Combining Derivations and Refutations for Cut-free Completeness in Bi-Intuitionistic Logic

Rajeev Goré
Rajeev.Gore@anu.edu.au

Linda Postniece
Linda.Postniece@anu.edu.au

Computer Sciences Laboratory
The Australian National University
Canberra ACT 0200, Australia

June 19, 2008

Abstract

Bi-intuitionistic logic is the union of intuitionistic and dual intuitionistic logic, and was introduced by Rauszer as a Hilbert calculus with algebraic and Kripke semantics. But her subsequent "cut-free" sequent calculus has recently been shown to fail cut-elimination. We present a new cut-free sequent calculus for bi-intuitionistic logic, and prove it sound and complete with respect to its Kripke semantics. Ensuring completeness is complicated by the interaction between intuitionistic implication and dual intuitionistic exclusion, similarly to future and past modalities in tense logic. Our calculus handles this interaction using derivations and refutations as first class citizens. We employ extended sequents which pass information from premises to conclusions using variables instantiated at the leaves of refutations, and rules which compose certain refutations and derivations to form derivations. Automated deduction using terminating backward search is also possible, although this is not our main purpose.

1 Introduction

Bi-intuitionistic logic, also known as subtractive logic and Heyting-Brouwer logic, is the union of intuitionistic logic and dual intuitionistic logic, and it is a conservative extension of both [26, 27]. While the proof theory of intuitionistic logic and dual intuitionistic logic separately has been studied extensively and there are many cut-free sequent calculi for intuitionistic logic (*e.g.* [12, 9, 8]) and dual intuitionistic logic (*e.g.* [31, 6]), the case for bi-intuitionistic logic is less satisfactory. Although Rauszer presented a sequent calculus for bi-intuitionistic logic and "proved" it cut-free [26], Uustalu [32] has recently shown that it fails cut-elimination. Rauszer's sequent calculus is incomplete without cut because it does not handle the interaction between intuitionistic implication $\to$ and dual intuitionistic exclusion $-\!\!<$. We explain this interaction in more detail in Section 2.

Our previous cut-free sequent calculus [3] for bi-intuitionistic logic had an operational reading, and handled the interaction using extended sequents and non-standard rules with inter-premise communication. The operational nature of our rules made our soundness proof non-traditional since it relied on the proof search strategy. Here, we give a purely syntactic (non-operational and cut-free) sequent calculus for bi-intuitionistic logic which combines derivations and refutations as first-class citizens. In particular, both soundness and completeness are proved in a purely top-down manner.

Before describing our sequent calculus, we set the scene by reviewing the notions of traditional derivation calculi, theorems, refutation calculi, non-theorems, proof/refutation search and counter-models, and explaining why a combined calculus makes sense.

Derivation calculi are used to reason about a syntactic derivability relation $\vdash$. For example, Gentzen's `LJ` [12] is a sequent calculus for propositional intuitionistic logic, where a judgement $\vdash \Gamma \Rightarrow \varphi$ means the sequent $\Gamma \Rightarrow \varphi$ is derivable: that is, the formula φ is syntactically derivable from the multiset of formulae Γ in intuitionistic logic. Increasingly, sequent calculi are used to decide whether $\Gamma \Rightarrow \varphi$ is derivable by applying the rules backwards, so it has become important to study such calculi from this "**proof-search**" perspective. Indeed, a "contraction-free" variant of `LJ`, called `LJT` [9] can be used for proof-search in intuitionistic logic. But note that a single non-derivation is not really a first-class citizen in this setting.

Refutation calculi are syntactic formalisms for reasoning about a syntactic refutability relation $\dashv$ (say). They show syntactically that a formula is a non-theorem and were introduced to modern logic by Łukasiewicz [23], although the idea originated from Aristotle. For example, Goranko has given refutation calculi for some modal logics [14] where the judgement $\dashv \varphi$ means that φ is refutable, i.e., a non-theorem of the logic. The notion of "backward refutation-search" asks whether φ is refutable under the assumptions in Γ, and some refutation calculi have been designed with this aim. For example, using $\hat{\Gamma}/\check{\Delta}$ as a conjunction/disjunction of all the members of Γ/Δ, Pinto and Dyckhoff use "sequents" of the form $\Gamma \not\Rightarrow \Delta$ to give a refutation "sequent" calculus `CRIP` for intuitionistic logic [25] where the judgement $\vdash_{CRIP} \Gamma \not\Rightarrow \Delta$ means that the formula $\hat{\Gamma} \to \check{\Delta}$ is a non-theorem. Importantly, these calculi produce refutations that are first-class objects (trees).

As usual, we can relate syntactic derivability (in `LJT`) to semantics if the calculus is sound and complete: thus $\vdash \Gamma \Rightarrow \varphi$ (in `LJT`) iff $\hat{\Gamma} \to \varphi$ is **valid** (in intuitionistic logic). Such a correspondence is vital in many applications: we pinpoint why φ is not derivable from Γ by constructing a **counter-model** showing that $\hat{\Gamma} \to \varphi$ is **falsifiable**. But since derivation calculi do not construct counter-models directly, the counter-model is constructed using meta-level reasoning to "stitch" together many non-derivations of the sequent $\Gamma \Rightarrow \varphi$.

Dually, we can relate syntactic refutability (in `CRIP`) to semantics if the calculus is sound and complete: thus $\vdash \Gamma \not\Rightarrow \varphi$ (in `CRIP`) iff the formula $\hat{\Gamma} \to \varphi$ is **falsifiable** (in intuitionistic logic). Indeed, specially designed refutation calculi, such as `CRIP`, allow us to reason about refutability **and** obtain a counter-model since a single refutation corresponds directly to a counter-model. Of course, they are not immediately suitable for demonstrating validity.

Although derivation calculi and refutation calculi are usually studied as distinct calculi, there are desirable **meta-level** relationships between derivability and refutability (for the same logic). For example, for any input Γ and φ, either there is a derivation of $\Gamma \Rightarrow \varphi$ in `LJT`, or a refutation of $\Gamma \not\Rightarrow \varphi$ in CRIP [25]. It therefore makes sense to ask what would happen if we were to combine derivation calculi with refutation calculi in one single setting. For example, the *modus tollens* rule used in some refutation calculi **combines** a **derivation** of $\varphi \to \psi$ and a **refutation** of ψ to obtain a **refutation** of φ. As Goranko suggests, we could also combine **derivations** and **refutations** to produce **derivations**. Indeed, he predicts that such **combined deductive systems** *"have a greater potential efficiency than the orthodox ones, since they can employ on a syntactic level self-reference to some of their meta-features, which are beyond the expressive abilities of the traditional systems"* [14].

To retain the link with semantics as well as the potential for backward (proof/refutation) search, the combined calculus must be such that derivations/refutations preserve validity / counter-models downwards while providing a decision procedure if our logic is decidable. There is a subtlety here, for Larchey-Wendling [22] has already combined proof search and explicit counter-model construction to obtain an efficient decision procedure for an extension of intuitionistic logic called Gödel-Dummett logic. But Larchey-Wendling constructs a counter-model merely as a tool used at certain times during proof search. Thus his calculus does not contain derivations and refutations as first-class citizens.

We now give an overview of our sequent calculus. A sequent is an expression $\mathcal{S}\ \Gamma \triangleright \Delta\ \mathcal{P}$ or $\mathcal{S}\ \Gamma \triangleleft\ \Delta\ \mathcal{P}$ where Γ/Δ are traditional sets of formulae, $\mathcal{S}/\mathcal{P}$ are sets of sets of formulae, and the turnstiles $\triangleright$ and $\triangleleft$ indicate whether we have a derivation or a refutation. The extra components $\mathcal{S}/\mathcal{P}$ are variables, which are a mechanism to pass information from premises to conclusions [28], similarly to attributes in attribute grammars. In our case, the variables are sets of sets of formulae containing subformulae discovered at the leaves of refutation trees. Our rules transmit these essential formulae down towards the root of refutations, and use these formulae to obtain a derivation from a refutation. In fact, we obtain a demand-driven cut as viewed from a backward (proof/refutation) search perspective: rather than having to guess cut formulae at each sequent, we perform cut-free backward search as usual, and use the contents of variables when we find a refutation: Section 4 gives details.

We then relate our generalised syntactic judgement $\bowtie$ (either $\triangleright$ or $\triangleleft$) to a generalised semantic judgement (either validity or falsifiability) via a combined soundness and completeness proof. We show that the rules preserve the generalised semantic judgement downwards: validity/falsifiability is preserved downwards in derivation/refutation trees. Thus derivable sequents are valid and refutable sequents are falsifiable. Additionally, we show that in certain special cases we can combine a refutation with a derivation to obtain a derivation. Finally, we give a terminating procedure which decides whether a generalised sequent $\Gamma \bowtie \Delta$ is derivable or refutable using a number of side conditions that the rules must obey. Thus completeness follows directly from our ability to derive or refute every input sequent, rather than indirectly from the failure of a systematic proof search procedure.

