Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents

Rajeev Goré, Linda Postniece and Alwen Tiu

ABSTRACT. We propose a new sequent calculus for bi-intuitionistic logic which sits somewhere between display calculi and traditional sequent calculi by using nested sequents. Our calculus enjoys a simple (purely syntactic) cut-elimination proof as do display calculi. But it has an easily derivable variant calculus which is amenable to automated proof search as are (some) traditional sequent calculi. We first present the initial calculus and its cut-elimination proof. We then present the derived calculus, and then present a proof-search strategy which allows it to be used for automated proof search. We prove that this search strategy is terminating and complete by showing how it can be used to mimic derivations obtained from an existing calculus GBiInt for bi-intuitionistic logic. As far as we know, our new calculus is the first sequent calculus for bi-intuitionistic logic which uses no semantic additions like labels, which has a purely syntactic cut-elimination proof, and which can be used naturally for backwards proof-search.

Keywords: Bi-intuitionistic logic, display calculi, proof search.

1 Introduction

Bi-intuitionistic logic (BiInt) is obtained by extending intuitionistic logic with a binary connective variously called "subtraction" or "exclusion" or even "co-implication", which we write as $-\!\!<$. Intuitively, the formula $A -\!\!< B$ reads "A excludes B". We assume the reader is familiar with the Kripke semantics for intuitionistic logic using a binary reflexive and transitive relation $\leq$. Then the forcing relation for the exclusion operator is defined as

$$w \Vdash A -\!\!< B \quad \text{iff} \quad \exists v \leq w . v \Vdash A \text{ and } v \not\Vdash B.$$

In a sequent setting, the introduction rules for exclusion are dual to the introduction rules for implication, e.g., the left-introduction rule $-\!\!<_L$ is dual to the right-introduction rule $\to_R$ for implication as shown below:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \to_R \qquad\qquad \frac{A \vdash B, \Delta}{A -\!\!< B \vdash \Delta} -\!\!<_L$$

If we remove the implication connective from bi-intuitionistic logic, we obtain a logic called *dual intuitionistic logic*. Both intuitionistic logic and dual intuitionistic logic have simple sequent calculus formulations which enjoy cut

Advances in Modal Logic, Volume 7.

elimination. However, the combined logic has surprisingly no known simple sequent calculus formulation which enjoys true cut-elimination.

Rauszer [21] formalised a sequent calculus for BiInt, but it was later found to fail cut elimination (see [5] for a counterexample). Crolard's work [7] is based upon Rauszer's calculus and so also fails cut-elimination. In response, Pinto and Uustalu announced a labelled sequent calculus with cut-elimination for BiInt [28], but have yet to provide a full paper outlining details. Their calculus does not provide a purely syntactic account of BiInt since it uses labelled formulae of the form $x : A$ to capture the semantic notion that "Kripke world x makes formula A true". Independently, a purely syntactic cut-free calculus for (proof-search in) BiInt was given by Postniece (previously Buisman) and Goré [5, 13] by combining a "refutation" calculus and a "provability" calculus. However, their cut-elimination result is obtained indirectly via a semantic argument which shows that the cut-free fragment of their calculus is complete with respect to the Kripke semantics of BiInt. Thus the only truly cut-free calculus for BiInt appears to be the display calculus due to Goré [12] (see [30] for a variation).

Although display calculi were not designed for automated proof-search there is a surprising lack of interest in the study of proof search for display logics: the only exceptions are the works of Wansing [29] and Restall [23]. The main difficulty in using display calculi for proof search are the invertible structural display postulate rules which are at the heart of display calculi. Although these rules guarantee the display property, they allow "pointless" shuffling of structures and easily lead to non-termination of proof search if applied naively. Another issue is the presence of explicit contraction and weakening rules in display calculi which are couched in terms of structures rather than formulae. Replacing these rules with ones based on formulae can break one of the conditions for a display calculus, namely, the (C6/C7) condition that "each rule is closed under simultaneous substitution of arbitrary structures for congruent parameters" [18]. Absorbing them completely to obtain a "contraction-free" calculus is thus not an obvious step.

Here, we present two sequent-like calculi LBiInt_1 and LBiInt_2 for bi-intuitionistic logic which sit somewhere in between display calculi and traditional sequent calculi in terms of cut-elimination and proof-search.

$\mathbf{LBiInt}_1$: The calculus LBiInt_1 shares some features of display calculi, in that it has certain structural rules that allow shuffling of structures in a sequent, akin to the display postulates used in display calculi to display a formula nested in a structure. The syntactic judgments in LBiInt_1 can be seen as a tree of (traditional) sequents, and the structural rules can be used to "display" a sequent by bringing it to the root of an equivalent tree. The logical rules of LBiInt_1 are similar to those in Gentzen's traditional sequent calculus, as they apply only to the topmost sequent in the tree of sequents. The virtue of LBiInt_1 is twofold: its contraction and weakening rules can be restricted to formulae while its purely syntactic cut-elimination proof is simple and very similar to the cut-elimination proof for display calculi.

LBiInt$_2$: The calculus LBiInt$_2$ is a refinement of LBiInt$_1$ and is obtained by absorbing all the structural rules of LBiInt$_1$ into the logical rules. The calculus LBiInt$_2$ is easily shown to be sound, since its rules are derivable in LBiInt$_1$. But from a proof-search perspective, we are able to associate a terminating and systematic backward proof-search strategy for applying the rules of LBiInt$_2$. However, we currently do not have a direct syntactic proof of completeness of LBiInt$_2$ with respect to LBiInt$_1$. Instead, completeness of LBiInt$_2$ is shown by an encoding of the calculus GBiInt [5], which is known to be sound and complete for bi-intuitionistic logic. The translation is natural and shows an interesting duality between GBiInt and LBiInt$_2$. It also gives a first simple proof theoretic account of the proof search strategy associated with GBiInt (which is largely semantically motivated).

Our methodology is to use structures (called nested sequents) which are similar to the structures in display calculi but which are more restricted than those used in display calculi. In particular, not all the display structural connectives used in Goré's calculus [12] are allowed and certain display postulates are missing. The idea is to get as close as possible to sequent calculus, because then we may be able to use the standard saturation techniques for proof search common in sequent calculus. Since our calculi are not display calculi, Belnap's general cut elimination theorem [2] cannot be used directly to prove cut elimination for our calculi. One way of showing cut elimination for LBiInt$_1$ would be an indirect proof via a detour through display calculus. That is, one first designs a corresponding display system for LBiInt$_1$ for which Belnap's cut elimination theorem can be used, e.g., by modifying Goré's calculus to work with a more restricted form of structures, and then showing that the cut free proofs of this display system can be mapped to cut-free proofs of LBiInt$_1$. We show here a simple and direct cut elimination proof, without detour through display calculus and Belnap's theorem, by using a certain proof substitution technique, which is very similar to Belnap's original cut elimination proof. We believe that this cut elimination proof can be extended to other logics which contain pairs of adjoint connectives like classical modal (tense) logics such as $KtS4$ and its cousins. If so, then there is a possibility of obtaining general characterisations of cut admissibility like those of Belnap's for these logics.

Outline of the paper Sections 2-4 present the calculus LBiInt$_1$ and its meta theory via theorems on cut elimination, soundness and completeness. While the structural rules in LBiInt$_1$ are somewhat more restrictive than display calculi, and hence reduce slightly the non-determinism arising from the structural rules of display calculi, they still pose some difficulty in proof search. In Section 5, we present a restricted version of LBiInt$_1$, called LBiInt$_2$, in which all the structural rules are omitted and are instead absorbed into introduction rules. We give a terminating proof search strategy for LBiInt$_2$. The idea behind backward proof search for LBiInt$_2$ is that the introduction rules for implication and subtraction can be used to 'suspend'

Identity and cut:

$$\frac{}{X, A \vdash A, Y}\ id \qquad \frac{X_1 \vdash Y_1, A \quad A, X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

Structural rules:

$$\frac{X \vdash Y}{X, A \vdash Y}\ w_L \qquad \frac{X \vdash Y}{X \vdash A, Y}\ w_R \qquad \frac{X, A, A \vdash Y}{X, A \vdash Y}\ c_L \qquad \frac{X \vdash A, A, Y}{X \vdash A, Y}\ c_R$$

$$\frac{(X_1 < Y_1), X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ s_L \qquad \frac{X_1 \vdash Y_1, (X_2 > Y_2)}{X_1, X_2 \vdash Y_1, Y_2}\ s_R$$

$$\frac{X_2 \vdash Y_2, Y_1}{X_1, (X_2 < Y_2) \vdash Y_1}\ < \qquad \frac{X_1, X_2 \vdash Y_2}{X_1 \vdash Y_1, (X_2 > Y_2)}\ >$$

