Deep inference in bi-intuitionistic logic

Linda Postniece

Logic and Computation Group
College of Computer Science and Engineering
The Australian National University
Linda.Postniece@anu.edu.au

Abstract. Bi-intuitionistic logic is the extension of intuitionistic logic with exclusion, a connective dual to implication. Cut-elimination in bi-intuitionistic logic is complicated due to the interaction between these two connectives, and various extended sequent calculi, including a display calculus, have been proposed to address this problem.

In this paper, we present a new extended sequent calculus DBiInt for bi-intuitionistic logic which uses nested sequents and "deep inference", i.e., inference rules can be applied at any level in the nested sequent. We show that DBiInt can simulate our previous "shallow" sequent calculus LBiInt. In particular, we show that deep inference can simulate the residuation rules in the display-like shallow calculus LBiInt. We also consider proof search and give a simple restriction of DBiInt which allows terminating proof search. Thus our work is another step towards addressing the broader problem of proof search in display logic.

1 Introduction

Bi-intuitionistic logic (BiInt) is the extension of intuitionistic logic with exclusion $-\!\!<$ (also known as "subtraction" and "co-implication"), a connective dual to implication $\to$. In a sequent calculus setting, the left-introduction rule for exclusion is dual to the right introduction rule for implication:

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

BiInt was first studied by Rauszer as a Hilbert calculus with algebraic and Kripke semantics [13]. More recently, Crolard has investigated applications of BiInt to type theory [3]. The duality between implication and exclusion also makes it interesting to study BiInt purely from a proof-theoretic point of view, since cut-elimination in BiInt is non-trivial. That is, the only cut-free calculi for BiInt either use extended sequent mechanisms such as labels [12], variables [6] or nested sequents [8], or display calculi that rely on residuation [5].

In this paper we follow up on our previous work on nested sequent calculi for BiInt [8], as well as our more recent work on deep inference for tense logics [7]. Nested sequents are structures that can be seen as trees of traditional sequents, and have been studied, among others, by Kashima [11] and Brünnler [2] in the context of classical modal and tense logics. Nested sequent calculi allow either "shallow" or "deep" inference: in shallow inference calculi inference rules are applied at the top level only, and residuation rules are used to re-orient the trees to bring the required structures to the top-level. In deep inference calculi, inference rules can be applied at any level, and propagation rules move formulae around the trees. We are not aware of any nested sequent calculi for intuitionistic logics that use deep inference, although there has been work on deep inference in the calculus of structures for intuitionistic logic [14]; see Section 6 for details.

In [7], we showed that deep inference in nested sequents for tense logic can simulate the residuation rules of a shallow inference nested sequent calculus. Since BiInt is the intuitionistic analog of tense logic, a natural question is whether deep inference can be applied to BiInt. We address this question here, and show that indeed we can simulate residuation using deep inference for BiInt.

More precisely, we show that residuation, an operation on the trees encoded in nested sequents, can be simulated by propagation rules, which only move formulae between the nodes rather than change the shape of the trees. Due to the intuitionistic nature of the logic, some differences from [7] arise. Namely, we introduce the concept of polarity, and some rules are only applicable in positive or negative sub-structures, rather than general sub-structures as in tense logic.

The rest of the paper is organised as follows. First, in Section 2, we give the syntax of BiInt as well as our nested sequent structures. We then present two nested sequent calculi for BiInt in Section 3: we recall the shallow inference calculus **LBiInt** [8] and give a new deep inference calculus **DBiInt**. In Section 4, we show that provability in **DBiInt** is equivalent to provability in **LBiInt**, which is the central result of our paper. The non-trivial part is showing that the residuation rules of **LBiInt** can be simulated by the propagation rules and deep inference of **DBiInt**. In Section 5, we give a simple restriction of **DBiInt** that allows terminating backward proof search. In Section 6, we describe related work and outline future work. The Appendix contains more detailed proofs.

2 Nested Sequents

The **formulae** of BiInt are built from a set of atoms *Atoms* according to the following grammar, where $p \in \mathit{Atoms}$:

$$A := p \mid \top \mid \bot \mid A \to A \mid A \mathrel{-\!\!<} A \mid A \land A \mid A \lor A.$$

A structure is defined by the following grammar, where A is a BiInt formula:

$$X := \emptyset \mid A \mid (X, X) \mid X \triangleright X.$$

The structural connective "," (comma) is associative and commutative and $\emptyset$ is its unit. We always consider structures modulo these equivalences. To reduce parentheses, we assume that "," binds tighter than "$\triangleright$". Thus, we write $X, Y \triangleright Z$ to mean $(X, Y) \triangleright Z$. If X and Y are structures, then $X \triangleright Y$ is a **nested deep sequent**, and $X \Rightarrow Y$ is a **nested shallow sequent**.

A **context** is a deep sequent with a single hole, and is written $\Sigma[]$. We write $\Sigma[X]$ to denote the sequent that results from filling the hole in $\Sigma[]$ with X. For example, if $\Sigma_1[] = Z_1 \rhd [], Z_2$ then $\Sigma_1[X_1 \rhd Y_1] = Z_1 \rhd (X_1 \rhd Y_1), Z_2$.

A hole in a context can have either negative or positive **polarity**. If $X, [] \rhd Y$ is a substructure of Σ, then $\Sigma[]$ is a negative context and we write $\Sigma^-[]$. If $X \rhd [], Y$ is a substructure of Σ, then $\Sigma[]$ is a positive context and we write $\Sigma^+[]$. For example, $\Sigma^-[] = X_1 \rhd ([] \rhd Y_1)$ is a negative context and $\Sigma^+[] = (X_1 \rhd []) \rhd Y_1$ is a positive context.

We define the **immediate super-structure** of a context as: $\overbrace{\Sigma[]} = X \rhd Y$ such that $X \rhd Y$ is a sub-structure of Σ and $X = [], X'$ for some structure X' or $Y = [], Y'$ for some structure Y'. We define the **top-level** formulae of a structure as: $\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$. For example, if $\Sigma[] = A, B \rhd C, (D, (E \rhd F) \rhd [])$, then $\overbrace{\Sigma[G]} = (D, (E \rhd F) \rhd G)$, and $\{D, (E \rhd F)\} = \{D\}$.