In Section 2, we motivate our work by showing why the interaction between $\to$ and $-\!\!<$ in bi-intuitionistic logic poses difficulties for traditional cut-free sequent calculi. In Section 3, we define the syntax and semantics of bi-intuitionistic logic. In Section 4, we introduce our sequent calculus. We prove its soundness and completeness in Section 5. In Section 6, we describe a decision procedure for bi-intuitionistic logic and analyse its computational complexity. In Section 7, we describe related work.

2 Motivation and Example

We now give a high-level overview of the difficulties posed by interaction formulae in bi-intuitionistic logic, and how our sequent calculus solves them. As we have not formally defined the syntax and semantics of bi-intuitionistic logic yet, this section is intended to give a general overview and motivation for the problem before delving into the technicalities.

The sequent $p \Rightarrow q, r \to ((p \mathbin{-\!\!<} q) \wedge r)$ [32] has a derivation using cut, as shown below:

Example 1. *Below is a derivation of Uustalu's sequent using Rauszer's calculus with cut, where the rule for* $\mathbin{-\!\!<}$ *on the right is symmetric to the intuitionistic rule for* $\to$ *on the left:*

$$\dfrac{\dfrac{\dfrac{}{\mathbf{q} \Rightarrow \mathbf{q}}\,(\mathrm{Id}) \qquad \dfrac{}{p \Rightarrow p}\,(\mathrm{Id})}{p \Rightarrow \mathbf{q}, \mathbf{p} \mathbin{-\!\!<} \mathbf{q}}\,(\mathbin{-\!\!<}_R) \qquad \dfrac{\dfrac{\dfrac{}{p \mathbin{-\!\!<} q, r \Rightarrow p \mathbin{-\!\!<} q}\,(\mathrm{Id}) \qquad \dfrac{}{p \mathbin{-\!\!<} q, r \Rightarrow r}\,(\mathrm{Id})}{p \mathbin{-\!\!<} q, r \Rightarrow (p \mathbin{-\!\!<} q) \wedge r}\,(\wedge_R)}{p \mathbin{-\!\!<} q \Rightarrow r \to ((p \mathbin{-\!\!<} q) \wedge r)}\,(\to_R)}{p \Rightarrow \mathbf{q}, r \to ((\mathbf{p} \mathbin{-\!\!<} \mathbf{q}) \wedge r)}\,\mathit{(cut)}$$

The end sequent contains three complementary pairs, a positive and negative occurrence of p, q and r respectively: note that p occurs positively and q negatively in the $p \mathbin{-\!\!<} q$ in the end sequent. All three pairs occur in the axioms and are essential for the derivation.

Now consider a backward attempt to derive this sequent without cut. The only non-structural rule that could give the conclusion is the $(\to_R)$ rule. But for intuitionistic soundness, the conclusion of $(\to_R)$ in Rauszer's calculus needs to be restricted to a singleton succedent. Thus we must weaken away the q to obtain the premise $p \Rightarrow r \to ((p \mathbin{-\!\!<} q) \wedge r)$. But we have lost the essential occurrence of q. The only alternative is to weaken away p or $r \to ((p \mathbin{-\!\!<} q) \wedge r)$. But neither of the resulting premises is derivable.

Crolard's dependency tracking [5] is one way to relax the intuitionistic restriction of "singletons on the right" to retain the essential q, so that cases like the above example remain cut-free derivable whilst retaining soundness. The idea is to record the dependencies between antecedents and succedents of the axioms, and use these dependencies to impose side conditions on the rules. Thus dependency tracking is not immediately suitable for backward proof search since the side conditions need to be known when the rules are applied backwards, before the axioms, and hence the dependencies, are computed.

A cut-free derivation of Uustalu's sequent in our calculus ends as shown below:

$$\dfrac{\begin{matrix}\cdots \\ \{\cdots\}\ p, r \ \lhd\ (\mathbf{p} \mathbin{-\!\!<} \mathbf{q}) \wedge r\ \{\{\mathbf{p} \mathbin{-\!\!<} \mathbf{q}\}\}\end{matrix} \qquad \begin{matrix}\cdots \\ p\ \rhd\ \mathbf{q}, r \to ((\mathbf{p} \mathbin{-\!\!<} \mathbf{q}) \wedge r), \mathbf{p} \mathbin{-\!\!<} \mathbf{q}\end{matrix}}{p\ \rhd\ \mathbf{q}, r \to ((p \mathbin{-\!\!<} q) \wedge r)}\,(\to_{R2})$$

In addition to the traditional antecedent and succedent, our sequents contain two non-traditional components, which we call variables. In the derivation sketch above, $\{\{p \mathbin{-\!\!<} q\}\}$ is one of the variables of the top left sequent; we have not shown the variables of the other sequents. The "$\lhd$" turnstile in the left premise denotes a refutation, the "$\rhd$" turnstile in the right premise denotes a derivation, and the $(\to_{R2})$ rule allows us to compose the refutation with a derivation. Notice that the variable $\{\{p \mathbin{-\!\!<} q\}\}$ in the left premise contains the crucial subformula $p \mathbin{-\!\!<} q$, which is also present in the formula part of the right premise.

$w \Vdash \varphi \vee \psi$	iff	$w \Vdash \varphi$ or $w \Vdash \psi$	$w \dashv\!\mid \varphi \vee \psi$	iff	$w \dashv\!\mid \varphi$ & $w \dashv\!\mid \psi$
$w \Vdash \varphi \wedge \psi$	iff	$w \Vdash \varphi$ & $w \Vdash \psi$	$w \dashv\!\mid \varphi \wedge \psi$	iff	$w \dashv\!\mid \varphi$ or $w \dashv\!\mid \psi$
$w \Vdash \neg\varphi$	iff	$\forall u \geq w \ . \ u \dashv\!\mid \varphi$	$w \dashv\!\mid \neg\varphi$	iff	$\exists u \geq w \ . \ u \Vdash \varphi$
$w \Vdash \varphi \to \psi$	iff	$\forall u \geq w \ . \ u \dashv\!\mid \varphi$ or $u \Vdash \psi$	$w \dashv\!\mid \varphi \to \psi$	iff	$\exists u \geq w \ . \ u \Vdash \varphi$ & $u \dashv\!\mid \psi$
$w \Vdash \sim\varphi$	iff	$\exists u \leq w \ . \ u \dashv\!\mid \varphi$	$w \dashv\!\mid \sim\varphi$	iff	$\forall u \leq w \ . \ u \Vdash \varphi$
$w \Vdash \varphi \prec \psi$	iff	$\exists u \leq w \ . \ u \Vdash \varphi$ & $u \dashv\!\mid \psi$	$w \dashv\!\mid \varphi \prec \psi$	iff	$\forall u \leq w \ . \ u \dashv\!\mid \varphi$ or $u \Vdash \psi$

Figure 1: Forcing and rejecting of formulae

3 Syntax and Semantics of `BiInt`

The **formulae** *Fml* of `BiInt` are built from a set of atoms *Atoms*, the constants $\top$ and $\bot$, using the binary connectives $\wedge$, $\vee$, $\to$, $\prec$ and unary connectives $\neg$ and $\sim$. The connectives $\neg$ and $\to$ are those of intuitionistic logic, the connectives $\sim$ and $\prec$ are those of dual intuitionistic logic and the connectives $\vee$ and $\wedge$ are from both. The **length** of a `BiInt` formula is the number of symbols it contains.