Logical rules:

$$\frac{X, B_i \vdash Y}{X, B_1 \wedge B_2 \vdash Y}\ \wedge_L\ i \in \{1,2\} \qquad \frac{X \vdash A, Y \quad X \vdash B, Y}{X \vdash A \wedge B, Y}\ \wedge_R$$

$$\frac{X, A \vdash Y \quad X, B \vdash Y}{X, A \vee B \vdash Y}\ \vee_L \qquad \frac{X \vdash B_i, Y}{X \vdash B_1 \vee B_2, Y}\ \vee_R\ i \in \{1,2\}$$

$$\frac{X \vdash A, Y \quad X, B \vdash Y}{X, A \to B \vdash Y}\ \to_L \qquad \frac{X, A \vdash B}{X \vdash Y, A \to B}\ \to_R$$

$$\frac{A \vdash B, Y}{X, A \prec B \vdash Y}\ \prec_L \qquad \frac{X \vdash A, Y \quad X, B \vdash Y}{X \vdash A \prec B, Y}\ \prec_R$$

Figure 1. LBiInt_1: a sequent calculus for bi-intuitionistic logic

proof search of a (top-level) sequent and to 'restart' it at a later stage. Such restart rules are already known in the literature, but as far as we are aware, our work is the first time they have been given a purely proof-theoretic setting. In the same section we also show that we can encode the sound and complete calculus GBiInt [5] into LBiInt_2, thereby giving an indirect proof of the completeness of LBiInt_2. Section 6 discusses related and future work. An extended version of the paper with more details will be made available on the web.

2 System LBiInt_1

Formulas of bi-intuitionistic logic are given by the following grammar:

$$A := p \mid A \to A \mid A \prec A \mid A \wedge A \mid A \vee A.$$

Negative and positive structures [1] are expressions generated, respectively, from the following grammars:

$$N := \emptyset \mid A \mid (N, N) \mid N < P \qquad\qquad P := \emptyset \mid A \mid (P, P) \mid N > P.$$

The structural (comma) connective "," is associative and commutative and $\emptyset$ is its unit. We always consider structures modulo these equivalences.

A *sequent* is an expression of the form $X \vdash Y$, where X is a negative structure and Y is a positive structure. To reduce parentheses, we assume that the structural connective "," binds tighter than $>$ and $<$. Thus, we write $X, Y > Z$ to mean $(X, Y) > Z$.

A *context* is a structure with a hole, denoted with $Z[]$. We write $Z[X]$ to denote a structure resulting from filling the hole in the context $Z[]$ with the structure X. Note that such a replacement does not always give a legal structure. For example, if $Z[]$ is $[] < p$ and X is $q > r$, then $Z[X] = (q > r) < p$ is not a structure since we have a positive structure to the left of $<$. A k-hole context is a context with k holes. Given a k-hole context $Z[\cdots]$ we write $Z[X^k]$ to stand for the structure obtained from $Z[\cdots]$ by replacing each hole with an occurrence of the structure X. An *anti-positive context* is a context $Z[]$ such that $Z[X]$ is a negative structure for every positive structure X. An *iso-positive context* is a context $Z[]$ such that $Z[X]$ is a positive structure for every positive structure X. Likewise, an *anti-negative context* is a context $Z[]$ such that $Z[X]$ is a positive structure for every negative structure X, and $Z[]$ is an *iso-negative context* if $Z[X]$ is a negative structure for every negative structure X. These definitions extend straightforwardly to multiple-hole contexts.

The structural connective comma "," is a proxy for conjunction (on the left) and disjunction (on the right), while $<$ is a proxy for exclusion and $>$ is a proxy for implication as we shall show later. Note that, unlike display calculi, $<$ appears only on the left of turnstile at the top level while $>$ appears only on the right at the top level, thus there is no overloading of these structural connectives.

Our first sequent system LBiInt_1 for bi-intuitionistic logic is given in Figure 1. The introduction rules for the logical connectives are the standard ones. The logical rules are non-invertible, since they lose structures or formulas going upwards. Since we have contraction and weakening, on both sides of the sequent, it is possible to formulate invertible logical rules by implicit contraction, as we shall see later. LBiInt_1 is very similar to the display calculus for bi-intuitionistic logic of Goré [11], but with some differences:

- Sequents are of a more restricted form than in display calculus. For example, we do not allow sequents of the form $X \vdash A < B$.

[1] Only recently, we have realised that this may not be the simplest class of structures needed for our calculus. That is, the structural connective $<$ may not be needed, as we can overload the connective $>$, by interpreting it differently in positive and negative contexts, just as the structural connective ',' can be overloaded to represent both disjunction and conjunction in different contexts. The current notation was chosen to conform with Goré's display system, where $<$ is used as a structural proxy for $-\!\!<$.

- The contraction and the weakening rules are applicable to formulae only, not structures in general like in display calculi. But we shall see that the general contraction and weakening are derivable from the "atomic" ones in LBiInt_1, which is not the case for Goré's system.

- The structural rules s_L and s_R are more general than the display postulates in display logic. These rules are derivable in Goré's system, but one needs to use contraction and weakening on structures.

As a consequence of these differences, cut elimination for LBiInt_1 does not necessarily follow from cut elimination for its display calculus counterpart. However, it may be possible to modify Goré's system in such a way that there is a mapping between the cut free proofs of both LBiInt_1 and the modified system. We leave the details of such a connection to future work.

The following two propositions state the admissibility of the general contraction and weakening rules. These can be proved by using the structural rules s_L, s_R, $>$ and $<$.

PROPOSITION 1. **Admissibility of general contraction.** *The two contraction rules shown below are cut-free admissible in $LBiInt_1$:*

$$\frac{X, Y, Y \vdash Z}{X, Y \vdash Z}\ gc_L \qquad \frac{X \vdash Y, Y, Z}{X \vdash Y, Z}\ gc_R$$

Proof. We prove this simultaneously by induction on the size of Y. We show a derivation of the gc_L rule; the case for gc_R is symmetric. The non-trivial case is when $Y = Y_1 < Y_2$. We show that in this case, the contraction rule can be reduced to contractions on smaller structures, which therefore are admissible by the induction hypothesis:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X, (Y_1 < Y_2), (Y_1 < Y_2) \vdash Z}{(Y_1 < Y_2), (Y_1 < Y_2) \vdash X > Z}\ >}{(Y_1 < Y_2), Y_1 \vdash Y_2, (X > Z)}\ s_L}{Y_1, Y_1 \vdash Y_2, Y_2, (X > Z)}\ s_L}{Y_1, Y_1 \vdash Y_2, (X > Z)}\ gc_R}{Y_1 \vdash Y_2, (X > Z)}\ gc_L}{Y_1 < Y_2 \vdash X > Z}\ <}{X, (Y_1 < Y_2) \vdash Z}\ s_R$$

■

PROPOSITION 2. **Admissibility of general weakening.**
The two weakening rules below are cut-free admissible in $LBiInt_1$:

$$\frac{X \vdash Z}{X, Y \vdash Z}\ gw_L \qquad \frac{X \vdash Z}{X \vdash Y, Z}\ gw_R$$

PROPOSITION 3. *The id rule can be restricted to the atomic form:*

$$\frac{}{p \vdash p}\ id$$

So from now on, we assume that all *id* rules are of the atomic form.

3 Cut elimination

Although the proof system LBiInt_1 shares some similarity with traditional Gentzen's systems, cut elimination for LBiInt_1 as presented here follows a different technique from the standard cut elimination technique for sequent calculus. In particular, when the cut formula is not principal in either one of the premises of the cut rule, no cut reductions are required in our cut elimination proof. Instead, the structural rules s_L and s_R allow us to carry the context of one premise of the cut to its other premise resulting in a "proof substitution" akin to the normalisation proofs in natural deduction. Apart from Belnap's cut-elimination proof for display logic, the closest technique we know of is the cut elimination proof for classical logic in a proof system using *deep inference* [3].