While deep inference allows us to "zoom-in" to any sub-structure deep inside the nested sequent, the concept of an immediate super-structure acts the opposite way in that it allows us to "zoom-out" from a context to its immediate surrounding nested structure. This will be useful when we restrict our rules for terminating proof-search, allowing us to impose local checks on the rules.

3 Nested Sequent Calculi

We now present the two nested sequent calculi that we will use in the rest of the paper: the shallow inference calculus **LBiInt** from our previous work [8] and a new deep inference calculus **DBiInt**. We repeat the rules of **LBiInt** here to make the present paper self-contained, see [8] for the full details.

Fig. 1 gives the rules of the cut-free fragment of **LBiInt** (**LBiInt** has cut-elimination). Central to this calculus is the idea that inference rules can only be applied to formulae at the top level of nested sequents, and the structural rules s_L, s_R, $\rhd_L$ and $\rhd_R$ are used to bring the required sub-structures to the top level. These rules, very similar to residuation postulates in display logic, are essential for the cut-elimination proof of **LBiInt**, however, they contain too much non-determinism for effective proof search.

Note that we have changed the notation slightly from [8] for an expository purpose: we are using $\rhd$ as the only structural connective, while the original **LBiInt** had $<$ in negative sub-structures and $>$ in positive sub-structures. Also, the $\rhd_L$ and $\rhd_R$ rules contained an implicit weakening which we have removed here. We also use $\Rightarrow$ for the sequent turnstile, reserving $\vdash$ for denoting provability. That is, we write $\vdash_{\mathbf{LBiInt}} \Pi : X \Rightarrow Y$ to mean that there exists an **LBiInt**-derivation Π of the sequent $X \Rightarrow Y$.

Fig. 2 gives the rules of our new deep inference calculus **DBiInt**. Here the inference rules can be applied at any level of the nested sequent, indicated by the use of contexts. Notably, there are no residuation rules; indeed the main goal of our paper is to show that the residuation rules of **LBiInt** can be simulated by

Identity and logical constants:

$$\frac{}{X, A \Rightarrow A, Y}\ id \qquad \frac{}{X, \bot \Rightarrow Y}\ \bot_L \qquad \frac{}{X \Rightarrow \top, Y}\ \top_R$$

Structural rules:

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

$$\frac{(X_1 \triangleright Y_1), X_2 \Rightarrow Y_2}{X_1, X_2 \Rightarrow Y_1, Y_2}\ s_L \qquad \frac{X_1 \Rightarrow Y_1, (X_2 \triangleright Y_2)}{X_1, X_2 \Rightarrow Y_1, Y_2}\ s_R$$

$$\frac{X_2 \Rightarrow Y_2, Y_1}{(X_2 \triangleright Y_2) \Rightarrow Y_1}\ \triangleright_L \qquad \frac{X_1, X_2 \Rightarrow Y_2}{X_1 \Rightarrow (X_2 \triangleright Y_2)}\ \triangleright_R$$

Logical rules:

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

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

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

$$\frac{A \Rightarrow B, Y}{X, A -\!\!< B \Rightarrow Y}\ -\!\!<_L \qquad \frac{X \Rightarrow A, Y \qquad X, B \Rightarrow Y}{X \Rightarrow A -\!\!< B, Y}\ -\!\!<_R$$

Fig. 1. LBiInt: a shallow inference system for BiInt

deep inference and propagation rules in **DBiInt**. We write $\vdash_{\mathbf{DBiInt}} \Pi : X \triangleright Y$ to mean that there exists a **DBiInt**-derivation Π of the sequent $X \triangleright Y$.

We write $|\Pi|$ for the height of a derivation, i.e., the number of sequents on the longest branch, where Π is either an **LBiInt**-derivation or a **DBiInt**-derivation.

3.1 Examples

We give two examples to illustrate the difference between shallow inference in **LBiInt** and deep inference in **DBiInt**.

Example 1. The following is a derivation of Uustalu's formula [12] in **LBiInt**:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \Rightarrow q, p}\ id \qquad \dfrac{}{p, q \Rightarrow q}\ id}{p \Rightarrow q, p -\!\!< q}\ -\!\!<_R}{p \triangleright q \Rightarrow p -\!\!< q}\ \triangleright_L}{p \triangleright q, r \Rightarrow p -\!\!< q}\ w_L \qquad \dfrac{}{(p \triangleright q), r \Rightarrow r}\ id}{(p \triangleright q), r \Rightarrow (p -\!\!< q) \wedge r}\ \wedge_R}{p \triangleright q \Rightarrow r \to ((p -\!\!< q) \wedge r)}\ \to_R}{p \Rightarrow q, r \to ((p -\!\!< q) \wedge r)}\ s_L$$

Identity and logical constants:

$$\dfrac{}{\Sigma[X, A \triangleright A, Y]}\;id \qquad \dfrac{}{\Sigma^{-}[\bot]}\;\bot_L \qquad \dfrac{}{\Sigma^{+}[\top]}\;\top_R$$

Propagation rules:

$$\dfrac{\Sigma^{-}[\{X\}, (X \triangleright Y)]}{\Sigma^{-}[X \triangleright Y]}\;\triangleright_{L1} \qquad \dfrac{\Sigma^{+}[(X \triangleright Y), \{Y\}]}{\Sigma^{+}[X \triangleright Y]}\;\triangleright_{R1}$$

$$\dfrac{\Sigma[X \triangleright (W, (\{X\}, Y \triangleright Z))]}{\Sigma[X \triangleright (W, (Y \triangleright Z))]}\;\triangleright_{L2} \qquad \dfrac{\Sigma[((X \triangleright Y, \{Z\}), W) \triangleright Z]}{\Sigma[((X \triangleright Y), W) \triangleright Z]}\;\triangleright_{R2}$$