We use classical first-order logic for reasoning about `BiInt` at the meta-level, using "$\Rightarrow$" for classical implication, "or" for classical disjunction, and " & " for classical conjunction. A `BiInt` **frame** is a pair $\langle \mathcal{W}, \mathcal{R} \rangle$, where $\mathcal{W}$ is a non-empty set of worlds and $\mathcal{R}$ is a reflexive and transitive binary accessibility relation. A `BiInt` **model** $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ is a `BiInt` frame $\langle \mathcal{W}, \mathcal{R} \rangle$ together with a truth valuation function ϑ: $Atoms \cup \{\top, \bot\} \to 2^{\mathcal{W}}$ such that $\vartheta(\top) = \mathcal{W}$, $\vartheta(\bot) = \emptyset$ and which obeys **persistence**:

$$\forall u, w \in \mathcal{W}. \forall p \in Atoms. w\mathcal{R}u \ \& \ w \in \vartheta(p) \Rightarrow u \in \vartheta(p).$$

Definition 2. *Given two valuations $\vartheta_1 = Atoms_1 \cup \{\top, \bot\} \to 2^{\mathcal{W}_1}$ and $\vartheta_2 = Atoms_2 \cup \{\top, \bot\} \to 2^{\mathcal{W}_2}$ with $\mathcal{W}_1 \cap \mathcal{W}_2 = \emptyset$, we define the disjoint union of ϑ_1 and ϑ_2 as a set of pairs:*

$$\vartheta_1 \cup \vartheta_2 := \{(p, S) \mid p \in Atoms_1 \cup Atoms_2 \cup \{\top, \bot\} \text{ and } S = \{w \mid w \in \mathcal{W}_1 \cup \mathcal{W}_2 \text{ and } [w \in \vartheta_1(p) \text{ or } w \in \vartheta_2(p)]\}.$$

For some statement X, we sometimes abbreviate:

$\forall u \in \mathcal{W}. w\mathcal{R}u \Rightarrow X$	to	$\forall u \geq w. X$	$\forall u \in \mathcal{W}. u\mathcal{R}w \Rightarrow X$	to	$\forall u \leq w. X$
$\exists u \in \mathcal{W}. w\mathcal{R}u \ \& \ X$	to	$\exists u \geq w. X$	$\exists u \in \mathcal{W}. u\mathcal{R}w \ \& \ X$	to	$\exists u \leq w. X.$

Definition 3. *Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, a world $w \in \mathcal{W}$ and an atom or a logical constant $\varphi \in Atoms \cup \{\top, \bot\}$, we write $w \Vdash \varphi$ (w forces φ) iff $w \in \vartheta(\varphi)$, and we write $w \dashv\!\mid \varphi$ (w rejects φ) iff $w \notin \vartheta(\varphi)$. We define forcing and rejecting of compound formulae by mutual recursion in Figure 1.*

From the semantics, it is clear that the connectives $\neg$ and $\sim$ can be derived from $\to$ and $\prec$ respectively. Therefore we restrict our attention to the connectives $\to$, $\prec$, $\wedge$, $\vee$ only.

By induction on the length of a formula φ, it follows that the persistence property also holds for formulae, and the reverse persistence property holds for formulae, that is:

Persistence: $\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle . \forall w \in \mathcal{W} . \forall \varphi \in Fml . \forall u \geq w . (w \Vdash \varphi \Rightarrow u \Vdash \varphi)$.

Reverse persistence: $\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle . \forall w \in \mathcal{W} . \forall \varphi \in Fml . \forall u \leq w . (w \dashv\vert \varphi \Rightarrow u \dashv\vert \varphi)$.

We write $\emptyset$ to mean the empty set. Given a formula φ and two sets Δ and Γ of formulae, we write Δ, Γ for $\Delta \cup \Gamma$ and we write Δ, φ for $\Delta \cup \{\varphi\}$.

Definition 4. *Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and a world $w \in \mathcal{W}$, we write:*

$$w \Vdash \Gamma \quad \textit{iff} \quad \forall \varphi \in \Gamma . w \Vdash \varphi \qquad\qquad w \dashv\vert \Delta \quad \textit{iff} \quad \forall \varphi \in \Delta . w \dashv\vert \varphi .$$

Definition 5 (Validity and falsifiability)**.** *Given sets Γ and Δ of* `BiInt` *formulae, a* `BiInt` *formula φ and a* `BiInt` *model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, we write:*

$$\begin{array}{lll} \mathcal{M} \Vdash \varphi & \textit{iff} & \forall w \in \mathcal{W} . w \Vdash \varphi \\ \mathcal{M} \Vdash \Gamma & \textit{iff} & \forall w \in \mathcal{W} . w \Vdash \Gamma \end{array} \qquad \begin{array}{lll} \mathcal{M} \dashv\vert \varphi & \textit{iff} & \exists w \in \mathcal{W} . w \dashv\vert \varphi \\ \mathcal{M} \dashv\vert \Delta & \textit{iff} & \exists w \in \mathcal{W} . w \dashv\vert \Delta . \end{array}$$

Then validity, falsifiability, satisfiability and unsatisfiability are:

$$\textit{validity} \qquad \forall \mathcal{M} . \mathcal{M} \Vdash \varphi \tag{3.1}$$

$$\textit{falsifiability} \qquad \exists \mathcal{M} . \mathcal{M} \dashv\vert \varphi \tag{3.2}$$

$$\textit{satisfiability} \qquad \exists \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle . \exists w \in \mathcal{W} . w \Vdash \varphi \tag{3.3}$$

$$\textit{unsatisfiability} \qquad \forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle . \forall w \in \mathcal{W} . w \dashv\vert \varphi . \tag{3.4}$$

We write $\vDash_{\texttt{BiInt}} \varphi$ for "φ is valid", and $\Dashv_{\texttt{BiInt}} \varphi$ for "φ is falsifiable".

4 Our Sequent Calculus

We now present a Gentzen-style sequent calculus for `BiInt`. The sequents have a non-traditional component in the form of variables that are sets of sets of formulae. When our calculus is used for backward search, the variables are instantiated at certain leaves of the search tree, and passed to lower sequents from premises to conclusion. Note that the variables are not names for Kripke worlds, so our sequents contain no semantic features.

4.1 Sequents

We introduce an extended syntax to simplify the presentation of some of our sequent rules.

Definition 6. *If φ is a* `BiInt` *formula, then φ is an extended* `BiInt` *formula. If $\mathcal{Q}$ is a set $\{\{\varphi_0^0, \cdots, \varphi_0^{n_0}\}, \cdots, \{\varphi_m^0, \cdots, \varphi_m^{n_m}\}\}$ of sets of* `BiInt` *formulae, then $\bigvee \mathcal{Q}$ and $\bigwedge \mathcal{Q}$ are extended* `BiInt` *formulae with intended semantics*

$$\bigvee \mathcal{Q} \equiv (\varphi_0^0 \wedge \cdots \wedge \varphi_0^{n_0}) \vee \cdots \vee (\varphi_m^0 \wedge \cdots \wedge \varphi_m^{n_m}) \tag{4.1}$$

$$\bigwedge \mathcal{Q} \equiv (\varphi_0^0 \vee \cdots \vee \varphi_0^{n_0}) \wedge \cdots \wedge (\varphi_m^0 \vee \cdots \vee \varphi_m^{n_m}) . \tag{4.2}$$

The following semantics follows directly from Definition 6:

Definition 7. *Given a* `BiInt` *model* $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, *a world* $w \in \mathcal{W}$, *and two extended* `BiInt` *formulae* $\bigvee \mathcal{S}$ *and* $\bigwedge \mathcal{P}$, *we write:*

$$w \Vdash \bigvee \mathcal{S} \quad \textit{iff} \quad \exists \Sigma \in \mathcal{S}. \forall \varphi \in \Sigma. w \Vdash \varphi \qquad w \dashV \bigvee \mathcal{S} \quad \textit{iff} \quad \forall \Sigma \in \mathcal{S}. \exists \varphi \in \Sigma. w \dashV \varphi$$
$$w \Vdash \bigwedge \mathcal{P} \quad \textit{iff} \quad \forall \Pi \in \mathcal{P}. \exists \varphi \in \Pi. w \Vdash \varphi \qquad w \dashV \bigwedge \mathcal{P} \quad \textit{iff} \quad \exists \Pi \in \mathcal{P}. \forall \varphi \in \Pi. w \dashV \varphi.$$

We can now extend Definition 4 in the obvious way. That is, if Γ *and* Δ *are sets of extended* `BiInt` *formulae, and* φ *is an extended* `BiInt` *formula, then:*

$$w \Vdash \Gamma \quad \textit{iff} \quad \forall \varphi \in \Gamma. w \Vdash \varphi \qquad\qquad w \dashV \Delta \quad \textit{iff} \quad \forall \varphi \in \Delta. w \dashV \varphi$$

Definition 8 (Sequent)**.** *A sequent/antisequent is an expression of one of the forms*

$$\mathcal{S}\ \Gamma \ \triangleright\ \Delta\ \mathcal{P} \qquad\qquad \mathcal{S}\ \Gamma \ \triangleleft\ \Delta\ \mathcal{P}$$

and consists of the following components: a ***left hand side (LHS)*** Γ *which is a set of extended* `BiInt` *formulae; a* ***right hand side (RHS)*** Δ *which is a set of extended* `BiInt` *formulae; two* ***variables*** $\mathcal{S}$, $\mathcal{P}$, *each of which is a set of sets of* `BiInt` *formulae; and a* ***turnstile*** *which is either* $\triangleright$ *for traditional sequents or* $\triangleleft$ *for antisequents.*

We shall often use the following simplifications when referring to sequents:

$\mathcal{S}\ \Gamma \ \bowtie\ \Delta\ \mathcal{P}$ when we are referring to either a traditional sequent or an antisequent;

$\Gamma \triangleright \Delta$ or $\Gamma \triangleleft \Delta$ or $\Gamma \bowtie \Delta$ when the values of the variables are not important.