For example, suppose π_1 is the cut-free derivation below where the occurrence of p in the root sequent participates in n instances of *id* in the leaves of π_1:

$$\begin{array}{c} \dfrac{}{p \vdash p}\ id \quad \cdots \quad \dfrac{}{p \vdash p}\ id \\ \vdots \\ X_1 \vdash Y_1, p \end{array}$$

Let ξ be the derivation below which ends in an instance of cut on p:

$$\dfrac{\begin{array}{c}\pi_1 \\ X_1 \vdash Y_1, p\end{array} \quad \begin{array}{c}\pi_2 \\ p, X_2 \vdash Y_2\end{array}}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

Then a cut free derivation for $X_1, X_2 \vdash Y_1, Y_2$ can be obtained by replacing the parametric ancestors of the cut formula p in π_1 with the structure $(X_2 > Y_2)$ and replacing the leaves of π_1, where the cut formula p is used, with the derivation π_2. This cut-free derivation is schematically presented as follows:

$$\begin{array}{c} \dfrac{\begin{array}{c}\pi_2 \\ p, X_2 \vdash Y_2\end{array}}{p \vdash (X_2 > Y_2)}\ > \quad \cdots \quad \dfrac{\begin{array}{c}\pi_2 \\ p, X_2 \vdash Y_2\end{array}}{p \vdash (X_2 > Y_2)}\ > \\ \vdots \\ \dfrac{X_1 \vdash Y_1, (X_2 > Y_2)}{X_1, X_2 \vdash Y_1, Y_2}\ s_R \end{array}$$

The reductions for the cases where the cut formula is non-atomic follow essentially the same idea. That is, we substitute the cut formula on one premise of the cut rule with the context of the other premise, and expand this context when the cut formula is used. The only difference is that in the case of non-atomic cut formula, we need to produce extra cuts to make this substitution work. But all the cuts produced are of smaller size, therefore the whole process terminates.

In the following, we write $|A|$ for the *size* of the formula A: the number of logical operators appearing in A. In an instance of a cut rule

$$\frac{X_1 \vdash Y_1, A \quad A, X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

the formula A is called the *cut formula* of the cut instance. The *cut-rank* of the cut instance is $|A|$. Given a derivation π, we denote with $mc(\pi)$ the maximum of the cut-ranks in π. If there are no cuts in π then $mc(\pi) = 0$.

Lemma 4 states the proof substitutions needed to eliminate atomic cuts.

LEMMA 4. *Suppose $p, X \vdash Y$ is cut-free derivable for some fixed p, X and Y. Then for any k-hole anti-positive context $Z_1[\cdots]$ and any l-hole iso-positive context $Z_2[\cdots]$, if $Z_1[p^k] \vdash Z_2[p^l]$ is cut-free derivable, then $Z_1[(X > Y)^k] \vdash Z_2[(X > Y)^l]$ is cut-free derivable.*

Proof. Let π be a cut-free derivation of $p, X \vdash Y$ and let ξ be a cut-free derivation of $Z_1[p^k] \vdash Z_2[p^l]$. We construct a cut-free derivation ξ' of $Z_1[(X > Y)^k] \vdash Z_2[(X > Y)^l]$ by induction on the height of ξ. Most cases follow straightforwardly from the induction hypothesis. The only non-trivial case is when p is active in the derivation, i.e., when ξ ends with an *id* rule or a contraction rule applied to an occurence of p to be substituted for:

- Suppose ξ is

$$\frac{}{Z_1'[p^k], p \vdash p, Z_2'[p^{l-1}]}\ id$$

  Note that the p immediately to the left of the turnstile cannot be part of the p^k by the restrictions on the context $Z_1[\cdots]$. The derivation ξ' is then constructed as follows, where we use double lines to abbreviate derivations:

$$\dfrac{\dfrac{\overset{\pi}{p, X \vdash Y}}{p \vdash (X > Y)}\ >}{\overline{\overline{Z_1'[(X > Y)^k], p \vdash (X > Y), Z_2'[(X > Y)^{l-1}]}}}\ gw_R; gw_L$$

- Suppose ξ is

$$\frac{\overset{\xi_1}{Z_1[p^k] \vdash p, p, Z_2'[p^{l-1}]}}{Z_1[p^k] \vdash p, Z_2'[p^{l-1}]}\ c_R$$

  By induction hypothesis, we have a cut-free derivation ξ_1' of

$$Z_1[(X > Y)^k] \vdash (X > Y), (X > Y), Z_2'[(X > Y)^{l-1}].$$

The derivation ξ' is then constructed as follows:

$$\dfrac{\begin{array}{c}\xi_1' \\ Z_1[(X > Y)^k] \vdash (X > Y), (X > Y), Z_2'[(X > Y)^{l-1}]\end{array}}{Z_1[(X > Y)^k] \vdash (X > Y), Z_2'[(X > Y)^{l-1}]}\ gc_R$$

Note that gw_R and gc_R and gw_L are cut-free derivable in LBiInt_1 by Proposition 1 and Proposition 2. ■

Lemmas 5-9 state the proof substitutions needed for non-atomic cuts.

LEMMA 5. *Let ξ be a derivation of*

$$Z_1[(A_1 \vee A_2)^k] \vdash Z_2[(A_1 \vee A_2)^l]$$

for some k-hole iso-negative context $Z_1[\cdots]$ and l-hole anti-negative context $Z_2[\cdots]$, such that $mc(\xi) < |A_1 \vee A_2|$. Let π_i be a derivation of $X \vdash Y, A_i$, for some $i \in \{1, 2\}$, such that $mc(\pi_i) < |A_1 \vee A_2|$. Then there is a derivation ξ' with $mc(\xi') < |A_1 \vee A_2|$ of

$$Z_1[(X < Y)^k] \vdash Z_2[(X < Y)^l].$$

Proof. By induction on the height of ξ. In the following, we let $A = A_1 \vee A_2$. Most cases follow straightforwardly from the induction hypothesis. The only interesting case is when a left-rule is applied to an occurence of $A_1 \vee A_2$ which is to be replaced by $X < Y$. That is, ξ is

$$\dfrac{\begin{array}{c}\xi_1 \\ Z_1'[A^{k-1}], A_1 \vdash Z_2[A^l]\end{array} \quad \begin{array}{c}\xi_2 \\ Z_1'[A^{k-1}], A_2 \vdash Z_2[A^l]\end{array}}{Z_1'[A^{k-1}], A_1 \vee A_2 \vdash Z_2[A^l]}\ \vee_L$$

By induction hypothesis, we have a derivation ξ_i', for each $i \in \{1, 2\}$, of

$$Z_1'[(X < Y)^{k-1}], A_i \vdash Z_2[(X < Y)^l]$$

with $mc(\xi_i') < |A_1 \vee A_2|$. The derivation ξ' is then constructed as follows:

$$\dfrac{\dfrac{\begin{array}{c}\pi_i \\ X \vdash Y, A_i\end{array}}{X < Y \vdash A_i}\ < \quad \begin{array}{c}\xi_i' \\ Z_1'[(X < Y)^{k-1}], A_i \vdash Z_2[(X < Y)^l]\end{array}}{Z_1'[(X < Y)^{k-1}], (X < Y) \vdash Z_2[(X < Y)^l]}\ cut$$

■

LEMMA 6. *Let ξ be a derivation of*

$$Z_1[(A_1 \wedge A_2)^k] \vdash Z_2[(A_1 \wedge A_2)^l]$$

for some k-hole iso-negative context $Z_1[\cdots]$ and l-hole anti-negative context $Z_2[\cdots]$ with $mc(\xi) < |A_1 \wedge A_2|$. Let π_1 be a derivation of $X \vdash Y, A_1$ and *let π_2 be a derivation of $X \vdash Y, A_2$ with $mc(\pi_1) < |A_1 \wedge A_2|$ and $mc(\pi_2) < |A_1 \wedge A_2|$. Then there is a derivation ξ' with $mc(\xi') < |A_1 \wedge A_2|$ of*

$$Z_1[(X < Y)^k] \vdash Z_2[(X < Y)^l].$$

Proof. Analogous to the proof of Lemma 5. ■

LEMMA 7. *Let ξ be a derivation of*

$$Z_1[(A \to B)^k] \vdash Z_2[(A \to B)^l]$$

for some k-hole iso-negative context $Z_1[\cdots]$ and l-hole anti-negative context $Z_2[\cdots]$ with $mc(\xi) < |A \to B|$. Let π be a derivation of $X, A \vdash B$ with $mc(\pi) < |A \to B|$. Then there is a derivation ξ' with $mc(\xi') < |A \to B|$ of

$$Z_1[X^k] \vdash Z_2[X^l].$$

Proof. By induction on the height of ξ. As in the previous lemmas, the non-trivial case is when ξ ends with $\to_L$ on $A \to B$:

$$\dfrac{\stackrel{\xi_1}{Z_1'[(A \to B)^{k-1}] \vdash A, Z_2[(A \to B)^l]} \quad \stackrel{\xi_2}{Z_1'[(A \to B)^{k-1}], B \vdash Z_2[(A \to B)^l]}}{Z_1'[(A \to B)^{k-1}], A \to B \vdash Z_2[(A \to B)^l]} \to_L$$