Logical rules:

$$\dfrac{\Sigma^{-}[A \vee B, A] \qquad \Sigma^{-}[A \vee B, B]}{\Sigma^{-}[A \vee B]}\;\vee_L \qquad \dfrac{\Sigma^{+}[A \vee B, A, B]}{\Sigma^{+}[A \vee B]}\;\vee_R$$

$$\dfrac{\Sigma^{-}[A \wedge B, A, B]}{\Sigma^{-}[A \wedge B]}\;\wedge_L \qquad \dfrac{\Sigma^{+}[A \wedge B, A] \qquad \Sigma^{+}[A \wedge B, B]}{\Sigma^{+}[A \wedge B]}\;\wedge_R$$

$$\dfrac{\Sigma^{-}[A \mathbin{-\!\!<} B, (A \triangleright B)]}{\Sigma^{-}[A \mathbin{-\!\!<} B]}\;\mathbin{-\!\!<}_L \qquad \dfrac{\Sigma^{+}[A \to B, (A \triangleright B)]}{\Sigma^{+}[A \to B]}\;\to_R$$

$$\dfrac{\Sigma[X, A \to B \triangleright A, Y] \qquad \Sigma[X, A \to B, B \triangleright Y]}{\Sigma[X, A \to B \triangleright Y]}\;\to_L$$

$$\dfrac{\Sigma[X \triangleright Y, A \mathbin{-\!\!<} B, A] \qquad \Sigma[X, B \triangleright Y, A \mathbin{-\!\!<} B]}{\Sigma[X \triangleright Y, A \mathbin{-\!\!<} B]}\;\mathbin{-\!\!<}_R$$

Fig. 2. DBiInt: a deep inference system for BiInt

This example uses the rules $\triangleright_L$ and s_L to bring the required sub-structures to the top-level to apply the inference rules.

Example 2. The following is a derivation of Uustalu's formula in **DBiInt** where we abbreviate $A = r \to ((p \mathbin{-\!\!<} q) \wedge r)$, $B = (p \mathbin{-\!\!<} q) \wedge r$ and $X = p, r \triangleright B, p \mathbin{-\!\!<} q$ to save space. For readability, we draw a box around the structure that the inference rule is applied to, unless it is the top-level structure:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \triangleright q, A, X, B, p \mathbin{-\!\!<} q, p}\;id \qquad \dfrac{}{p, q \triangleright q, A, X, B, p \mathbin{-\!\!<} q}\;id}{p \triangleright q, A, X, B, p \mathbin{-\!\!<} q}\;\mathbin{-\!\!<}_R}{p \triangleright q, A, \boxed{(p, r \triangleright B, p \mathbin{-\!\!<} q)}}\;\triangleright_{R1} \qquad \dfrac{}{p \triangleright q, A, \boxed{(p, r \triangleright B, r)}}\;id}{\dfrac{p \triangleright q, A, \boxed{(p, r \triangleright (p \mathbin{-\!\!<} q) \wedge r)}}{p \triangleright q, A, (r \triangleright (p \mathbin{-\!\!<} q) \wedge r)}\;\triangleright_{L2}}\;\wedge_R}{p \triangleright q, r \to ((p \mathbin{-\!\!<} q) \wedge r)}\;\to_R$$

This example uses deep inference to apply the inference rules at any level. The formula propagation rules $\triangleright_{R1}$ and $\triangleright_{L2}$ ensure that the required formulae are propagated to the appropriate sub-structure.

4 Soundness and Completeness of DBiInt

4.1 Soundness of DBiInt

We first show that the propagation rules of **DBiInt** can be derived in **LBiInt** using residuation. This is not a surprising result, since the residuation rules in display logics are used exactly for the purpose of displaying and un-displaying sub-sequents so that inference rules can be applied to them.

Theorem 1 (Soundness). *For any structures X and Y, if $\vdash_{\mathbf{DBiInt}} \varPi : X \triangleright Y$ then $\vdash_{\mathbf{LBiInt}} \varPi' : X \Rightarrow Y$.*

Proof. By induction on $|\varPi|$. We show one interesting case. The given **DBiInt**-derivation is on the left, and we obtain the **LBiInt**-derivation on the right, where $\varPi_1'$ is obtained by the induction hypothesis (IH).

$$\dfrac{\begin{matrix}\varPi_1\\ X \triangleright (Y_1 \triangleright Y_2), \{Y_2\}\end{matrix}}{X \triangleright (Y_1 \triangleright Y_2)}\ \triangleright_{R1} \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\varPi_1'\\ X \Rightarrow (Y_1 \triangleright Y_2), \{Y_2\}\end{matrix}}{X, Y_1 \Rightarrow Y_2, \{Y_2\}}\ s_R}{X, Y_1 \Rightarrow Y_2}\ c_R}{X \Rightarrow (Y_1 \triangleright Y_2)}\ \triangleright_R$$

4.2 Completeness of DBiInt

Our aim is to show that **DBiInt** is complete w.r.t. **LBiInt**. But first we state some basic lemmas, which can all be proved using simple induction on $|\varPi|$.

Lemma 1 (Admissibility of general weakening). *For any structures X and Y: if $\vdash_{\mathbf{DBiInt}} \varPi : \varSigma[X]$ then $\vdash_{\mathbf{DBiInt}} \varPi' : \varSigma[X, Y]$ such that $|\varPi'| \leq |\varPi|$.*

Lemma 2 (Invertibility). *All* **DBiInt** *rules are invertible: if the conclusion is derivable, so are all the premises.*

Lemma 3 (Admissibility of formula contraction). *For any structure X and formula A: if $\vdash_{\mathbf{DBiInt}} \varPi : \varSigma[X, A, A]$ then $\vdash_{\mathbf{DBiInt}} \varPi' : \varSigma[X, A]$.*

Corollary 1. *For any structure X, if $\vdash_{\mathbf{DBiInt}} \varPi : \varSigma[X, \{X\}]$ then $\vdash_{\mathbf{DBiInt}} \varPi' : \varSigma[X]$.*

We now show that the residuation rules of **LBiInt** are admissible in **DBiInt**; that is, they can be simulated by the propagation rules of **DBiInt**. Lemmas 4 to 7 are proved by induction on $|\varPi|$. We show some interesting cases, where $\varPi$ ends with a propagation rule.

Lemma 4 (Admissibility of s_L). *If $\vdash_{\mathbf{DBiInt}} \varPi : (X \triangleright Y), Z \triangleright W$ then $\vdash_{\mathbf{DBiInt}} \varPi' : X, Z \triangleright Y, W$.*

Proof. Suppose Π ends as below left. Then we obtain a derivation Π_1' of $X, \{X\}, Z \triangleright Y, W$ from the IH, and a derivation Π_1'' of $X, Z \triangleright Y, W$ from Corollary 1 and Π_1'. Then the derivation on the right gives the required:

$$\dfrac{\begin{matrix}\Pi_1\\ \{X\}, (X \triangleright Y), Z \triangleright W\end{matrix}}{(X \triangleright Y), Z \triangleright W}\,\triangleright_{L1} \qquad \leadsto \qquad \begin{matrix}\Pi_1''\\ X, Z \triangleright Y, W\end{matrix}$$

Lemma 5 (Admissibility of s_R). *If* $\vdash_{\mathbf{DBiInt}} \Pi : X \triangleright Y, (Z \triangleright W)$ *then* $\vdash_{\mathbf{DBiInt}} \Pi' : X, Z \triangleright Y, W$.