We now define the semantics of a sequent. In the following, τ is a translation from sequents to `BiInt` formulae, and σ is a translation from sequents to semantic judgements, and $\hat{\Gamma}/\check{\Delta}$ is a conjunction/disjunction of all the members of Γ/Δ:

$$\tau(\mathcal{S}\ \Gamma \ \bowtie\ \Delta\ \mathcal{P}) = \bigvee \mathcal{S} \wedge \hat{\Gamma} \to \check{\Delta} \vee \bigwedge \mathcal{P} \tag{4.3}$$
$$\sigma(\mathcal{S}\ \Gamma \ \triangleright\ \Delta\ \mathcal{P}) = \vDash_{\texttt{BiInt}} \tau(\mathcal{S}\ \Gamma \ \triangleright\ \Delta\ \mathcal{P}) \tag{4.4}$$
$$\sigma(\mathcal{S}\ \Gamma \ \triangleleft\ \Delta\ \mathcal{P}) = \Dashv_{\texttt{BiInt}} \tau(\mathcal{S}\ \Gamma \ \triangleleft\ \Delta\ \mathcal{P}) \tag{4.5}$$

The reading of (4.4)/(4.5) is that the formula corresponding to a sequent/antisequent is valid/falsifiable.

Definition 9. *A sequent* $\Gamma \bowtie \Delta$ *is* saturated *iff all of the following hold:*

$$\Gamma \cap \Delta = \emptyset$$
$$\Gamma \textit{ and } \Delta \textit{ contain only } \texttt{BiInt} \textit{ formulae}$$

if $\varphi \wedge \psi \in \Gamma$ *then* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$	*if* $\varphi \wedge \psi \in \Delta$ *then* $\varphi \in \Delta$ *or* $\psi \in \Delta$
if $\varphi \vee \psi \in \Gamma$ *then* $\varphi \in \Gamma$ *or* $\psi \in \Gamma$	*if* $\varphi \vee \psi \in \Delta$ *then* $\varphi \in \Delta$ *and* $\psi \in \Delta$
if $\varphi \to \psi \in \Gamma$ *then* $\varphi \in \Delta$ *or* $\psi \in \Gamma$	*if* $\varphi \prec \psi \in \Delta$ *then* $\varphi \in \Delta$ *or* $\psi \in \Gamma$
if $\varphi \prec \psi \in \Gamma$ *then* $\varphi \in \Gamma$	*if* $\varphi \to \psi \in \Delta$ *then* $\psi \in \Delta$.

Definition 10. *A sequent* $\Gamma \bowtie \Delta$ *is* strongly saturated *iff all of the following hold:*

(i) $\Gamma \bowtie \Delta$ *is saturated* *(ii) if* $\varphi \to \psi \in \Delta$ *then* $\varphi \in \Gamma$ *(iii) if* $\varphi \prec \psi \in \Gamma$ *then* $\psi \in \Delta$.

4.2 Sequent Rules and Various Calculi

We now describe the sequent rules, axioms and anti-axioms that are used to build derivation and refutation trees. Rather than summarising all the rules in one large table, we break them into groups and describe each group in turn. We start with the **axioms** and **anti-axiom**, which are the leaves of derivations and refutations respectively:

Axioms: (Id) $\emptyset\ \Gamma, \varphi \ \rhd\ \Delta, \varphi\ \emptyset$ $\quad$ $(\bot_L)$ $\emptyset\ \bot, \Gamma \ \rhd\ \Delta\ \emptyset$ $\quad$ $(\top_R)$ $\emptyset\ \Gamma \ \rhd\ \Delta, \top\ \emptyset$

Anti-axiom: (Ret) $\{\Gamma\}\ \Gamma \ \lhd\ \Delta\ \{\Delta\}$ where $\Gamma \lhd \Delta$ is strongly saturated.

The **axioms** (Id), $(\bot_L)$ and $(\top_R)$ have the traditional sequent turnstile "$\rhd$", while the **anti-axiom** (Ret) has the anti-sequent turnstile "$\lhd$". The rules will propagate these turnstiles down the trees, eventually arriving at the root, which will be a derivation if the root sequent is a traditional sequent ("$\rhd$"), or a refutation if the root sequent is an anti-sequent ("$\lhd$"). Note that the anti-axiom (Ret) instantiates the values of the $\mathcal{S}$ and $\mathcal{P}$ variables to $\{\Gamma\}$ and $\{\Delta\}$ respectively, while the axioms (Id), $(\bot_L)$ and $(\top_R)$ set the variables to empty sets $\emptyset$. The sequent rules will transmit the variables down the trees, and combine variables from multiple premises in some cases.

Using terminology from [16], the **static rules** of our sequent calculus are:

α-rules:

$$(\wedge_L)\ \frac{\mathcal{S}\ \Gamma, \varphi \wedge \psi, \varphi, \psi \ \bowtie_1\ \Delta\ \mathcal{P}}{\mathcal{S}\ \Gamma, \varphi \wedge \psi \ \bowtie_0\ \Delta\ \mathcal{P}} \qquad (\vee_R)\ \frac{\mathcal{S}\ \Gamma \ \bowtie_1\ \Delta, \varphi \vee \psi, \varphi, \psi\ \mathcal{P}}{\mathcal{S}\ \Gamma \ \bowtie_0\ \Delta, \varphi \vee \psi\ \mathcal{P}}$$

$$(-\!\!<^I_L)\ \frac{\mathcal{S}\ \Gamma, \varphi -\!\!< \psi, \varphi \ \bowtie_1\ \Delta\ \mathcal{P}}{\mathcal{S}\ \Gamma, \varphi -\!\!< \psi \ \bowtie_0\ \Delta\ \mathcal{P}} \qquad (\to^I_R)\ \frac{\mathcal{S}\ \Gamma \ \bowtie_1\ \Delta, \varphi \to \psi, \psi\ \mathcal{P}}{\mathcal{S}\ \Gamma \ \bowtie_0\ \Delta, \varphi \to \psi\ \mathcal{P}}$$

Where $\bowtie_1 = \bowtie_0 \in \{\rhd, \lhd\}$.

β-rules:

$$(\vee_L)\ \frac{\mathcal{S}_1\ \Gamma, \varphi \vee \psi, \varphi \ \bowtie_1\ \Delta\ \mathcal{P}_1 \qquad \mathcal{S}_2\ \Gamma, \varphi \vee \psi, \psi \ \bowtie_2\ \Delta\ \mathcal{P}_2}{\mathcal{S}_1 \cup \mathcal{S}_2\ \Gamma, \varphi \vee \psi \ \bowtie_0\ \Delta\ \mathcal{P}_1 \cup \mathcal{P}_2}$$

$$(\wedge_R)\ \frac{\mathcal{S}_1\ \Gamma \ \bowtie_1\ \Delta, \varphi \wedge \psi, \varphi\ \mathcal{P}_1 \qquad \mathcal{S}_2\ \Gamma \ \bowtie_2\ \Delta, \varphi \wedge \psi, \psi\ \mathcal{P}_2}{\mathcal{S}_1 \cup \mathcal{S}_2\ \Gamma \ \bowtie_0\ \Delta, \varphi \wedge \psi\ \mathcal{P}_1 \cup \mathcal{P}_2}$$

$$(\to_L)\ \frac{\mathcal{S}_1\ \Gamma, \varphi \to \psi \ \bowtie_1\ \Delta, \varphi\ \mathcal{P}_1 \qquad \mathcal{S}_2\ \Gamma, \varphi \to \psi, \psi \ \bowtie_2\ \Delta\ \mathcal{P}_2}{\mathcal{S}_1 \cup \mathcal{S}_2\ \Gamma, \varphi \to \psi \ \bowtie_0\ \Delta\ \mathcal{P}_1 \cup \mathcal{P}_2}$$

$$(-\!\!<_R)\ \frac{\mathcal{S}_1\ \Gamma, \psi \ \bowtie_1\ \Delta, \varphi -\!\!< \psi\ \mathcal{P}_1 \qquad \mathcal{S}_2\ \Gamma \ \bowtie_2\ \Delta, \varphi -\!\!< \psi, \varphi\ \mathcal{P}_2}{\mathcal{S}_1 \cup \mathcal{S}_2\ \Gamma \ \bowtie_0\ \Delta, \varphi -\!\!< \psi\ \mathcal{P}_1 \cup \mathcal{P}_2}$$

$$\text{Where } \bowtie_0 = \begin{cases} \rhd & \text{if } \bowtie_1 = \rhd \text{ and } \bowtie_2 = \rhd \\ \lhd & \text{otherwise.} \end{cases}$$

These rules use many features of Dragalin's `GHPC` [8] for intuitionistic logic (`Int`); we have added symmetric rules for the dual intuitionistic logic (`DualInt`) connective $\prec$. We chose Dragalin's **multi-succedent** calculus since the restriction to single succedents/antecedents for some sequents is one of the causes of incompleteness in Rauszer's calculus for `BiInt` [26]; see also Maehara [24] for early work on a multi-succedent calculus for `Int`. But using Dragalin's calculus and its dual does not give us `BiInt` completeness. We therefore also follow Schwendimann's approach [28] of passing relevant information from premises to conclusions using variables, which we instantiate at the refutation leaves: see (Ret) above.