By induction hypothesis, we have derivations ξ_1' and ξ_2' respectively of the sequents below where $mc(\xi_1') < |A \to B|$ and $mc(\xi_2') < |A \to B|$:

$$Z_1'[X^{k-1}] \vdash A, Z_2[X^l] \qquad\qquad Z_1'[X^{k-1}], B \vdash Z_2[X^l]$$

In the following, we let V_1 denote $Z_1'[X^{k-1}]$ and V_2 denote $Z_2[X^l]$. The derivation ξ' is constructed as follows:

$$\cfrac{\cfrac{\stackrel{\xi_1'}{V_1 \vdash A, V_2} \quad \cfrac{\stackrel{\pi}{X, A \vdash B} \quad \stackrel{\xi_2'}{V_1, B \vdash V_2}}{V_1, A, X \vdash V_2}\ cut}{V_1, V_1, X \vdash V_2, V_2}\ cut}{V_1, X \vdash V_2}\ gc_L; gc_R$$

■

LEMMA 8. *Let ξ be a derivation of*

$$Z_1[(A \prec B)^k] \vdash Z_2[(A \prec B)^l]$$

for some k-hole iso-negative context $Z_1[\cdots]$ and l-hole anti-negative context $Z_2[\cdots]$ with $mc(\xi) < |A \prec B|$. Let π_1 be a derivation of $X \vdash Y, A$ and let π_2 be a derivation of $X, B \vdash Y$ with $mc(\pi_1) < |A \prec B|$ and $mc(\pi_2) < |A \prec B|$. Then there is a derivation ξ' with $mc(\xi') < |A \prec B|$ of

$$Z_1[(X < Y)^k] \vdash Z_2[(X < Y)^l].$$

Proof. The non-trivial case is when ξ ends with $-\!\!<_L$ on $A-\!\!<B$:

$$\dfrac{\begin{array}{c}\xi_1\\ A \vdash B, Z_2[(A-\!\!<B)^l]\end{array}}{Z_1'[(A-\!\!<B)^{k-1}], A-\!\!<B \vdash Z_2[(A-\!\!<B)^l]}\ -\!\!<_L$$

By induction hypothesis, we have a derivation ξ_1' of

$$A \vdash B, Z_2[(X < Y)^l]$$

with $mc(\xi_1') < |A-\!\!<B|$. Let V denote the structure $Z_2[(X < Y)^l]$. Then ξ' is constructed as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_1\\ X \vdash Y, A\end{array} \quad \dfrac{\begin{array}{c}\xi_1'\\ A \vdash B, V\end{array} \quad \begin{array}{c}\pi_2\\ X, B \vdash Y\end{array}}{A, X \vdash Y, V}\ cut}{X, X \vdash Y, Y, V}\ cut}{X \vdash Y, V}\ gc_L; gc_R}{X < Y \vdash V}\ <}{Z_1'[(X < Y)^{k-1}], X < Y \vdash V}\ gw_L$$

■

LEMMA 9. *Let ξ be a derivation of $Z_1[A^k] \vdash Z_2[A^l]$ where A is a non-atomic formula, $Z_1[\cdots]$ is a k-hole anti-positive context, $Z_2[\cdots]$ is an l-hole iso-positive context, and $mc(\xi) < |A|$. Let π be a derivation of $A, X \vdash Y$ with $mc(\pi) < |A|$. Then there is a derivation ξ' with $mc(\xi') < |A|$ of $Z_1[(X > Y)^k] \vdash Z_2[(X > Y)^l]$.*

Proof. By induction on the height of ξ and case analysis on A. The non-trivial case is when ξ ends with a right-introduction rule on A. That is, in this case, we have $Z_2[A^l] = (Z_2'[A^{l-1}], A)$ for some iso-positive context $Z_2'[\cdots]$. We distinguish several cases depending on A. We show here the cases where A is either a disjunction $C \vee D$, or an implication $C \to D$.

- Suppose $A = C \vee D$ and ξ is the following derivation:

$$\dfrac{\begin{array}{c}\xi_1\\ Z_1[(C \vee D)^k] \vdash Z_2'[(C \vee D)^{l-1}], C\end{array}}{Z_1[(C \vee D)^k] \vdash Z_2'[(C \vee D)^{l-1}], C \vee D}\ \vee_R$$

  By induction hypothesis, we have a derivation ξ_1' of

$$Z_1[(X > Y)^k] \vdash Z_2'[(X > Y)^{l-1}], C$$

  such that $mc(\xi_1') < |C \vee D|$. Let $W_1 = Z_1[(X < Y)^k]$ and let $W_2 = Z_2[(X > Y)^{l-1}]$. Applying Lemma 5 to π and ξ_1', we obtain a derivation θ of

$$(W_1 < W_2), X \vdash Y$$

such that $mc(\theta) < |C \vee D|$. The derivation ξ' is then constructed as follows:

$$\dfrac{\dfrac{\begin{matrix}\xi_1'\\(W_1 < W_2), X \vdash Y\end{matrix}}{W_1 < W_2 \vdash X > Y}\,>}{W_1 \vdash W_2, (X > Y)}\,s_L$$

Clearly, $mc(\xi') < |C \vee D|$.

- Suppose $A = C \to D$ and ξ is

$$\dfrac{\begin{matrix}\xi_1\\Z_1[(C \to D)^k], C \vdash D\end{matrix}}{Z_1[(C \to D)^k] \vdash Z_2'[(C \to D)^{l-1}], C \to D}\,\to_R$$

By induction hypothesis, we have a derivation ξ_1' of

$$Z_1[(X > Y)^k], C \vdash D$$

Then the derivation ξ' is constructed as follows:

$$\dfrac{\begin{matrix}\theta\\Z_1[(X > Y)^k], X \vdash Y\end{matrix}}{Z_1[(X > Y)^k] \vdash Z_2[(X > Y)^{l-1}], (X > Y)}\,s_L$$

where θ is obtained by applying Lemma 7 to π and ξ_1'.

The other cases are treated analogously, using Lemmas 6 and Lemma 8. ■

Finally, cut elimination is proved by simple proof substitutions, the construction of which is given by the preceding lemmas.

THEOREM 10. *If $X \vdash Y$ is LBiInt$_1$-derivable then it is also cut-free derivable.*

Proof. As typical in cut elimination proofs, we remove topmost cuts in succession. Let π be a derivation of LBiInt$_1$ with a topmost cut instance

$$\dfrac{\begin{matrix}\pi_1\\X_1 \vdash Y_1, A\end{matrix} \quad \begin{matrix}\pi_2\\X_2, A \vdash Y_2\end{matrix}}{X_1, X_2 \vdash Y_1, Y_2}\,cut$$

Note that π_1 and π_2 are both cut-free since this is a topmost instance in π. We use induction on the size of A to eliminate this topmost instance of cut.

If A is an atomic formula p then the cut free derivation is constructed as follows where ξ is obtained from applying Lemma 4 to π_1 and π_2:

$$\dfrac{\begin{matrix}\xi\\X_1 \vdash Y_1, (X_2 > Y_2)\end{matrix}}{X_1, X_2 \vdash Y_1, Y_2}\,s_R$$

If A is non-atomic, using Lemma 9 we get the following derivation θ:

$$\dfrac{\begin{array}{c}\xi \\ X_1 \vdash Y_1, (X_2 > Y_2)\end{array}}{X_1, X_2 \vdash Y_1, Y_2}\ s_R$$

We have $mc(\theta) < |A|$ by Lemma 9, therefore by induction hypothesis, we can remove all the cuts in θ to get a cut-free derivation of $X_1, X_2 \vdash Y_1, Y_2$. ■

4 Soundness and completeness of $\mathbf{LBiInt}_1$

To prove soundness, we first define an interpretation of sequents as formulae as shown below using two extra logical constants $\top$ and $\bot$ which were not part of our original language for formulae. But it is easy to show that the system LBiInt_1 can be extended to cover these constants in the obvious way.