Lemma 6 (Admissibility of $\triangleright_L$). *If* $\vdash_{\mathbf{DBiInt}} \Pi : X \triangleright Y, Z$ *then* $\vdash_{\mathbf{DBiInt}} \Pi' : (X \triangleright Y) \triangleright Z$.

Lemma 7 (Admissibility of $\triangleright_R$). *If* $\vdash_{\mathbf{DBiInt}} \Pi : X, Y \triangleright Z$ *then* $\vdash_{\mathbf{DBiInt}} \Pi' : X \triangleright (Y \triangleright Z)$.

Proof. Suppose Π ends as below left. Then we obtain a derivation Π_1' of $(X_1 \triangleright X_2, \{Z\}) \triangleright (Y \triangleright Z)$ from the IH, and a derivation Π_1'' of $(X_1 \triangleright X_2, \{Z\}) \triangleright ((Y \triangleright Z), \{Z\})$ from Lemma 1 and Π_1'. Then the derivation on the right gives the required:

$$\dfrac{\begin{matrix}\Pi_1\\ (X_1 \triangleright X_2, \{Z\}), Y \triangleright Z\end{matrix}}{(X_1 \triangleright X_2), Y \triangleright Z}\,\triangleright_{R2} \qquad \leadsto \qquad \dfrac{\dfrac{\begin{matrix}\Pi_1''\\ (X_1 \triangleright X_2, \{Z\}) \triangleright ((Y \triangleright Z), \{Z\})\end{matrix}}{(X_1 \triangleright X_2) \triangleright ((Y \triangleright Z), \{Z\})}\,\triangleright_{R2}}{(X_1 \triangleright X_2) \triangleright (Y \triangleright Z)}\,\triangleright_{R1}$$

Lemma 8 (Admissibility of general contraction). *For any structures X and Y: if* $\vdash_{\mathbf{DBiInt}} \Pi : \Sigma[X, Y, Y]$ *then* $\vdash_{\mathbf{DBiInt}} \Pi' : \Sigma[X, Y]$.

Proof. By induction on the size of Y. The interesting case is when $Y = (Y_1 \triangleright Y_2)$. This can be reduced to contractions on Y_1 and Y_2, which are admissible by the IH; $Y_1 \triangleright Y_2$ can then be reconstructed using Lemmas 4 to 7.

Theorem 2 (Completeness). *For any structures X and Y, if* $\vdash_{\mathbf{LBiInt}} \Pi : X \Rightarrow Y$ *then* $\vdash_{\mathbf{DBiInt}} \Pi' : X \triangleright Y$.

Proof. By induction on $|\Pi|$. We illustrate one case where Π ends in a logical rule application and one where Π ends in a structural rule application. The other interesting cases use Lemmas 4 to 7.

- Suppose Π is as below left. Then we first obtain **DBiInt**-derivations Π_1' and Π_2' of $X \triangleright A, Y$ and $X, B \triangleright Y$ respectively from the IH. Second, we obtain Π_1'' and Π_2'' by Lemma 1 from Π_1' and Π_2'. Finally, the required **DBiInt**-derivation is as below right:

$$\dfrac{\begin{matrix}\Pi_1\\ X \Rightarrow A, Y\end{matrix} \qquad \begin{matrix}\Pi_2\\ X, B \Rightarrow Y\end{matrix}}{X, A \to B \Rightarrow Y}\,\to_L \quad \leadsto \quad \dfrac{\begin{matrix}\Pi_1''\\ X, A \to B \triangleright A, Y\end{matrix} \qquad \begin{matrix}\Pi_2''\\ X, A \to B, B \triangleright Y\end{matrix}}{X, A \to B \triangleright Y}\,\to_L$$

- Suppose Π is as below left. Then we first obtain a **DBiInt**-derivation Π_1' of $X, Y \rhd Z$ by the IH. Second, we obtain a **DBiInt**-derivation Π_1'' of $X \rhd (Y \rhd Z)$ from Π_1' by Lemma 7. Then the required **DBiInt**-derivation is as below right:

$$\dfrac{\begin{array}{c}\Pi_1\\ X, Y \Rightarrow Z\end{array}}{X \Rightarrow (Y \rhd Z)}\,\rhd_R \qquad \rightsquigarrow \qquad \begin{array}{c}\Pi_1''\\ X \rhd (Y \rhd Z)\end{array}$$

Theorem 3. *For any structures X and Y, $\vdash_{\mathbf{LBiInt}} \Pi : X \Rightarrow Y$ if and only if $\vdash_{\mathbf{DBiInt}} \Pi' : X \rhd Y$.*

Proof. By Theorems 1 and 2.

5 Proof Search

Naive proof search in **DBiInt** does not terminate. Consider the following proof attempt fragment, where $X = (A \to B) \to C, (D \to E) \to F$ and we only show the left premise of each $\to_L$ rule instance:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ X \rhd G, A \to B, (X, A \rhd B, D \to E, (X, A, D \rhd E, A \to B, (A \rhd B)))\end{array}}{X \rhd G, A \to B, (X, A \rhd B, D \to E, \boxed{(X, A, D \rhd E, A \to B)})}\,\to_R}{X \rhd G, A \to B, \boxed{(X, A \rhd B, D \to E, (X, A, D \rhd E))}}\,\to_L}{X \rhd G, A \to B, \boxed{(X, A \rhd B, D \to E, (D \rhd E))}}\,\rhd_{L2}}{X \rhd G, A \to B, \boxed{(X, A \rhd B, D \to E)}}\,\to_R}{X \rhd G, A \to B, \boxed{(X, A \rhd B)}}\,\to_L}{X \rhd G, A \to B, (A \rhd B)}\,\rhd_{L2}}{X \rhd G, A \to B}\,\to_R}{(A \to B) \to C, (D \to E) \to F \rhd G}\,\to_L$$

There is an interaction between the $\to_R$, $\rhd_{L2}$ and $\to_L$ rules that causes non-termination, even for the intuitionistic fragment of the logic. This well-known problem occurs in traditional sequent calculi as well, and it is caused by the implicit contraction in the $\to_L$ rule. For intuitionistic logic, this problem has been addressed by contraction-free calculi [4] and history-based loop-checks [10]. However, these methods are less suitable for BiInt where the interaction between $\to$ and $\prec$ formulae needs to be considered. Here we address termination using a saturation process and two derived rules that speed up proof search. The approach is similar to our previous work [8], but here we apply it to deep inference and contexts instead of top-level sequents only.