We have also added the static rule ($\rightarrow^I_R$) for implication on the right (and symmetrically, ($\prec^I_L$)) originally given by Śvejdar [29]. Although Śvejdar himself does not give the semantics behind this rule, and is unable to explain the precise role it plays in his calculus, his rules are best explained by reading them from conclusion to premises, as used in backward search. Consider the ($\rightarrow^I_R$) rule when $\bowtie_0 = \bowtie_1 = \lhd$. As we shall show later, finding a refutation of the conclusion involves falsifying the formula $\varphi \rightarrow \psi$. Rather than immediately creating the successor that falsifies $\varphi \rightarrow \psi$, the ($\rightarrow^I_R$) rule first pre-emptively adds ψ to the right hand side of the sequent. The rule effectively uses the reverse persistence property: if some successor v forces φ and rejects ψ, then the current world w must reject ψ too. These rules are very useful in our termination proof and saturation strategy in Section 6.

Contrary to `GHPC` and other traditional sequent calculi, our ($\rightarrow_L$) rule and the symmetric ($\prec_R$) contain implicit contractions on formulae other than just the principal formula. That is, during backward search, they carry their principal formula and all side formulae into the premises. Our rules ($\wedge_L$), ($\wedge_R$), ($\vee_L$) and ($\vee_R$) also carry their principal formula into their premises. We chose this approach because it allows us to give a semantic interpretation to the anti-axiom (Ret). Because the static rules keep the principal formula from conclusion to premises, we can immediately deduce that a strongly saturated sequent, i.e., an instance of (Ret), has a counter-model. The proof of Lemma 17, case "$\lhd$", makes use of this property of our calculus.

Many of our rules use the generic "$\bowtie$" turnstile, and a clause that specifies whether the conclusion should have the "$\rhd$" or the "$\lhd$" turnstile. This indicates that various combinations of "$\rhd$" and "$\lhd$" are possible for the premises, and determines the turnstile of the conclusion in each case, as illustrated by the following example.

Example 11. *All of the following are possible instances of the* ($\wedge_R$) *rule:*

1. *An instance which combines two derivations into a derivation:*

$$\frac{\emptyset\ q, r\ \rhd\ q \wedge r, q\ \emptyset \qquad \emptyset\ q, r\ \rhd\ q \wedge r, r\ \emptyset}{\emptyset\ q, r\ \rhd\ q \wedge r\ \emptyset}\ (\wedge_R)$$

2. *An instance which combines a derivation and a refutation into a refutation:*

$$\frac{\emptyset\ q, r\ \rhd\ q \wedge p, q\ \emptyset \qquad \{\{q, r\}\}\ q, r\ \lhd\ q \wedge p, p\ \{\{q \wedge p, p\}\}}{\{\{q, r\}\}\ q, r\ \lhd\ q \wedge p\ \{\{q \wedge p, p\}\}}\ (\wedge_R)$$

3. An instance which combines a refutation and a derivation into a refutation:

$$\dfrac{\{\{q,r\}\}\ q,r\ \lhd\ p\wedge q,p\ \{\{p\wedge q,p\}\} \qquad \emptyset\ q,r\ \rhd\ p\wedge q,q\ \emptyset}{\{\{q,r\}\}\ q,r\ \lhd\ p\wedge q\ \{\{p\wedge q,p\}\}}\ (\wedge_R)$$

4. An instance which combines two refutations into a refutation:

$$\dfrac{\{\{t,r\}\}\ t,r\ \lhd\ q\wedge p,q\ \{\{q\wedge p,q\}\} \qquad \{\{t,r\}\}\ t,r\ \lhd\ q\wedge p,p\ \{\{q\wedge p,p\}\}}{\{\{t,r\}\}\ t,r\ \lhd\ q\wedge p\ \{\{q\wedge p,q\},\{q\wedge p,p\}\}}\ (\wedge_R)$$

As Example 11 shows, the conclusion of each of our rules **assigns the variables** based on the variables returned from the premise(s). In defining the rules, we use the indices $i,1,2$ to indicate the premise from which the variable takes its value. For rules with a single premise, the variables are simply passed down from premise to conclusion. For example, the conclusion of $(\wedge_L)$ has the same value of the variable $\mathcal{S}$ as the premise. However, for rules with multiple premises, we take a union of the sets of sets corresponding to each premise. For example, in Example 11(4) above, the $\mathcal{P}$ variable contains both $\{q\wedge p,q\}$ and $\{q\wedge p,p\}$, where the first set is from the left premise and the second set is from the right premise.

Thus the sets of sets stored in our variables **determinise** the return of formulae to lower sequents: semantically, each refutable premise corresponds to an open branch, and at this point we do not know whether it will stay open once processed in conjunction with lower sequents. Therefore, we need to temporarily keep all open branches. See also Remark 21 for a syntactic motivation for the set-of-sets concept.

The following are the **transitional rules** of our sequent calculus:

$$(-\!\!<_{L1})\ \dfrac{\mathcal{S}\ \varphi\ \rhd\ \Delta,\psi\ \mathcal{P}}{\mathcal{S}\ \Gamma,\varphi-\!\!<\psi\ \rhd\ \Delta\ \mathcal{P}} \qquad (\to_{R1})\ \dfrac{\mathcal{S}\ \Gamma,\varphi\ \rhd\ \psi\ \mathcal{P}}{\mathcal{S}\ \Gamma\ \rhd\ \Delta,\varphi\to\psi\ \mathcal{P}}$$

$$(\text{Refute})\ \dfrac{\text{Prem}_1^{\prec}\ \cdots\ \text{Prem}_m^{\prec} \qquad \text{Prem}_1^{\to}\ \cdots\ \text{Prem}_n^{\to}}{\{\Gamma'\}\ \Gamma,\varphi_1\prec\psi_1,\cdots,\varphi_m\prec\psi_m\ \lhd\ \Delta,\chi_1\to\xi_1,\cdots,\chi_n\to\xi_n\ \{\Delta'\}}$$

where

(1) $\Gamma'=\Gamma,\varphi_1\prec\psi_1,\cdots,\varphi_m\prec\psi_m$ (2) $\Delta'=\Delta,\chi_1\to\xi_1,\cdots,\chi_n\to\xi_n$

(3) $\Gamma'\lhd\Delta'$ is saturated

(4) Γ does not contain $\prec$-formulae and Δ does not contain $\to$-formuale

(5) $\forall i\in\{1,\cdots,m\}\ \forall j\in\{1,\cdots,n\}$:

(*a*) $\text{Prem}_i^{\prec}=\mathcal{S}_i^{\prec}\ \varphi_i\ \lhd\ \Delta',\psi_i\ \mathcal{P}_i^{\prec}$ (*b*) $\text{Prem}_j^{\to}=\mathcal{S}_j^{\to}\ \Gamma',\chi_j\ \lhd\ \xi_j\ \mathcal{P}_j^{\to}$

(*c*) $\exists\Sigma\in\mathcal{S}_i^{\prec}.\Sigma\subseteq\Gamma'$ (*d*) $\exists\Pi\in\mathcal{P}_j^{\to}.\Pi\subseteq\Delta'$.

The $(\to_{R1})$ rule is from Dragalin's `GHPC` [8], and the $(-\!\!<_{L1})$ is symmetric for the `DualInt` case: these rules introduce their principal $\to$-formula on the right or $-\!\!<$ -formula on the left. The (Refute) rule composes refutations of its premises to give a refutation of a sequent that may contain a number of $\to$-formulae on the right and $-\!\!<$-formulae on the left. That the premises be refutable is stipulated by side conditions (5a) and (5b), which state that all premises have the "$\lhd$" turnstile and hence are refutations. The extra side conditions (3), (4), (5c) and (5d) ensure that the conclusion of an instance of (Refute) is falsifiable (see the proof of Lemma 19) meaning that only certain refutations can be combined using this rule.