DEFINITION 11. The two mutual-recursively functions τ_N and τ_P respectively translate a negative and positive structure into a BiInt-formula:

$$\begin{array}{lll lll}
\tau_N(\emptyset) & = & \top & \tau_P(\emptyset) & = & \bot \\
\tau_N(A) & = & A & \tau_P(A) & = & A \\
\tau_N(X, Y) & = & \tau_N(X) \wedge \tau_N(Y) & \tau_P(X, Y) & = & \tau_P(X) \vee \tau_P(Y) \\
\tau_N(X < Y) & = & \tau_N(X) \prec \tau_P(Y) & \tau_P(X > Y) & = & \tau_N(X) \to \tau_P(Y)
\end{array}$$

We assume the reader is familiar with the Kripke semantics [22, 12] for BiInt using a binary reflexive and transitive relation $\leq$, which extends the usual Kripke semantics for Int with an extra clause for exclusion given below:

$$w \Vdash A \prec B \quad \text{iff} \quad \exists v \leq w. v \Vdash A \text{ and } v \nVdash B.$$

THEOREM 12. **Soundness.** *Every LBiInt_1-derivable BiInt formula is valid.*

Proof. We show that for every rule ρ of LBiInt_1

$$\dfrac{X_1 \vdash Y_1 \quad \cdots \quad X_n \vdash Y_n}{X \vdash Y}\ \rho$$

the following holds: if for every $i \in \{1, \dots, n\}$, the formula $\tau_N(X_i) \to \tau_P(Y_i)$ is valid then the formula $\tau_N(X) \to \tau_P(Y)$ is valid. Since the formula-translation $(\tau_N(X) \wedge A) \to (A \vee \tau_P(Y))$ of the *id* rule is obviously valid, it then follows that every formula derivable in LBiInt_1 is also valid.

For all the rules of LBiInt_1, except $<$ and $\prec_L$, we can show the stronger statement that the following formula is valid:

$$[(\tau_N(X_1) \to \tau_P(Y_1)) \wedge \cdots \wedge (\tau_N(X_n) \to \tau_P(Y_n))] \to (\tau_N(X) \to \tau_P(Y)).$$

Soundness of $<$ and $\prec_L$ are shown in the standard way, by reasoning about the forcing relation $\Vdash$ and the reflexive and transitive relation $\leq$. ■

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

$$\frac{}{X, A \vdash A, Y}\ id \qquad \frac{\{X_1\}, X_2 \vdash Y_2}{X_1 \vdash Y_1, (X_2 > Y_2)} > \ \{X_1\} \not\subseteq \{X_2\}$$

$$\frac{X_2 \vdash Y_2, \{Y_1\}}{X_1, (X_2 < Y_2) \vdash Y_1} < \ \{Y_1\} \not\subseteq \{Y_2\}$$

$$\frac{X, B_1 \wedge B_2, B_i \vdash Y}{X, B_1 \wedge B_2 \vdash Y} \wedge_L \qquad \frac{X \vdash A \wedge B, A, Y \quad X \vdash A \wedge B, B, Y}{X \vdash A \wedge B, Y} \wedge_R$$

$$\frac{X, A \vee B, A \vdash Y \quad X, A \vee B, B \vdash Y}{X, A \vee B \vdash Y} \vee_L \qquad \frac{X \vdash B_1 \vee B_2, B_i, Y}{X \vdash B_1 \vee B_2, Y} \vee_R$$

$$\frac{X, A \to B \vdash A, Y \quad X, A \to B, B \vdash Y}{X, A \to B \vdash Y} \to_L \qquad \frac{X \vdash Y, A \to B, B}{X \vdash Y, A \to B} \to_{R1}$$

$$\frac{X, A \prec B, A \vdash Y}{X, A \prec B \vdash Y} \prec_{L1} \qquad \frac{X \vdash A, A \prec B, Y \quad X, B \vdash A \prec B, Y}{X \vdash A \prec B, Y} \prec_R$$

$$\frac{A \vdash B, \{Y\}, (X, A \prec B > Y)}{X, A \prec B \vdash Y} \prec_{L2} \qquad \frac{(X < Y, A \to B), \{X\}, A \vdash B}{X \vdash Y, A \to B} \to_{R2}$$

Figure 2. System LBiInt_2

Completeness is shown by embedding Rauszer's sequent calculus G1 [21] for BiInt into LBiInt_1. The calculus G1 contains the cut rule, and is shown to be complete by Rauszer [21]. The encoding of G1 into LBiInt_1 is obvious since all the rules of G1 are easily derivable from the rules of LBiInt_1.

THEOREM 13. **Completeness.** *Every BiInt-valid formula is* $\textit{LBiInt}_1$*-derivable.*

5 Proof search

System LBiInt_1 is not suitable for proof search, since the structural rules s_L, s_R, $<$ and $>$ can easily lead to non-termination if applied naively. In addition, we also have the usual problems with the contraction rules since they can be applied *ad infinitum*. We now present a refined version of LBiInt_1, called LBiInt_2, in which all the structural rules, except for $>$ and $<$, are absorbed into logical rules. The resulting calculus, for the intuitionistic fragment, resembles contraction-free calculi for the traditional Gentzen systems for intuitionistic logic, e.g., the system **G3i** in [26]. The underlying idea behind LBiInt_2 is that the right-introduction rule for $\to$ and the left introduction rule for $\prec$ act as an instruction to store the current state (of proof search), and the rules $>$ and $<$ act as an instruction to restart previously stored computation states.

The inference rules for LBiInt_2 are given in Figure 2 using the notation

$\{X\}$ to denote the *set of formulae* that appear at the top-level of X:

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}.$$

Intuitively, the set $\{X\}$ denotes X with all the substructures of the form $Y > Z$ or $Y < Z$ removed. For example, if X is $(A, B, (C > D))$, then $\{X\}$ is the set $\{A, B\}$. The right introduction rule for $\to$ splits into two rules: $\to_{R1}$ and $\to_{R2}$. The $\to_{R1}$ rule is strictly speaking not necessary as it can be derived using $\to_{R2}$ and $<$. However, it is useful in our proof search strategy which relies on a *saturation* process on sequents, as we shall see later. The rule $\to_{R2}$ incorporates some features of the structural rule s_L. The left introduction rule for $-\!\!<$ splits also into two rules with roles symmetric to those for $\to$.

5.1 A terminating proof search strategy

We classify the rules of LBiInt_2 into three groups:

Static Rules: $= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \to_L, -\!\!<_R, -\!\!<_{L1}, \to_{R1}\}$;

Jump Rules: $= \{-\!\!<_{L2}, \to_{R2}\}$; and

Return Rules: $= \{<, >\}$.

We call a sequence of static rule applications a *saturation.*

DEFINITION 14. A sequent $X \vdash Y$ is *saturated* iff it satisfies 1-8, and is *strongly saturated* iff it additionally satisfies 9:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$
5. If $A \vee B \in \{Y\}$ then $A \in \{Y\}$ and $B \in \{Y\}$
6. If $A \to B \in \{X\}$ then $A \in \{Y\}$ or $B \in \{X\}$
7. If $A -\!\!< B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{X\}$
8. If $A \to B \in \{Y\}$ then $B \in \{Y\}$ If $A -\!\!< B \in \{X\}$ then $A \in \{X\}$
9. If $A \to B \in \{Y\}$ then $A \in \{X\}$ If $A -\!\!< B \in \{X\}$ then $B \in \{Y\}$.

We say that an LBiInt_2 rule ρ is applicable to a sequent $\gamma_0 = X_0 \vdash Y_0$ if for every premise $X_i \vdash Y_i$ of ρ, $\{X_i\} \nsubseteq \{X_0\}$ or $\{Y_i\} \nsubseteq \{Y_0\}$. Thus only jump and return rules are applicable to saturated sequents.

DEFINITION 15 (Proof search strategy).
Function Prove
Input: sequent γ_0
Output: *true* (i.e. γ_0 is derivable) or *false* (i.e. γ_0 is not derivable)

1. If *id* is applicable to γ_0 then return *true*
2. Else if a static rule ρ is applicable to γ_0 then
   (a) Let $\gamma_1, \cdots, \gamma_n$ be the premises of ρ obtained from γ_0
   (b) Return $\bigwedge_{i=1}^{n} Prove(\gamma_i)$
3. Else if $Prove(\gamma_1) = true$ for some premise instance γ_1 obtained from γ_0 by applying $\rho \in \{ -\!\!<_{L2}, \to_{R2}, <, > \}$ backward then return *true*
4. Else return *false*.