Let $\prec_{L1}$ and $\to_{R1}$ denote two rules derived as below, where a dashed inference line means the conclusion is derived from the premise using Lemma 1:

$$\begin{array}{c}
\Sigma^-[A, A{-\!\!<}B] \\ \hdashline
\Sigma^-[A{-\!\!<}B, A, (A \triangleright B)] \\ \hline
\Sigma^-[A{-\!\!<}B, (A \triangleright B)] \\ \hline
\Sigma^-[A{-\!\!<}B]
\end{array}
\begin{array}{l}
\text{Lemma 1} \\ \triangleright_{L1} \\ {-\!\!<}_L
\end{array}
\quad \rightsquigarrow \quad
\dfrac{\Sigma^-[A, A{-\!\!<}B]}{\Sigma^-[A{-\!\!<}B]}\ {-\!\!<}_{L1}$$

$$\begin{array}{c}
\Sigma^+[A \to B, B] \\ \hdashline
\Sigma^+[A \to B, (A \triangleright B), B] \\ \hline
\Sigma^+[A \to B, (A \triangleright B)] \\ \hline
\Sigma^+[A \to B]
\end{array}
\begin{array}{l}
\text{Lemma 1} \\ \triangleright_{R1} \\ \to_R
\end{array}
\quad \rightsquigarrow \quad
\dfrac{\Sigma^+[A \to B, B]}{\Sigma^+[A \to B]}\ \to_{R1}$$

Let $\Sigma[Z]$ be any sequent. Then let $X \triangleright Y = \overbrace{\Sigma[Z]}$. We say that $\Sigma[Z]$ is **saturated** iff all the following conditions are met:

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 $B \in \{Y\}$
9. If $A{-\!\!<}B \in \{X\}$ then $A \in \{X\}$

Let X and Y be two structures. We say that a formula $A \to B$ is **realised** by $X \triangleright Y$ iff there exists a structure $Z \triangleright W \in Y$ such that $A \in Z$ and $B \in W$. We say that a formula $C{-\!\!<}D$ is **realised** by $X \triangleright Y$ iff there exists a structure $Z \triangleright W \in X$ such that $C \in Z$ and $D \in W$. We define the super-set relation on sequents as follows: $X_1 \triangleright Y_1 \supset X_0 \triangleright Y_0$ iff $\{X_1\} \supset \{X_0\}$ or $\{Y_1\} \supset \{Y_0\}$. Then the following simple modifications of **DBiInt** ensure termination using only local checks:

Definition 1. *Let* $\mathbf{DBiInt_1}$ *be the system obtained from* **DBiInt** *with the following changes:*

1. *Add the derived rules* ${-\!\!<}_{L1}$ *and* $\to_{R1}$.
2. *Replace rules* ${-\!\!<}_L$, $\to_R$ *by the following:*

$$\dfrac{\Sigma^-[A{-\!\!<}B, (A \triangleright B)]}{\Sigma^-[A{-\!\!<}B]}\ {-\!\!<}_L$$

where $\Sigma^-[A{-\!\!<}B]$ *is saturated and* $A{-\!\!<}B$ *is not realised by* $\overbrace{\Sigma^-[A{-\!\!<}B]}$

$$\dfrac{\Sigma^+[A \to B, (A \triangleright B)]}{\Sigma^+[A \to B]}\ \to_R$$

where $\Sigma^+[A \to B]$ *is saturated and* $A \to B$ *is not realised by* $\overbrace{\Sigma^+[A \to B]}$

3. Replace rules $\triangleright_{L2}$ and $\triangleright_{R2}$ by the following:

$$\frac{\Sigma[X \triangleright (W, (\{X\}, Y \triangleright Z))]}{\Sigma[X \triangleright (W, (Y \triangleright Z))]} \triangleright_{L2} \quad \textit{where } \{X\} \supset \{Y\}$$

$$\frac{\Sigma[((X \triangleright Y, \{Z\}), W) \triangleright Z]}{\Sigma[((X \triangleright Y), W) \triangleright Z]} \triangleright_{R2} \quad \textit{where } \{Z\} \supset \{Y\}$$

4. Replace rules $\to_L$, $-\!\!\!<_R$, $\triangleright_{L1}$, $\triangleright_{R1}$, $\wedge_L$, $\wedge_R$, $\vee_L$, $\vee_R$ with the following restricted versions:
   (a) Let γ_0 be the conclusion of the rule let γ_1 (and γ_2) be the premises. The rule is applicable only if: $\overbrace{\gamma_1} \supset \overbrace{\gamma_0}$ and $\overbrace{\gamma_2} \supset \overbrace{\gamma_0}$.

Theorem 4. *For any structures X and Y, $\vdash_{\mathbf{DBiInt}} \Pi : X \triangleright Y$ if and only if $\vdash_{\mathbf{DBiInt_1}} \Pi' : X \triangleright Y$.*

Theorem 5. *For any X and Y, backward proof search in $\mathbf{DBiInt_1}$ for $X \triangleright Y$ terminates.*

6 Related Work, Future Work and Conclusion

Deep inference: Deep inference in the calculus of structures was pioneered by Guglielmi [9]. In his work, inference rules can be applied deep inside formulae, not just deep inside nested sequent structures as in our case. This method has also been applied to intuitionistic logic [14]. The works of Kashima [11] and Brünnler [2] are closer to ours since their deep inference rules are applied to nested structures (Brünnler calls them deep sequents). However, both [11] and [2] only cover classical modal and tense logics, while we have extended the notion of deep inference to bi-intuitionistic logic using polarised contexts.

Taming display logic: Areces and Bernardi [1] appear to be the first to have noticed the connection between deep inference and residuation in display logic in the context of categorial grammar. However, they do not give an explicit proof of this correspondence as we have done here for our calculi.

Extensions and restrictions: Since BiInt is a conservative extension of intuitionistic logic, our calculi are also sound and complete for the intuitionistic fragment of BiInt: we simply need to ignore all rules for $-\!\!\!<$. We are also interested in extending our technique to similar logics such as Lambek-Grishin logic. Since many of our proofs use associativity and commutativity, it is not obvious that our technique will be immediately applicable to substructural logics.