Example 12. *The following is an example instance of* (Refute)*:*

$$\frac{\{s\}\ s\ \lhd\ b, a \to b, t\ \{b, a \to b, t\} \qquad \{r, s, q \to r, s \prec t, a\}\ r, s, q \to r, s \prec t, a\ \lhd\ b\ \{b\}}{\{r, s, q \to r, s \prec t\}\ r, s, q \to r, s \prec t\ \lhd\ b, a \to b\ \{b, a \to b\}}\ (\text{Refute})$$

In this case:

- $m = 1,\ n = 1,\ \Gamma = \{r, s, q \to r\},\ \Delta = \{b\}$
- $Prem_1^{\prec} = \{s\}\ s\ \lhd\ b, a \to b, t\ \{b, a \to b, t\}$
- $Prem_1^{\to} = \{r, s, q \to r, s \prec t, a\}\ r, s, q \to r, s \prec t, a\ \lhd\ b\ \{b\}.$

The following **special logical rules** are used to derive additional transitional rules:

$$(\textstyle\bigvee_L)\ \frac{\mathcal{S}_1\ \Gamma, \Pi_1\ \rhd\ \Delta\ \mathcal{P}_1 \quad \cdots \quad \mathcal{S}_m\ \Gamma, \Pi_m\ \rhd\ \Delta\ \mathcal{P}_m}{\bigcup_1^m \mathcal{S}_i\ \Gamma, \bigvee(\{\Pi_1, \cdots, \Pi_m\})\ \rhd\ \Delta\ \bigcup_1^m \mathcal{P}_i}$$

$$(\textstyle\bigwedge_R)\ \frac{\mathcal{S}_1\ \Gamma\ \rhd\ \Sigma_1, \Delta\ \mathcal{P}_1 \quad \cdots \quad \mathcal{S}_n\ \Gamma\ \rhd\ \Sigma_n, \Delta\ \mathcal{P}_n}{\bigcup_1^n \mathcal{S}_i\ \Gamma\ \rhd\ \bigwedge(\{\Sigma_1, \cdots, \Sigma_n\}), \Delta\ \bigcup_1^n \mathcal{P}_i}$$

$$(\textstyle\bigwedge_L)\ \frac{\mathcal{S}_1\ \Gamma, \varphi_1\ \rhd\ \Delta\ \mathcal{P}_1 \quad \cdots \quad \mathcal{S}_k\ \Gamma, \varphi_k\ \rhd\ \Delta\ \mathcal{P}_k}{\bigcup_1^k \mathcal{S}_i\ \Gamma, \bigwedge(\{\{\varphi_1, \cdots, \varphi_n\}\})\ \rhd\ \Delta\ \bigcup_1^k \mathcal{P}_i} \qquad (\textstyle\bigwedge_L^{\cup})\ \frac{\mathcal{S}\ \Gamma, \bigwedge\Sigma_1, \bigwedge\Sigma_2\ \rhd\ \Delta\ \mathcal{P}}{\mathcal{S}\ \Gamma, \bigwedge(\Sigma_1 \cup \Sigma_2)\ \rhd\ \Delta\ \mathcal{P}}$$

$$(\textstyle\bigvee_R)\ \frac{\mathcal{S}_1\ \Gamma\ \rhd\ \varphi_1, \Delta\ \mathcal{P}_1 \quad \cdots \quad \mathcal{S}_k\ \Gamma\ \rhd\ \varphi_k, \Delta\ \mathcal{P}_k}{\bigcup_1^k \mathcal{S}_i\ \Gamma\ \rhd\ \bigvee(\{\{\varphi_1, \cdots, \varphi_k\}\}), \Delta\ \bigcup_1^k \mathcal{P}_i} \qquad (\textstyle\bigvee_R^{\cup})\ \frac{\mathcal{S}\ \Gamma\ \rhd\ \bigvee\Pi_1, \bigvee\Pi_2, \Delta\ \mathcal{P}}{\mathcal{S}\ \Gamma\ \rhd\ \bigvee(\Pi_1 \cup \Pi_2), \Delta\ \mathcal{P}}$$

These rules simply allow us to introduce extended `BiInt` formulae. The ($\bigvee_L$) rule allows us to introduce $\bigvee$-formulae on the left, and the symmetric ($\bigwedge_R$) rule allows us to introduce $\bigwedge$-formulae on the right. The ($\bigwedge_L$) allows us to introduce a $\bigwedge$-formula, containing a single set containing a set of formulae, on the left, and the ($\bigwedge_L^{\cup}$) rule allows us to introduce a larger $\bigwedge$-formula from two smaller ones; the ($\bigvee_R$) and ($\bigvee_R^{\cup}$) rules are symmetric.

The following **structural rules** are also used to derive additional transitional rules:

$$(\text{cut})\ \frac{\mathcal{S}\ \Gamma\ \rhd\ \Delta, \varphi\ \mathcal{P} \qquad \mathcal{S}\ \varphi, \Gamma\ \rhd\ \Delta\ \mathcal{P}}{\mathcal{S}\ \Gamma\ \rhd\ \Delta\ \mathcal{P}}$$

$$(\text{LW})\ \frac{\mathcal{S}\ \Gamma\ \rhd\ \Delta\ \mathcal{P}}{\mathcal{S}\ \varphi, \Gamma\ \rhd\ \Delta\ \mathcal{P}} \qquad (\text{RW})\ \frac{\mathcal{S}\ \Gamma\ \rhd\ \Delta\ \mathcal{P}}{\mathcal{S}\ \Gamma\ \rhd\ \Delta, \varphi\ \mathcal{P}}$$

	GBiInt0	GBiInt1	GBiInt
Axioms, anti-axioms, static and transitional rules	✓	✓	✓
Special logical rules	✓	✓	
Structural rules	✓	✓	
Derived transitional rules		✓	✓

Figure 2: Calculi

Finally, the following **derived transitional rules** are used to achieve cut-free completeness, and their derived status will be explained shortly:

$$(\prec_{L2})\ \frac{\mathcal{S}\ \varphi\ \lhd\ \Delta,\psi\ \mathcal{P} \qquad \mathcal{S}_1\ \Gamma,\varphi\prec\psi,\Sigma_1\ \bowtie_1\ \Delta\ \mathcal{P}_1\ \cdots\ \mathcal{S}_n\ \Gamma,\varphi\prec\psi,\Sigma_n\ \Delta\ \bowtie_n\ \mathcal{P}_n}{\bigcup_1^n \mathcal{S}_i\ \Gamma,\varphi\prec\psi\ \bowtie_0\ \Delta\ \bigcup_1^n \mathcal{P}_i}$$

$$\text{Where } \mathcal{S} = \{\Sigma_1,\cdots,\Sigma_n\} \text{ for } n \geq 1 \text{ and } \bowtie_0 = \begin{cases} \rhd & \text{if } \bowtie_i = \rhd \text{ for all } 1 \leq i \leq n \\ \lhd & \text{otherwise} \end{cases}$$

$$(\to_{R2})\ \frac{\mathcal{S}\ \Gamma,\varphi\ \lhd\ \psi\ \mathcal{P} \qquad \mathcal{S}_1\ \Gamma\ \bowtie_1\ \Pi_1,\Delta,\varphi\to\psi\ \mathcal{P}_1\ \cdots\ \mathcal{S}_m\ \Gamma\ \bowtie_m\ \Pi_m,\Delta,\varphi\to\psi\ \mathcal{P}_m}{\bigcup_1^m \mathcal{S}_i\ \Gamma\ \bowtie_0\ \Delta,\varphi\to\psi\ \bigcup_1^m \mathcal{P}_i}$$

$$\text{Where } \mathcal{P} = \{\Pi_1,\cdots,\Pi_m\} \text{ for } m \geq 1 \text{ and } \bowtie_0 = \begin{cases} \rhd & \text{if } \bowtie_i = \rhd \text{ for all } 1 \leq i \leq m \\ \lhd & \text{otherwise} \end{cases}$$

These rules compose a refutation of the *left-most premise* with one or more derivations/refutations of the *right premises*, where the formula-parts of the right premises contain formula sets like Σ_i and Π_i found in the variables of the left-most premise. That is, the right premise $\mathcal{S}_i\ \Gamma\ \bowtie_i\ \Pi_i,\Delta,\varphi\to\psi\ \mathcal{P}_i$ of the $(\to_{R2})$ rule contains the formula set $\Pi_i \in \mathcal{P}$, where $\mathcal{P}$ is one of the variables of the left-most premise.

We now explain how we can use the $(\to_{R2})$ and $(\prec_{L2})$ rules during backward search by giving an operational left-to-right reading for the rules. We first refute the left-most premise, which gives an instantiation of $\mathcal{S}$ and $\mathcal{P}$. In the $(\to_{R2})$ case, we then extract the variable $\mathcal{P}$, and create $m \geq 1$ right premises, where each right premise corresponds to the conclusion together with additional formulae found in one of the members of $\mathcal{P}$. We then attempt to derive/refute the right premises using backward search, and put $\bowtie_0$ equal to "$\rhd$" or "$\lhd$" depending on whether or not all the right premises are derivable.