We shall show that the search strategy given in Definition 15 terminates, if given an input sequent with a certain simple structure, which is defined in the following.

DEFINITION 16. A structure is a *flat structure* if it contains no occurrences of the structural connectives $>$ and $<$. We use Γ and Δ to stand for flat structures since flat structures can be viewed as sets of formulae. The set of (positive/negative) *linear structures* is the smallest set of structures that satisfies the following:

1. Every flat structure is a linear structure.
2. If X is a positive linear structure and Γ is a flat structure, then $\Gamma < X$ is a negative linear structure.
3. If X is a negative linear structure and Δ is a flat structure, then $X > \Delta$ is a positive linear structure.
4. If X is a positive (negative) linear structure and Δ is a flat structure, then (X, Δ) is a positive (resp. negative) linear structure.

A sequent $X \vdash Y$ is a *linear sequent* if either X is a flat structure and Y is a positive linear structure, or X is a negative linear structure and Y is a flat structure.

The intuition of Definition 16 is that a linear sequent $X \vdash Y$ can take the form $(X' < Y'), \Gamma \vdash \Delta$, or $\Gamma \vdash \Delta, (X'' > Y'')$, or $\Gamma \vdash \Delta$, where $X' < Y'$ and $X'' > Y''$ store the sequent corresponding to the previous state of computation, and Γ and Δ are sets of formulae.

LEMMA 17. *Let $X \vdash Y$ be a linear sequent. Then for every LBiInt$_2$-derivation π of $X \vdash Y$, every sequent in π is a linear sequent.*

Proof. Given a derivation π of a linear sequent $X \vdash Y$, we show by induction on the length of π that every sequent in π is a linear sequent. This is straightforward by showing that in every rule of LBiInt$_2$, if the conclusion of the rule is a linear sequent, then every premise of the rule is also a linear sequent, which can be verified by inspection of the rules of LBiInt$_2$. ■

Note that as a consequence of Lemma 17, every sequent that arises during proof search for a linear sequent $X \vdash Y$, using the search procedure given in Definition 15, is a linear sequent.

We now define a translation from linear sequents to linked lists, consisting of nodes that are pairs of sets of formulae, linked by labels marked either R or R^{-1}.

DEFINITION 18.

$$\begin{array}{lcl} list(\Gamma \vdash \Delta) & = & \langle \Gamma, \Delta \rangle \\ list((X' < Y'), \Gamma \vdash \Delta) & = & list(X' \vdash Y') \;\; R \;\; \langle \Gamma, \Delta \rangle \\ list(\Gamma \vdash \Delta, (X'' > Y'')) & = & list(X'' \vdash Y'') \;\; R^{-1} \;\; \langle \Gamma, \Delta \rangle \end{array}$$

We write $length(L)$ to mean the number of nodes in the list L.

COROLLARY 19. *A backward $LBiInt_2$ rule application to a linear sequent $X \vdash Y$ can be viewed as an operation on $list(X \vdash Y)$, where the conclusion (resp. premise) is the list before (resp. after) the operation. The jump rules append a node to the list, and the static rules saturate the end node. The return rules remove a node from the end of the list, and add subformulae to the penultimate node.*

For example, below left is is an instance of $\rightarrow_{R2}$ with the corresponding list structures of the premise and conclusion on the right:

$$\frac{(C < B, A \rightarrow B), C, A \vdash B}{C \vdash B, A \rightarrow B} \rightarrow_{R2} \qquad \frac{\langle \{C\}, \{B, A \rightarrow B\} \rangle \;\; R \;\; \langle \{C, A\}, \{B\} \rangle}{\langle \{C\}, \{B, A \rightarrow B\} \rangle}$$

We now define a metric that we will use in the main termination proof.

DEFINITION 20. The degree of a formula is:

$$\begin{array}{rcl} deg(p) & = & 0 \\ deg(A \wedge B) = deg(A \vee B) & = & max(deg(A), deg(B)) \\ deg(A \rightarrow B) = deg(A \prec B) & = & 1 + max(deg(A), deg(B)). \end{array}$$

The degree of a sequent is:

$$\begin{array}{rcl} deg_L(X \vdash Y) & = & max\{deg(A) \mid A \in \{X\}\} \\ deg_R(X \vdash Y) & = & max\{deg(B) \mid B \in \{Y\}\} \\ deg(X \vdash Y) & = & max(deg_L(X \vdash Y), deg_R(X \vdash Y)). \end{array}$$

Note that only logical connectives contribute to these metrics.

We denote with $sf(A)$ the set of subformulae of A, and

$$sf(\Gamma) = \bigcup_{A \in \Gamma} sf(A)$$

the set of subformulae of Γ. In the following, we assume that the initial input to the search procedure Prove is a linear sequent $\Gamma_0 \vdash \Delta_0$, and we define $m = |sf(\Gamma_0 \cup \Delta_0)|$.

LEMMA 21. *Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, $list(X \vdash Y)$ can be extended at most $\mathcal{O}(m^2)$ times.*

Proof. We show that the number of jump rule applications is bounded by $\mathcal{O}(m^2)$.

First, we show that there can be at most m consecutive jumps in the same direction. In the forward case, consider a backwards application of $\rightarrow_{R2}$ with principal formula $A \rightarrow B$. After this application, A will be added to the LHS of the sequent, and remain on the LHS during saturation and forward jumps. Should $A \rightarrow B$ reappear on the RHS, B will be added to the RHS by the $\rightarrow_{R1}$ rule during saturation, so a repeated application of $\rightarrow_{R2}$ to $A \rightarrow B$ will be blocked by the general blocking condition. Thus since the number of $\rightarrow$-formulae is bounded by m and we can only jump on each $\rightarrow$-formula once, there can be at most m consecutive forward jumps. The backward case is symmetric.

We now show that we can switch direction at most m times. Consider a direction switch, e.g., a forward jump using $\rightarrow_{R2}$ followed by a backward jump $-\!\!<_{L2}$ (the other case is symmetric), and any static rule applications in between. Let γ_0 and γ_1 be the conclusion and premise of the $\rightarrow_{R2}$ rule respectively, and let γ_2 and γ_3 be the conclusion and premise of the $-\!\!<_{R2}$ rule respectively, as shown below:

$$
\dfrac{\dfrac{\vdots}{\gamma_3 = C \vdash D, \Delta, ((X < Y, A \rightarrow B), \Gamma, C -\!\!< D > \Delta)}}{\gamma_2 \;=\; (X < Y, A \rightarrow B), \Gamma, C -\!\!< D \vdash \Delta} \; -\!\!<_{L2}
$$

$$
\vdots
$$

$$
\dfrac{\gamma_1 \;=\; (X < Y, A \rightarrow B), \{X\}, A \vdash B}{\gamma_0 \;=\; X \vdash Y, A \rightarrow B} \; \rightarrow_{R2}
$$

$$
\vdots
$$

Let $d_0 = deg(\gamma_0)$. We will show that $deg(\gamma_3) \leq d_0 - 1$. By inspection of the rules and Definition 20, we have the following:

$$
\begin{array}{l}
deg_L(\gamma_1) \leq d_0 \\
deg_R(\gamma_1) \leq d_0 - 1 \\
deg_L(\gamma_2) = deg_L(\gamma_1) \leq d_0 \\
deg_R(\gamma_2) \leq max(deg_L(\gamma_1) - 1, deg_R(\gamma_1)) = d_0 - 1 \\
deg_L(\gamma_3) \leq deg_L(\gamma_2) - 1 = d_0 - 1 \\
deg_R(\gamma_3) \leq deg(\gamma_2) - 1 = d_0 - 1
\end{array}
$$

Therefore $deg(\gamma_3) = max(deg_L(\gamma_3), deg_R(\gamma_3)) \leq d_0 - 1$.

After a direction switch, we can again make at most m jumps in one direction. Therefore the total number of jump rule applications is bounded by $\mathcal{O}(m^2)$. ■

LEMMA 22. *Let $X \vdash Y$ be any sequent encountered during proof search. Then the saturation process for $X \vdash Y$ terminates after $\mathcal{O}(m)$ steps.*

Proof. Every application of a static rule adds a subformula of $sf(\Gamma_0 \cup \Delta_0)$ to the sequent. After at most m applications of static rules, the sequent will contain all subformulae of the original sequent, and hence will be saturated. ■

THEOREM 23. *The proof search strategy of Definition 15 terminates.*

Proof. Suppose for a contradiction that the strategy does not terminate. From Lemmas 21 and 22, we can conclude that the only way to get non-termination is for the jump and return rules to repeatedly create and remove nodes.

