic 57 (1992) 795–807.

[15] A. Felty, A logic program for transforming sequent proofs to natural deduction proofs, in: P. Schroeder-Heister (Ed.), Proc. 1989 Internat. Workshop on Extensions of Logic Programming, Lecture Notes in Artificial Intelligence, 1991, pp. 157–178.

[16] M.C. Fitting, Intuitionistic Logic, Model Theory and Forcing, North-Holland, Amsterdam, 1969.

[17] M. Fitting, Proof Methods for Modal and Intuitionistic Logics, Synthese Library, vol. 169, D. Reidel, Dordrecht, 1983.

[18] M. Fitting, A modal Herbrand theorem, Fund. Inform. 28 (1996) 101–122.

[19] J.H. Gallier, Logic for Computer Science: Foundations of Automated Theorem Proving, Harper and Row, New York, 1986.

[20] J. Gallier, Constructive logics. I. A tutorial on proof systems and typed λ-calculi, Theoret. Comput. Sci. 110 (2) (1993) 249–339.

[21] D. Galmiche, G. Perrier, On proof normalization in linear logic, Theoret. Comput. Sci. 135 (1994) 67–110.

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

[23] H. Herbelin, A λ-caculus structure isomorphic to Gentzen-style sequent calculus structure, in: CSL 94, Springer, Berlin, 1994, pp. 61–75.

[24] W.A. Howard, The formulae-as-types notion of construction, in: To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism, Academic Press, New York, 1980, pp. 479–490.

[25] P. Jackson, H. Reichgelt, A general proof method for first-order modal logic, in: Proc. IJCAI, 1987, pp. 942–944.

[26] S.C. Kleene, Permutation of inferences in Gentzen's calculi LK and LJ, in: Two Papers on the Predicate Calculus, American Mathematical Society, Providence, RI, 1951, pp. 1–26.

[27] S.C. Kleene, Introduction to Metamathematics, North-Holland, Amsterdam, 1952.
[28] S.A. Kripke, Semantical analysis of intuitionistic logic, in: J. Crossley, M.A.E. Dummett (Eds.), Formal Systems and Recursive Functions, North-Holland, Amsterdam, 1965, pp. 92–130.
[29] P.D. Lincoln, N. Shankar, Proof search in first-order linear logic and other cut-free sequent calculi, in: LICS, 1994, pp. 282–291.
[30] P. Martin-Löf, Constructive mathematics and computer programming, in: L.J. Cohen (Ed.), Logic, Methodology and Philosophy of Science, vol. VI, North-Holland, Amsterdam, 1982, pp. 153–175.
[31] D. Miller, A logical analysis of modules in logic programming, J. Logic Programming 6 (1–2) (1989) 79–108.
[32] D. Miller, G. Nadathur, F, Pfenning, A. Scedrov, Uniform proofs as a foundation for logic programming, Ann. Pure Appl. Logic 51 (1991) 125–157.
[33] D. Miller, A multiple-conclusion meta-logic, in: S. Abramsky (Ed.), Proc. Internat. Symp. on Logics Comput. Sci., 1994, pp. 272–281.
[34] G. Nadathur, A proof procedure for the logic of hereditary Harrop formulas, J. Automat. Reason. 11 (1993) 115–145.
[35] H.J. Ohlbach, Semantics-based translation methods for modal logics, J. Logic Comput. 1 (1991) 691–746.
[36] H.J. Ohlbach, Optimized translation of multi modal logic into predicate logic, in: A. Voronkov (Ed.), Logic Programming and Automated Reasoning, Lecture Notes in Computer Science, vol. 698, Springer, Berlin, 1993, pp. 253–264.
[37] J. Otten, C. Kreitz, T-string-unification: unifying prefixes in non-classical proof methods, in: TABLEAUX 96, Lecture Notes in Artificial Intelligence, vol. 1071, Springer, Berlin, 1996, pp. 244–260.
[38] D. Prawitz, Ideas and results in proof theory, in: J.E. Fenstad (Ed.), Proc. 2nd Scand. Logic Symp., North-Holland, Amsterdam, 1971, pp. 235–307.
[39] H. Schellinx, Some syntactical observations on linear logic, J. Logic Comput. 1 (1991) 537–559.
[40] N. Shankar, Proof search in the intuitionistic sequent calculus, in: Automated Deduction: CADE 11, Springer, Berlin, 1992, pp. 522–536.
[41] R.M. Smullyan, First-order Logic, Ergebnisse der Mathematik und ihere Grenzgebeite, vol. 43, Springer, Berlin, 1968.
[42] R.M. Smullyan, A generalization of intuitionistic and modal logics, in: H. Leblanc (Ed.), Truth, Syntax and Modality, North-Holland, Amsterdam, 1973, pp. 274–293.
[43] M. Stone, Efficient constraints on possible worlds for reasoning about necessity, Technical Report IRCS 97-7 and CIS 97-10, University of Pennsylvania.
[44] T. Tammet, Proof strategies in linear logic, J. Automat. Reason. 12 (1994) 273–304.
[45] N. Tennant, Autologic, Edinburgh Information Technology Series, No. 9, Edinburgh University Press, Edinburgh, 1992.
[46] A.S. Troelstra, D. van Dalen, Constructivism in Mathematics, vol. 1, North-Holland, Amsterdam, 1988.
[47] A.S. Troelstra, D. van Dalen, Constructivism in Mathematics, vol. 2, North-Holland, Amsterdam, 1988.
[48] W. Veldman, An intuitionistic completeness theorem for intuitionistic predicate logic J. Symbolic Logic 41 (1976) 159–166.
[49] A. Voronkov, Proof-search in intuitionistic logic based on constraint satisfaction, in: TABLEAUX 96, Lecture Notes in Artificial Intelligence, vol. 1071, Springer, Berlin, 1996, pp. 312–329.
[50] L.A. Wallen, Automated Proof Search in Non-Classical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics, MIT Press, Cambridge, 1990.
[51] S. Schmitt, Ein erweiterter intuitionistischer Sequenzenkalkül und dessen Anwendung im intutitiotistischen Konnektionsbeweisen, Master's Thesis, TH Darmstadt, 1994.
[52] S. Schmitt and C. Kreitz, On transforming intuitionistic matrix proofs into standard-sequent proofs, in: Proc. Fourth Workshop on Theorem Proving with Analytic Tableaux and Related Methods, volume 918 of LNAI, pp. 106–121, 1995.
[53] S. Schmitt and C. Kreitz, Converting non-classical matrix proofs into sequent-style systems, in: M. McRobbie and J. Slaney, Eds., Proc. Thirteenth International Conference on Automated Deduction, volume 1104 of LNAI, pp. 418–432, 1996.