Having introduced all the sequent rules, we now define several sub-calculi that we shall use throughout the rest of the paper: see Figure 2. `GBiInt0` is the base system, which is sound (Lemmas 16 to 19) and complete (although we do not show it), but uses the cut rule. `GBiInt1` is obtained from `GBiInt0` by adding two rules $(\to_{R2})$ and $(\prec_{L2})$, which are `GBiInt0`-derivable and hence sound (Lemma 22). `GBiInt` is obtained from `GBiInt1` by removing the special rules and structural rules and is cut-free, sound (Theorem 25) and complete (Theorem 41). `GBiInt` with additional blocking conditions is also the sequent calculus we use for backward search in Section 6. We use the generic name `GBiInt•` when we refer to any of the calculi, for example, in definitions, descriptions of rules and so on.

Note that our main calculus `GBiInt` is cut-free: the cut rule is used only for showing the soundness of our derived transitional rules ($\to_{R2}$) and ($-\!\!<_{L2}$). Intuitively, we show how variable-passing absorbs essential instances of (cut) in a demand-driven way. Proving that variables absorb *all* essential cuts would give syntactic cut-admissibility.

`GBiInt` also has the subformula property. This is obvious for the LHS- and RHS-components of the sequents. For the variables, the subformula property is of a global nature: when the variables are instantiated at instances of the (Ret) anti-axiom, they take values from the LHS- and RHS-components of this anti-axiom. When the variables are passed down towards the root of refutations, they are combined using the union operator, so no new formulae are created. Thus all formulae are subformulae of the end-sequent.

`GBiInt` is also free of all other structural rules; that is, explicit contraction and weakening is not required to achieve completeness. We could have also started with sequents as multisets instead of sets and shown that contraction is admissible, but since all our static rules contain implicit contractions, this would be a simple and redundant exercise.

Definition 13. *A* `GBiInt•` *tree is a tree of sequents where each leaf is an instance of the* `GBiInt•` *axioms or anti-axiom, and parents are obtained from children by instantiating a* `GBiInt•` *rule. The height of a* `GBiInt•` *tree is the number of sequents on the longest branch. A **derivation** is a* `GBiInt•` *tree rooted at $\mathcal{S}\ \Gamma \rhd \Delta\ \mathcal{P}$. A sequent is **derivable** if there exists a derivation for it; we write $\vdash \mathcal{S}\ \Gamma \rhd \Delta\ \mathcal{P}$. A **refutation** is a* `GBiInt•` *tree rooted at $\mathcal{S}\ \Gamma \lhd \Delta\ \mathcal{P}$. A sequent is **refutable** if there exists a refutation for it; we write $\vdash \mathcal{S}\ \Gamma \lhd \Delta\ \mathcal{P}$.*

We deliberately use $\vdash$ for derivability and refutability to emphasise their first-class status.

4.3 Example

We now revisit Uustalu's [32] example that we first saw in Section 2, and show the full derivation of this example using `GBiInt`. In all cases below, $X := (p -\!\!< q) \wedge r$.
Let (1) be the refutation below:

$$\dfrac{\dfrac{}{\{\{p,r,q\}\}\ p,r,q \lhd X,\mathbf{p}\prec\mathbf{q}\ \{\{X,\mathbf{p}\prec\mathbf{q}\}\}}\,\text{(Ret)} \qquad \dfrac{}{\emptyset\ p,r \rhd X,p\prec q,p\ \emptyset}\,\text{(Id)}}{\{\{p,r,q\}\}\ p,r \lhd X,p\prec q\ \{\{X,\mathbf{p}\prec\mathbf{q}\}\}}\,(\prec_R)$$

Let (2) be the derivation below:

$$\dfrac{\dfrac{}{\emptyset\ p,\mathbf{q} \rhd q,r\to X,X,p\prec q\ \emptyset}\,\text{(Id)} \qquad \dfrac{}{\emptyset\ p \rhd q,r\to X,X,p\prec q,\mathbf{p}\ \emptyset}\,\text{(Id)}}{\emptyset\ p \rhd q,r\to X,X,\mathbf{p}\prec\mathbf{q}\ \emptyset}\,(\prec_R)$$

Then the following is a cut-free derivation of Uustalu's [32] formula $p \to (q \vee (r \to ((p -\!\!< q) \wedge r))$, simplified to the sequent $p \bowtie q, r \to ((p -\!\!< q) \wedge r)$:

$$\dfrac{\dfrac{(1) \qquad \dfrac{}{\emptyset\ p,r \rhd X,r\ \emptyset}\,\text{(Id)}}{\{\{p,r,q\}\}\ p,r \lhd (p\prec q)\wedge r\ \{\{X,\mathbf{p}\prec\mathbf{q}\}\}}\,(\wedge_R) \qquad (2)}{\emptyset\ p \rhd q,r\to((p\prec q)\wedge r)\ \emptyset}\,(\to_{R2})$$

The top left anti-axiom in (1) is an instance of (Ret) because the sequent is strongly saturated. The variables $\mathcal{S}$ and $\mathcal{P}$ that are assigned at this (Ret) anti-axiom transmit information down to the parents and across to their siblings via the $(\to_{R2})$ rule.

The key to the derivation is the bolded $\mathbf{p} \mathbf{-\!\!<} \mathbf{q}$ formula that occurs in the variable $\mathcal{P}$ of the left-most leaf of (1) and in the RHS of the right premise (2) of $(\to_{R2})$. Note that $(\to_{R2})$ has only one right premise here, since the $\mathcal{P}$ variable contains only one set of formulae.

We can also read the above derivation as a backward search. We start with the end-sequent $p \bowtie q, r \to ((p \prec q) \wedge r)$, which we want to prove or refute. Since the only possible principal formula is a $\to$-formula on the right, we know that we need to use either $(\to_{R1})$, $(\to_{R2})$ or (Refute). In all cases, we need to consider the sequent $p, r \bowtie (p \prec q) \wedge r$. We then find a refutation of the sequent $p, r \bowtie (p \prec q) \wedge r$, obtaining $\{\{p, r, q\}\}\ p, r\ \triangleleft\ (p \prec q) \wedge r\ \{\{X, \mathbf{p} \prec \mathbf{q}\}\}$ and thus receiving back the variables $\mathcal{S} = \{\{p, r, q\}\}$ and $\mathcal{P} = \{\{X, \mathbf{p} \prec \mathbf{q}\}\}$. We then apply the $(\to_{R2})$ rule since its side conditions are met. The left premise is $\{\{p, r, q\}\}\ p, r\ \triangleleft\ (p \prec q) \wedge r\ \{\{X, \mathbf{p} \prec \mathbf{q}\}\}$, and we create a single right premise because the $\mathcal{P}$ variable contains a single member $\{X, \mathbf{p} \prec \mathbf{q}\}$. Since the right premise is derivable, so is the end-sequent, so we put $\bowtie\, = \,\triangleright$ and obtain $\emptyset\ p\ \triangleright\ q, r \to ((p \prec q) \wedge r)\ \emptyset$.

5 Soundness and Completeness

In this section, we prove the soundness and completeness of `GBiInt` with respect to the semantics of `BiInt`. We start by proving that the base rules of `GBiInt0` are sound.

5.1 Soundness of `GBiInt0`

We first observe that the variables are empty at the root of derivations.

Lemma 14. *If $\mathcal{S}\ \Gamma\ \triangleright\ \Delta\ \mathcal{P}$ is derivable then $\mathcal{S} = \emptyset$ and $\mathcal{P} = \emptyset$.*

Proof. By induction on the height of the given derivation. This is obvious for all `GBiInt•` rules which combine "$\triangleright$"-premises into a "$\triangleright$"-conclusion since the union of empty sets is $\emptyset$.

The more interesting cases are the rules $(\to_{R2})$ and $(-\!\!<_{L2})$, which combine a refutation ("$\triangleleft$") of the left-most premise with a combination of refutations ("$\triangleleft$") or derivations ("$\triangleright$") of the right premises. Here, the variables at the conclusion are the union of the variables of the right premises only. Since the condition in these rules specifies that a derivation of the conclusion is obtained only when *all* the *right* premises are derivations, again the variables at the conclusion are the union of empty sets, giving the empty set as required. ☐

We will prove soundness of the rules by showing that each rule preserves the semantic judgement σ from (4.4) and (4.5) downwards, so we start by formally defining this concept.

Definition 15. *A `GBiInt•` rule ρ with conclusion $\mathcal{S}_0\ \Gamma_0\ \bowtie\ \Delta_0\ \mathcal{P}_0$ and $n \geq 1$ premises, with i-th premise $\mathcal{S}_i\ \Gamma_i\ \bowtie\ \Delta_i\ \mathcal{P}_i$, preserves the semantic judgement σ downwards if:*

$$\forall i \in \{1, \cdots, n\}. \sigma(\mathcal{S}_i\ \Gamma_i\ \bowtie_i\ \Delta_i\ \mathcal{P}_i) \Rightarrow \sigma(\mathcal{S}_0\ \Gamma_0\ \bowtie_0\ \Delta_0\ \mathcal{P}_0).$$

Lemma 16. *The static, special,* $(\to_{R1})$ *and* $(\prec_{L1})$ *logical rules, and all structural rules preserve the semantic judgement* σ *downwards.*

Proof. Easily follows from translations 4.4 and 4.5 and the definitions of the rules. ☐

Lemma 17. *The semantic judgement* σ *holds at the leaves of* `GBiInt●` *trees. That is, the* $\triangleright$*-leaves are valid, and the* $\triangleleft$*-leaves are falsifiable.*

Proof.