The length of the list is at least 1 because the first node cannot be removed. We call a node that cannot be removed *stable*. Every time a return rule removes node i from the list, it adds one or more new subformulae of $\Gamma_0 \cup \Delta_0$ to node $i-1$. After at most m such updates, node $i-1$ will contain every subformula, and the return rules will no longer be applicable to node i because their side conditions will not hold. Then node $i-1$ will become stable. Eventually all nodes will become stable, and the return rules will no longer be applicable to the end of the list. Contradiction. ■

5.2 Soundness and completeness of LBiInt$_2$

For soundness of LBiInt$_2$ we show that every LBiInt$_2$ rule is derivable in LBiInt$_1$.

THEOREM 24. **Soundness of LBiInt$_2$.** *If the sequent $X \vdash Y$ is derivable in LBiInt$_2$ then it is also derivable in LBiInt$_1$.*

To prove completeness of LBiInt$_2$, we take a detour through Buisman and Goré's calculus GBiInt [5, 13]. That is, we show that every derivation of a formula in GBiInt can be translated into a derivation of the same formula in LBiInt$_2$. Due to space limits, we give here only the outline of the translation. The full proof is available in the extended version of the paper.

The GBiInt proof system makes use of two forms of sequents:

$$\mathcal{S}\ \Gamma \lhd \Delta\ \mathcal{P} \qquad\qquad \mathcal{S}\ \Gamma \rhd \Delta\ \mathcal{P}.$$

The former denotes a *refutation* of the sequent, whereas the latter denotes its provability. In both sequents, Γ and Δ are sets of formulas, and $\mathcal{S}$ and $\mathcal{P}$ are sets of sets of formulas. $\mathcal{S}$ and $\mathcal{P}$ are called *variables* in [5], and they contain a counter model for "failed" proof search, i.e., a refutation. Proof search in GBiInt proceeds by saturation of sequents, until they become strongly saturated, at which point we can close the search by the following rule:

$$\frac{}{\{\Gamma\}\ \Gamma \lhd \Delta\ \{\Delta\}}\ Ret$$

This rule applies only in case where $\Gamma \vdash \Delta$ is strongly saturated. It basically says that there is no proof for this sequent, since one can construct a counter-model for this sequent. The information about the counter-model is then stored in the variables of the sequents, and passed back to a previous search point. To see how this information is used, consider the following (simplified) right introduction rule for implication in GBiInt (which is also called $\rightarrow_{R2}$):

$$\frac{\mathcal{S}\ \Gamma, A \lhd B\ \mathcal{P} \quad \Gamma \rhd \Delta, A \rightarrow B, P_1 \quad \cdots \quad \Gamma \rhd \Delta, A \rightarrow B, P_n}{\Gamma \rhd \Delta, A \rightarrow B} \rightarrow_{R2}$$

where $\mathcal{P} = \{P_1, \ldots, P_n\}$ and $P_i \not\subseteq \Delta$. Here, in order to simplify presentation, we consider an instance of the GBiInt rule $\rightarrow_{R2}$ where the lower sequent contains no variables. Intuitively, proof search using this rule tries to decompose $A \rightarrow B$ and see if it can find a counter model (the left premise). If the left premise does return some counter-model, i.e., $\mathcal{P}$ is not empty, then continue the search (right-premise) using the extra information gathered from the counter-model. Naturally, the formula $A \rightarrow B$ needs no further decomposition in subsequent proof search.

The translation from GBiInt to LBiInt_2 is surprisingly natural and uncovers a nice duality between the two calculi. To illustrate the idea behind the translation, consider the following derivation scheme in GBiInt, where the last rule is $\rightarrow_{R2}$. Let π be a GBiInt derivation:

$$\begin{array}{c} \dfrac{}{\{S_1\}\ S_1 \lhd P_1\ \{P_1\}}\ Ret \quad \cdots \quad \dfrac{}{\{S_n\}\ S_n \lhd P_n\ \{P_n\}}\ Ret \\ \vdots \\ \{S_1, \ldots, S_n\}\ \Gamma, A \lhd B\ \{P_1, \ldots, P_n\} \end{array}$$

with n-instances of Ret in its leaves. Let ξ_i, for each $i \in \{1, \ldots, n\}$, be a GBiInt derivation of $\Gamma \rhd \Delta, A \rightarrow B, P_i$ and let ξ be the GBiInt derivation

$$\frac{\overset{\pi}{\mathcal{S}\ \Gamma, A \lhd B\ \mathcal{P}} \quad \overset{\xi_1}{\Gamma \rhd \Delta, A \rightarrow B, P_1} \quad \cdots \quad \overset{\xi_n}{\Gamma \rhd \Delta, A \rightarrow B, P_n}}{\Gamma \rhd \Delta, A \rightarrow B} \rightarrow_{R2}$$

where $\mathcal{S} = \{S_1, \ldots, S_n\}$ and $\mathcal{P} = \{P_1, \ldots, P_n\}$. Then we construct an LBiInt_2 derivation of $\Gamma \vdash \Delta, A \rightarrow B$ as follows: We first translate each ξ_i into an LBiInt_2 derivation ξ_i', and then "plug" in the derivation ξ_i to the i-th instance of Ret in π. This can be done by storing the context $\Gamma < \Delta$ in the left-hand side of the sequents appearing in π, and restoring them at the leaves which are instances of Ret. The following schematic derivation shows this process of storing and restoring of context:

$$\begin{array}{c} \dfrac{\overset{\xi_1'}{\Gamma \vdash \Delta, A \rightarrow B, P_1}}{(\Gamma < \Delta, A \rightarrow B), S_1 \vdash P_1} < \quad \cdots \quad \dfrac{\overset{\xi_n'}{\Gamma \vdash \Delta, A \rightarrow B, P_n}}{(\Gamma < \Delta, A \rightarrow B), S_n \vdash P_n} < \\ \vdots \\ \dfrac{(\Gamma < \Delta, A \rightarrow B), \Gamma, A \vdash B}{\Gamma \vdash \Delta, A \rightarrow B} \rightarrow_{R_2} \end{array}$$

Notice that while proof search in GBiInt involves backtracking and passing back information from failed attempts, in LBiInt_2 we simply go forward and restart a stored computation state when proof search in the current state (i.e., top level sequent) can no longer progress.

THEOREM 25. *If A is derivable in GBiInt then A is also derivable in LBiInt$_2$.*

COROLLARY 26. *LBiInt$_2$ is sound and complete with respect to BiInt.*

Since procedure Prove from Defn 15 mimics this translation, we have:

THEOREM 27. *Any formula A is LBiInt$_2$-derivable if and only if* Prove($\emptyset \vdash A$) *returns true.*

Thus LBiInt$_2$ gives us a decision procedure for BiInt. The fact that BiInt is decidable is already known:

THEOREM 28. *The decision problem for BiInt is* PSPACE*-complete.*

Proof. We first show that BiInt is in PSPACE. To do this, we can easily extend the polynomial Gödel translation of intuitionistic logic into the modal logic S4 [10], to a translation of BiInt into the tense modal logic KtS4. Since KtS4 is in PSPACE [24], we know that BiInt is also in PSPACE.

We now show that BiInt is PSPACE-hard. To do this, we use the fact that BiInt is an extension of Int, which is PSPACE-complete [25], and hence PSPACE-hard. Therefore BiInt is PSPACE-complete. ■

6 Related and future work

We know of only two other sequent calculi for BiInt, one due to Pinto and Uustalu [28], and the other due to Crolard [7]. But the labelled calculus of Pinto and Uustalu is still not available in full detail, and Crolard's calculus fails cut-elimination for the same reasons as does Rauszer's original calculus [21]. The display calculus for BiInt due to Goré [12] and its more recent extension due to Wansing [30] both have a purely syntactic cut-elimination proof of course. But as stated in our introduction, neither of these is really suitable for proof-search.

Hypersequents have been used for many modal [1], intutionistic and intermediate logics [19, 6]. Similarly, Dosen's "higher level" sequents [8] can cater for many different logics in one (cut-free) setting: for example, both intuitinistic logic, classical logic and modal logics S4 and S5. But we know of no actual work which uses either framework for intuitionistic logics with a "converse" modality like BiInt.