Our contributions: The main contribution of our paper is showing that deep inference in nested sequent calculi for bi-intuitionistic logic can mimic residuation in display-like calculi. Thus our work is another step towards addressing the broader problem of proof search in display logic. Secondly, our calculus **DBiInt** and its restriction $\mathbf{DBiInt_1}$ are interesting calculi for proof search in BiInt in their own right. We leave the details of an efficient implementation of $\mathbf{DBiInt_1}$ for future work.

Acknowledgements: We would like to thank Rajeev Goré, Alwen Tiu and the anonymous reviewers for their comments on an earlier version of this paper.

References

1. C. Areces and R. Bernardi. Analyzing the core of categorial grammar. *Journal of Logic, Language, and Information*, 13(2):121–137, 2004.
2. 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.
3. T. Crolard. A formulae-as-types interpretation of Subtractive Logic. *Journal of Logic and Computation*, 14(4):529–570, August 2004.
4. R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, September 1992.
5. R. Goré. Substructural logics on display. *LJIGPL*, 6(3):451–504, 1998.
6. R. Goré and L. Postniece. Combining derivations and refutations for cut-free completeness in bi-intuitionistic logic. *Journal of Logic and Computation.* To appear, Advance Access: http://logcom.oxfordjournals.org/cgi/content/abstract/exn067.
7. R. Goré, L. Postniece, and A. Tiu. Taming displayed tense logics using nested sequents with deep inference. To appear in Proceedings of TABLEAUX 2009.
8. R. Goré, L. Postniece, and A. Tiu. Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents. In *Advances in Modal Logic 7*, pages 43–66. College Publications, 2008.
9. A. Guglielmi. A system of interaction and structure. *ACM Trans. Comput. Log.*, 8(1), 2007.
10. A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Proceedings of Tableaux*, volume 1071 of *LNAI*, pages 210–225, 1996.
11. R Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53:119–135, 1994.
12. L. Pinto and T. Uustalu. Proof search and counter-model construction for bi-intuitionistic propositional logic with labelled sequents. To appear in Proceedings of TABLEAUX 2009.
13. C. Rauszer. An algebraic and Kripke-style approach to a certain extension of intuitionistic logic. *Dissertationes Mathematicae*, 168, 1980.
14. A. Tiu. A local system for intuitionistic logic. In *Proceedings of LPAR, vol 4246, LNCS*, pages 242–256, 2006.

A Proofs

Proof of Theorem 1:

Proof. By induction on $|\Pi|$. We show the interesting cases. In each case the given **DBiInt**-derivation is on the left, and we obtain the **LBiInt**-derivation on the right, where Π_1' is obtained by the IH.

$$\dfrac{\overset{\Pi_1}{\{X_1\}, (X_1 \triangleright X_2) \triangleright Y}}{(X_1 \triangleright X_2) \triangleright Y}\,{\triangleright_{L1}} \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\overset{\Pi_1'}{\{X_1\}, (X_1 \triangleright X_2) \Rightarrow Y}}{\{X_1\}, X_1 \Rightarrow X_2, Y}\,s_L}{X_1 \Rightarrow X_2, Y}\,c_L}{(X_1 \triangleright X_2) \Rightarrow Y}\,{\triangleright_L}$$

$$\dfrac{\begin{matrix}\Pi_1\\ X \rhd (W, (\{X\}, Y_1 \rhd Y_2))\end{matrix}}{X \rhd (W, (Y_1 \rhd Y_2))}\,\rhd_{L2} \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\Pi'_1\\ X \Rightarrow W, (\{X\}, Y_1 \rhd Y_2)\end{matrix}}{X \rhd W \Rightarrow \{X\}, Y_1 \rhd Y_2}\,\rhd_L}{(X \rhd W), \{X\}, Y_1 \Rightarrow Y_2}\,s_R}{(X \rhd W), \{X\} \Rightarrow Y_1 \rhd Y_2}\,\rhd_R}{X \rhd W \Rightarrow \{X\} \rhd (Y_1 \rhd Y_2)}\,\rhd_R}{X \Rightarrow W, (\{X\} \rhd (Y_1 \rhd Y_2))}\,s_L}{X, \{X\} \Rightarrow W, (Y_1 \rhd Y_2)}\,s_R}{X \Rightarrow W, (Y_1 \rhd Y_2)}\,c_L$$

$$\dfrac{\begin{matrix}\Pi_1\\ ((X \rhd Y, \{Z\}), W) \rhd Z\end{matrix}}{((X \rhd Y), W) \rhd Z}\,\rhd_{R2} \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\Pi'_1\\ (X \rhd Y, \{Z\}), W \Rightarrow Z\end{matrix}}{X \rhd Y, \{Z\} \Rightarrow W \rhd Z}\,\rhd_R}{X \Rightarrow Y, \{Z\}, (W \rhd Z)}\,s_L}{X \rhd Y \Rightarrow \{Z\}, (W \rhd Z)}\,\rhd_L}{(X \rhd Y) \rhd \{Z\} \Rightarrow W \rhd Z}\,\rhd_L}{(X \rhd Y) \rhd \{Z\}, W \Rightarrow Z}\,s_R}{(X \rhd Y), W \Rightarrow \{Z\}, Z}\,s_L}{(X \rhd Y), W \Rightarrow Z}\,c_R$$

Each of the following proofs is by induction on $|\Pi|$. Π'_1 is obtained from Π_1 using the IH. A dashed inference line labeled W means that the conclusion is obtained from the premise using Lemma 1. A dashed inference line labeled C means that the conclusion is obtained from the premise using Corollary 1. **Proof of Lemma 4:**

Proof.

$$\dfrac{\begin{matrix}\Pi_1\\ \Sigma[(X \rhd (\{X\}, Y_1 \rhd Y_2)), Z \rhd W]\end{matrix}}{\Sigma[(X \rhd (Y_1 \rhd Y_2)), Z \rhd W]}\,\rhd_{L2} \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[X, Z \rhd (\{X\}, Y_1 \rhd Y_2), W]\\ \hdashline \Sigma[X, Z \rhd (\{X, Z\}, Y_1 \rhd Y_2), W]\end{array}\,W}{\Sigma[X, Z \rhd (Y_1 \rhd Y_2), W]}\,\rhd_{L2}$$

$$\dfrac{\begin{matrix}\Pi_1\\ \Sigma[(X \rhd Y), Z \rhd (W_1 \rhd W_2), \{W_2\}\end{matrix}}{\Sigma[(X \rhd Y), Z \rhd (W_1 \rhd W_2)}\,\rhd_{R1} \quad\rightsquigarrow\quad \dfrac{\begin{matrix}\Pi'_1\\ \Sigma[X, Z \rhd Y, (W_1 \rhd W_2), \{W_2\}]\end{matrix}}{\Sigma[X, Z \rhd Y, (W_1 \rhd W_2)]}\,\rhd_{R1}$$

$$\dfrac{\begin{matrix}\Pi_1\\ \Sigma[(X \rhd Y, \{W\}), Z \rhd W]\end{matrix}}{\Sigma[(X \rhd Y), Z \rhd W]}\,\rhd_{R2} \quad\rightsquigarrow\quad \begin{array}{c}\Pi'_1\\ \Sigma[X, Z \rhd Y, \{W\}, W]\\ \hdashline \Sigma[X, Z \rhd Y, W]\end{array}\,C$$

Proof of Lemma 5:

Proof.

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[\{X_1\}, (X_1 \triangleright X_2) \triangleright Y, (Z \triangleright W)] & \\ \hline \Sigma[(X_1 \triangleright X_2) \triangleright Y, (Z \triangleright W)] & \triangleright_{L1} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[\{X_1\}, (X_1 \triangleright X_2), Z \triangleright Y, W] & \\ \hline \Sigma[(X_1 \triangleright X_2), Z \triangleright Y, W] & \triangleright_{L1} \end{array}$$

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[X \triangleright Y, (\{Z_1\}, (Z_1 \triangleright Z_2) \triangleright W)] & \\ \hline \Sigma[X \triangleright Y, (Z_1 \triangleright Z_2 \triangleright W)] & \triangleright_{L1} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[X, \{Z_1\}, (Z_1 \triangleright Z_2) \triangleright Y, W] & \\ \hline \Sigma[X, (Z_1 \triangleright Z_2) \triangleright Y, W] & \triangleright_{L1} \end{array}$$

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[X \triangleright Y, (\{X\}, Z \triangleright W)] & \\ \hline \Sigma[X \triangleright Y, (Z \triangleright W)] & \triangleright_{L2} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[X, \{X\}, Z \triangleright Y, W] & \\ \hdashline \Sigma[X, Z \triangleright Y, W] & C \end{array}$$

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[X \triangleright Y, (Z \triangleright W), \{W\}] & \\ \hline \Sigma[X \triangleright Y, (Z \triangleright W)] & \triangleright_{R1} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[X, Z \triangleright Y, W, \{W\}] & \\ \hdashline \Sigma[X, Z \triangleright Y, W] & C \end{array}$$

$$\begin{array}{cl} \Pi'_1 & \\ \Sigma[X \triangleright Y, ((Z_1 \triangleright Z_2, \{W\}) \triangleright W)] & \\ \hline \Sigma[X \triangleright Y, ((Z_1 \triangleright Z_2) \triangleright W)] & \triangleright_{R2} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[X, (Z_1 \triangleright Z_2, \{W\}) \triangleright Y, W] & \\ \hline \Sigma[X, (Z_1 \triangleright Z_2) \triangleright Y, W] & \triangleright_{R2} \end{array}$$

Proof of Lemma 6:

Proof.

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[\{X_1\}, (X_1 \triangleright X_2) \triangleright Y, Z] & \\ \hline \Sigma[(X_1 \triangleright X_2) \triangleright Y, Z] & \triangleright_{L1} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[(\{X_1\}, (X_1 \triangleright X_2) \triangleright Y) \triangleright Z] & \\ \hline \Sigma[((X_1 \triangleright X_2) \triangleright Y) \triangleright Z] & \triangleright_{L1} \end{array}$$

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[X \triangleright (\{X\}, Y_1 \triangleright Y_2), Z] & \\ \hline \Sigma[X \triangleright (Y_1 \triangleright Y_2), Z] & \triangleright_{L2} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[(X \triangleright (\{X\}, Y_1 \triangleright Y_2)) \triangleright Z] & \\ \hline \Sigma[(X \triangleright (Y_1 \triangleright Y_2)) \triangleright Z] & \triangleright_{L2} \end{array}$$

$$\begin{array}{cl} \Pi_1 & \\ \Sigma[X \triangleright Y, (\{X\}, Z_1 \triangleright Z_2)] & \\ \hline \Sigma[X \triangleright Y, (Z_1 \triangleright Z_2)] & \triangleright_{L2} \end{array} \quad \rightsquigarrow \quad \begin{array}{cl} \Pi'_1 & \\ \Sigma[(X \triangleright Y) \triangleright (\{X\}, Z_1 \triangleright Z_2)] & \\ \hdashline \Sigma[\{X\}, (X \triangleright Y) \triangleright (\{X\}, Z_1 \triangleright Z_2)] & W \\ \hline \Sigma[\{X\}, (X \triangleright Y) \triangleright (Z_1 \triangleright Z_2)] & \triangleright_{L2} \\ \hline \Sigma[(X \triangleright Y) \triangleright (Z_1 \triangleright Z_2)] & \triangleright_{L1} \end{array}$$

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[X \triangleright (Y_1 \triangleright Y_2), \{Y_2\}, Z]\end{array}}{\Sigma[X \triangleright (Y_1 \triangleright Y_2), Z]}\triangleright_{R1} \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[(X \triangleright (Y_1 \triangleright Y_2), \{Y_2\}) \triangleright Z]\end{array}}{\Sigma[(X \triangleright (Y_1 \triangleright Y_2)) \triangleright Z]}\triangleright_{R1}$$

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[X \triangleright Y, (Z_1 \triangleright Z_2), \{Z_2\}]\end{array}}{\Sigma[X \triangleright Y, (Z_1 \triangleright Z_2)]}\triangleright_{R1} \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[(X \triangleright Y) \triangleright (Z_1 \triangleright Z_2), \{Z_2\}]\end{array}}{\Sigma[(X \triangleright Y) \triangleright (Z_1 \triangleright Z_2)]}\triangleright_{R1}$$

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[(X_1 \triangleright X_2, \{Y, Z\}) \triangleright Y, Z]\end{array}}{\Sigma[(X_1 \triangleright X_2) \triangleright Y, Z]}\triangleright_{R2} \quad\rightsquigarrow\quad \dfrac{\dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[((X_1 \triangleright X_2, \{Y, Z\}) \triangleright Y) \triangleright Z]\\ \hdashline \Sigma[((X_1 \triangleright X_2, \{Y, Z\}) \triangleright Y, \{Z\}) \triangleright Z]\end{array}\ W}{\Sigma[((X_1 \triangleright X_2) \triangleright Y, \{Z\}) \triangleright Z]}\triangleright_{R2}}{\Sigma[((X_1 \triangleright X_2) \triangleright Y) \triangleright Z]}\triangleright_{R2}$$

Proof of Lemma 7:

Proof.