The closest calculus to ours appears to be the sequent calculus for the Lambek-Grishin logic **LG** of Moortgat [20], for which he proves a purely syntactic cut-elimination result. Briefly, the logic **NL** is a substructural intuitionistic logic which has a single non-associative and non-commutative conjunction $\otimes$, a single non-associative and non-commutative disjunction $\oplus$, and two implication connectives $A \to B$ and $B \leftarrow A$. The logic **LG** is an extension of **NL** with two exclusion connectives $A \prec B$ and $B \succ A$, and extra "mixed associativity" principles like $((A \prec B) \otimes C) \to ((A \otimes C) \prec B)$ amongst others. Moortgat uses invertible residuation rules which are similar to our rules $s_L, s_R, <, >$ since both logics permit such "flip-flopping". But the lack of associativity and commutativity of $\oplus$ and $\otimes$ means there is less non-determinism in the calculus in terms of proof-search. Indeed, the decision procedure for **LG** obtained by Moortgat runs in polynomial time while it is known that the decision problem for BiInt is PSPACE-complete.

The main difference between Int and BiInt is of course the exclusion connective $-\!\!<$, whose Kripke semantic clause "looks backward against" the accessibility relation $\leq$. Thus it makes sense to look at sequent-like calculi for the modal companions of BiInt, namely the tense logics Kt and KtS4.

There are many sequent-like calculi for the related normal modal logics Kt, its reflexive and transitive cousin KtS4, or even just good old classical modal logic S5 [27, 17, 1, 14, 15, 16, 4]. In each such calculus there is a rule (or rules) which allow us to "return" to previously seen worlds when the rules are viewed from the perspective of counter-model construction. These calculi can be broadly separated into two groups: those which use two-sided sequents, and those which use a negation-normal-form to make do with right-sided sequents. It should be possible to extend most of the former calculi to handle BiInt, but it is unlikely that the latter can be so extended since BiInt has no negation-normal-form theorem.

On this note, we intend to consider the following ideas for future work. Suppose we posit two additional structural connectives $\circ$ and $\bullet$ with $\circ$ standing as a proxy for $\langle F\rangle/[F]$ on the left/right, and $\bullet$ standing as a proxy for $\langle P\rangle/[P]$ on the left/right. Are the following rules enough to give us a cut-free calculus for tense logic Kt which is also amenable for backward proof-search:

$$\frac{\Gamma, A, \vdash \circ(X > Y), \Delta}{X, [P]\Gamma, \langle P\rangle A \vdash Y, \langle P\rangle\Delta}\ \langle P\rangle_L \qquad \frac{\Gamma, \circ(X < Y) \vdash A, \Delta}{X, [P]\Gamma \vdash Y, [P]A, \langle P\rangle\Delta}\ [P]_R$$

$$\frac{\Gamma, A, \vdash \bullet(X > Y), \Delta}{X, [F]\Gamma, \langle F\rangle A \vdash Y, \langle F\rangle\Delta}\ \langle F\rangle_L \qquad \frac{\Gamma, \bullet(X < Y) \vdash A, \Delta}{X, [F]\Gamma \vdash Y, [F]A, \langle F\rangle\Delta}\ [F]_R$$

As we have noted in the introduction, the calculus LBiInt_1 (and its derivative LBiInt_2) is motivated by display calculi. It can be seen as an attempt to tame proof search in display calculi. In this preliminary work, we have been able to derive more proof-search friendly calculi essentially by constraining the use of the display postulates of display calculi. However, there is still a methodological gap in our results. We have not been able to show a direct relation between LBiInt_2 and LBiInt_1: that is, we still need the help of an "external" calculus GBiInt for our completeness result for LBiInt_2. It is important that this gap be closed in order to generalise these results beyond bi-intuitionistic logic. Our ultimate goal is to obtain a systematic way to "sequentialize" a given display calculus to one with nested sequents, and derive a proof search strategy for the latter.

Acknowledgments. We thank the anonymous referees for their comments on an earlier draft of the paper.

BIBLIOGRAPHY

[1] A. Avron. The method of hypersequents in proof theory of propositional non-classical logics. In C Steinhorn W Hodges, M Hyland and J Truss, editors, *Logic: Foundations to Applications*, pages 1–32. Oxford Science Publications, 1996.

[2] N. Belnap. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

[3] K. Brünnler. Atomic cut elimination for classical logic. In *CSL*, volume LNCS 2803, pages 86–97. Springer, 2003.

[4] K. Brünnler. Deep sequent systems for modal logic. In G. Governatori et al, editor, *Advances in Modal Logic 6*, pages 107–119. College Publications, 2006.

[5] L. Buisman and R. Goré. A cut-free sequent calculus for bi-intuitionistic logic. In N Olivetti, editor, *TABLEAUX*, volume 4548 of *LNCS*, pages 90–106. Springer, 2007.

[6] A. Ciabattoni and M. Ferrari. Hypersequent calculi for some intermediate logics with bounded Kripke models. *Journal of Logic and Computation*, 11(2):283–294, 2001.

[7] T. Crolard. Subtractive logic. *Theoretical Computer Science*, 254(1–2):151–185, 2001.

[8] K. Došen. Sequent-systems for modal logic. *Journal of Symbolic Logic*, 50(1):149–169, 1985.

[9] S. Feferman, editor. *Gödel's Collected Works.* Oxford University Press, Oxford, 1980.

[10] K. Gödel. Eine interpretation des intuitionistischen aussagenkalkuls. *Ergebnisse eines Mathematischen Kolloquiums.*, 4:39–40, 1933. English translation in [9].

[11] R. Goré. Substructural logics on display. *LJIGPL*, 6(3):451–504, 1998.

[12] R. Goré. Dual intuitionistic logic revisited. In Roy Dyckhoff, editor, *Proc. TABLEAUX-2000*, volume LNAI 1847, pages 252–267. Springer, 2000.

[13] R. Goré and L. Postniece. Combining derivations and refutations for cut-free completeness in bi-intuitionistic logic. *Journal of Logic and Computation.* To appear. Available via `http://users.rsise.anu.edu.au/~linda/BiIntLong.pdf`.

[14] A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Analytic Tableaux and Related Methods*, volume 1071 of *LNAI*, pages 210–225, 1996.

[15] A. Indrzejczak. Multiple sequent calculus for tense logics. International Conference on Temporal Logic, Leipzig 2000. 93–104.

[16] A. Indrzejczak. Cut-free double sequent calculus for S5. *LJIGPL*, 6(3):505–516, 1998.

[17] R. Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53:119–135, 1994.

[18] M. Kracht. Power and weakness of the modal display calculus. In H. Wansing, editor, *Proof Theory of Modal Logics*, pages 92–121. Kluwer, 1996.

[19] A. Ciabattoni, M. Baaz and C. G. Fermüller. Hypersequent calculi for Gödel logics — a survey. *Journal of Logic and Computation*, 13:835–861, 2003.

[20] M. Moortgat. Symmetries in natural language syntax and semantics: The Lambek-Grishin calculus. In *Logic, Language, Information and Computation: Proc. 14th International Workshop WoLLIC 2007*, LNCS 4576, pages 264–284. Springer, 2007.

[21] C. Rauszer. A formalization of the propositional calculus of H-B logic. *Studia Logica*, 33:23–34, 1974.

[22] C. Rauszer. An algebraic and Kripke-style approach to a certain extension of intuitionistic logic. *Dissertationes Mathematicae*, 168, 1980.

[23] G. Restall. *An Introduction to Substructural Logics.* Routledge, New York, 2000.

[24] E. Spaan. The complexity of propositional tense logics. In M. de Rijke, editor, *Diamonds and Defaults*, pages 287–309. Kluwer Academic Publishers, 1993.

[25] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9:67–72, 1979.

[26] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 1996.

[27] K. Trzesicki. Gentzen-style axiomatization of tense logic. *Bulleting of the Section of Logic*, 13(2):75–84, 1984.

[28] T. Uustalu and L. Pinto. Days in logic '06 conference abstract. Online at `http://www.mat.uc.pt/~kahle/dl06/tarmo-uustalu.pdf`, accessed on 27th October, 2006.

[29] H. Wansing. Modal tableaux based on residuation. *Journal of Logic and Computation*, 7(6):719–731, 1997.
[30] H. Wansing. Constructive negation, implication and co-implication. *Manuscript*, March:to appear, 2007.

Rajeev Goré
Computer Sciences Laboratory,
The Australian National University
Australia
Rajeev.Gore@anu.edu.au

Linda Postniece
Computer Sciences Laboratory,
The Australian National University
Australia
Linda.Postniece@anu.edu.au

Alwen Tiu
Computer Sciences Laboratory,
The Australian National University
Australia
Alwen.Tiu@anu.edu.au