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[\{X_1\}, (X_1 \triangleright X_2), Y \triangleright Z]\end{array}}{\Sigma[(X_1 \triangleright X_2), Y \triangleright Z]}\triangleright_{L1} \quad\rightsquigarrow\quad \dfrac{\dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[(X_1 \triangleright X_2) \triangleright (\{X_1\}, Y \triangleright Z)]\\ \hdashline \Sigma[(\{X_1\}, (X_1 \triangleright X_2)) \triangleright (\{X_1\}, Y \triangleright Z)]\end{array}\ W}{\Sigma[(\{X_1\}, (X_1 \triangleright X_2)) \triangleright (Y \triangleright Z)]}\triangleright_{L2}}{\Sigma[(X_1 \triangleright X_2) \triangleright (Y \triangleright Z)]}\triangleright_{L1}$$

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[X, \{Y_1\}, (Y_1 \triangleright Y_2) \triangleright Z]\end{array}}{\Sigma[X, (Y_1 \triangleright Y_2) \triangleright Z]}\triangleright_{L1} \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[X \triangleright (\{Y_1\}, (Y_1 \triangleright Y_2) \triangleright Z)]\end{array}}{\Sigma[X \triangleright ((Y_1 \triangleright Y_2) \triangleright Z)]}\triangleright_{L1}$$

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[X, Y \triangleright (\{X, Y\}, Z_1 \triangleright Z_2)]\end{array}}{\Sigma[X, Y \triangleright (Z_1 \triangleright Z_2)]}\triangleright_{L2} \quad\rightsquigarrow\quad \dfrac{\dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[X \triangleright (Y \triangleright (\{X, Y\}, Z_1 \triangleright Z_2))]\\ \hdashline \Sigma[X \triangleright (\{X\}, Y \triangleright (\{X, Y\}, Z_1 \triangleright Z_2))]\end{array}\ W}{\Sigma[X \triangleright (\{X\}, Y \triangleright (Z_1 \triangleright Z_2))]}\triangleright_{L2}}{\Sigma[X \triangleright (Y \triangleright (Z_1 \triangleright Z_2))]}\triangleright_{L2}$$

$$\dfrac{\begin{array}{c}\Pi_1\\ \Sigma[X, Y \triangleright (Z_1 \triangleright Z_2), \{Z_2\}]\end{array}}{\Sigma[X, Y \triangleright (Z_1 \triangleright Z_2)]}\triangleright_{R1} \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi'_1\\ \Sigma[X \triangleright (Y \triangleright ((Z_1 \triangleright Z_2), \{Z_2\}))]\end{array}}{\Sigma[X \triangleright (Y \triangleright (Z_1 \triangleright Z_2))]}\triangleright_{R1}$$

Proof of Lemma 8:

Proof. By induction on the size of Y. The interesting case is when $Y = (Y_1 \rhd Y_2)$. We show how this can be reduced to contractions on Y_1 and Y_2, which are admissible by the IH. A dashed inference line means that the conclusion is obtained from the premise using the respective Lemma or the IH. Suppose we have Y in a negative context, the other case is symmetric:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Sigma[(Y_1 \rhd Y_2), (Y_1 \rhd Y_2) \rhd Z]}{\Sigma[Y_1, (Y_1 \rhd Y_2) \rhd Y_2, Z]}\text{ Lemma 4}}{\Sigma[Y_1, Y_1 \rhd Y_2, Y_2, Z]}\text{ Lemma 4}}{\Sigma[Y_1 \rhd Y_2, Y_2, Z]}\text{ IH}}{\Sigma[Y_1 \rhd Y_2, Z]}\text{ IH}}{\Sigma[(Y_1 \rhd Y_2) \rhd Z]}\text{ Lemma 6}$$

Proof of Theorem 4:

Proof. For the left-to-right direction, use induction on $|\Pi|$. The interesting cases are when $|\Pi|$ ends with rule instance that does not meet one of the restrictions 2 to 4 imposed by Definition 1.

- Suppose restriction 2 of the rule $\to_R$ is not met, so that $\Sigma^+[A \to B]$ is not saturated. Then we use Lemma 2 and the IH to permute the offending rule instance upwards.
- Suppose restriction 2 of the rule $\to_R$ is not met, so that $A \to B$ is in fact realised by $\overbrace{\Sigma^+[A \to B]}$. Then Π is as below:

$$\dfrac{\begin{matrix}\Pi_1 \\ \Sigma^+[A \to B, (A \rhd B), (X, A \rhd B, Y)]\end{matrix}}{\Sigma^+[A \to B, (X, A \rhd B, Y)]} \to_R$$

  Then by the IH, there exists a $\mathbf{DBiInt_1}$-derivation Π_2 of $\Sigma^+[A \to B, (A \rhd B), (X, A \rhd B, Y)]$. By Lemma 1, there exists a $\mathbf{DBiInt_1}$-derivation Π_3 of $\Sigma^+[A \to B, (X, A \rhd B, Y), (X, A \rhd B, Y)]$. Finally, by Lemma 8, there exists a $\mathbf{DBiInt_1}$-derivation Π_1'' of $\Sigma^+[A \to B, (X, A \rhd B, Y)]$.

The right-to-left direction is obvious, since every rule of $\mathbf{DBiInt_1}$ is a rule of **DBiInt**, or can be derived in **DBiInt**.

Proof sketch of Theorem 5:

We can define a translation from $\mathbf{DBiInt_1}$ sequents to trees, similar to the one in [7]. We then show that the depth of the trees is bounded (using restriction 2 of Definition 1) and that the size of the nodes is bounded (using restrictions 3 and 4 of Definition 1).