$\triangleright$: A leaf $\Gamma \triangleright \Delta$ must be an instance of (Id), $(\bot_L)$ or $(\top_R)$. In all cases, the corresponding formula shown below is valid:

$$\begin{array}{lllllll} (\text{Id}) & \bigvee\emptyset \wedge \hat{\Gamma} \wedge \varphi & \to & \check{\Delta} \vee \varphi \vee \bigwedge\emptyset & = & \hat{\Gamma} \wedge \varphi & \to & \check{\Delta} \vee \varphi \\ (\bot_L) & \bigvee\emptyset \wedge \hat{\Gamma} \wedge \bot & \to & \check{\Delta} \vee \bigwedge\emptyset & = & \hat{\Gamma} \wedge \bot & \to & \check{\Delta} \\ (\top_R) & \bigvee\emptyset \wedge \hat{\Gamma} & \to & \check{\Delta} \vee \top \vee \bigwedge\emptyset & = & \hat{\Gamma} & \to & \check{\Delta} \vee \top. \end{array}$$

$\triangleleft$: A leaf $\Gamma \triangleleft \Delta$ must be an instance of (Ret). We show that the corresponding formula $\bigvee\mathcal{S} \wedge \hat{\Gamma} \to \check{\Delta} \vee \bigwedge\mathcal{P}$ is falsifiable. Since (Ret) assigns $\mathcal{S} := \{\Gamma\}$ and $\mathcal{P} := \{\Delta\}$, the corresponding formula under translation τ is

$$\bigvee\{\Gamma\} \wedge \hat{\Gamma} \to \check{\Delta} \vee \bigwedge\{\Delta\} \quad = \quad \hat{\Gamma} \wedge \hat{\Gamma} \to \check{\Delta} \vee \check{\Delta} \quad = \quad \hat{\Gamma} \to \check{\Delta}.$$

To falsify $\hat{\Gamma} \to \check{\Delta}$, we create a model with a single reflexive world w_0, and for every atom p in Γ, we let $\vartheta(p) = \{w_0\}$, and for every atom q in Δ, we let $\vartheta(q) = \emptyset$. An atom cannot be both in Γ and Δ since $\Gamma \triangleleft \Delta$ must be strongly saturated and thus $\Gamma \cap \Delta = \emptyset$.

To show that $\hat{\Gamma} \to \check{\Delta}$ is falsifiable at w_0, we need to show that $w_0 \Vdash \Gamma$ and $w_0 \dashv\!\mid \Delta$. For every atom in Γ and Δ, the valuation ensures both. For every composite formula φ, we do a simultaneous induction on its length. Since the side condition of (Ret) implies that $\Gamma \triangleleft \Delta$ is strongly saturated, we know that the required subformulae are already in Γ or Δ as appropriate, and they fall under the induction hypothesis.

Thus we know that $w_0 \Vdash \Gamma$ and $w_0 \dashv\!\mid \Delta$, therefore $\hat{\Gamma} \to \check{\Delta}$ is falsifiable.

☐

Definition 18. *Given an instance of* (Refute)*, we use* $\to$*-premises to refer to the premises* $\text{Prem}_1^{\to}, \cdots, \text{Prem}_n^{\to}$ *, and* $\prec$*-premises to refer to the premises* $\text{Prem}_1^{\prec}, \cdots, \text{Prem}_m^{\prec}$*.*

The proof of the next lemma has similarities to parts of a traditional completeness proof.

Lemma 19. *The* (Refute) *rule preserves the semantic judgement downwards.*

Proof. We assume that the semantic judgement σ holds for all the premises, and show that σ holds for the conclusion. That is, we assume that all the premises are falsifiable and show that the conclusion is falsifiable. To show that the conclusion is falsifiable, we need to show that there exists a `BiInt` model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and a world $w \in \mathcal{W}$ such that $w \Vdash \Gamma'$ and $w \dashv\!\mid \Delta'$. We construct the model as follows:

Step 1. Let $\mathcal{W} = \{w_0\}$ and $\mathcal{R}_0 = \{(w_0, w_0)\}$.

Step 2. For all atoms $p \in \Gamma$, let $\vartheta(p) = \{w_0\}$. For all atoms $q \in \Delta$, let $\vartheta(q) = \emptyset$.
That is, the valuation makes every atom in Γ true at w_0. Since side condition (3) of (Refute) ensures that the conclusion is saturated, Definition 9 implies $\Gamma \cap \Delta = \emptyset$, and hence the valuation makes every atom in Δ false at w_0. Then, since the conclusion is saturated, simultaneous induction on the length of Γ and Δ gives $w_0 \Vdash \Gamma$ and $w_0 \dashv\!\vert\, \Delta$.

Step 3. For each $\varphi_i \prec \psi_i \in \Gamma'$:

(a) Since the premise $Prem_i^{\prec} = \mathcal{S}_i^{\prec}\ \varphi_i \ \lhd\ \Delta', \psi_i\ \mathcal{P}_i^{\prec}$ is falsifiable by assumption, we know there exists a `BiInt` model $\mathcal{M}_i = \langle \mathcal{W}_i, \mathcal{R}_i, \vartheta_i \rangle$ and a world $w_i \in \mathcal{W}_i$ such that $w_i \Vdash \bigvee \mathcal{S}_i^{\prec}, \varphi_i$ and $w_i \dashv\!\vert\, \Delta', \psi_i, \bigwedge \mathcal{P}_i^{\prec}$. If necessary, we rename the worlds in $\mathcal{W}_i$ to ensure their names are disjoint from the names of worlds already in $\mathcal{W}$.

(b) Let $\mathcal{W} = \mathcal{W} \cup \mathcal{W}_i$.

(c) Let $\mathcal{R}_0 = \mathcal{R}_0 \cup \mathcal{R}_i \cup \{(w_i, w_0)\}$ thus making w_i an $\mathcal{R}_0$-predecessor of w_0.

(d) Let $\vartheta = \vartheta \cup \vartheta_i$ using Definition 2.

Step 4. For each $\chi_j \to \xi_j \in \Delta'$, perform an analogous procedure to Step 3, using $Prem_j^{\to} = \mathcal{S}_j^{\to}\ \Gamma', \chi_j \ \lhd\ \xi_j\ \mathcal{P}_j^{\to}$, except sub-step (c) becomes $\mathcal{R}_0 = \mathcal{R}_0 \cup \mathcal{R}_j \cup \{(w_0, w_j)\}$.

Step 5. Let $\mathcal{R}$ be the transitive closure of $\mathcal{R}_0$.

Step 6. We now have that $\langle \mathcal{W}, \mathcal{R} \rangle$ is a `BiInt` frame.

Step 7. To show that $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ is a `BiInt` model, we also need to show that it obeys persistence. From Steps 3 and 4 we know that w_0 has $\mathcal{R}_0$-predecessors w_i and $\mathcal{R}_0$-successors w_j. Forward persistence holds between all w_i and w_0, and between w_0 and all w_j because:

(a) Step 3 gives (i) $w_i \Vdash \bigvee \mathcal{S}_i^{\prec}, \varphi_i$. We have $\varphi_i \in \Gamma'$ because $\varphi_i \prec \psi_i \in \Gamma'$ and condition (3) of (Refute) implies $\Gamma' \lhd \Delta'$ is saturated. Therefore (ii) $w_0 \Vdash \varphi_i$. We have (iii) $w_0 \Vdash \bigvee \mathcal{S}_i^{\prec}$ because of side condition (5c) of (Refute) and the semantics of the $\bigvee$ connective: see Definition 7. From (ii) and (iii) we get $w_0 \Vdash \bigvee \mathcal{S}_i^{\prec}, \varphi_i$, and so every formula forced by the i-th $\mathcal{R}_0$-predecessor w_i is also forced by w_0.

(b) By inspection, all $\to$-premises 1 to n contain Γ'. This gives us that every formula found in Γ' and hence forced by w_0 is also forced by all $\mathcal{R}_0$-successors in Step 4.

Similarly, reverse persistence holds because of side condition (5d) of (Refute) and the fact that all $\prec$-premises 1 to m contain Δ'.

To show that persistence holds for all $\mathcal{R}$-related worlds, we use transitivity of the subset relation and the initial assumption, specifically the fact that persistence holds in all models $\mathcal{M}_i$ and $\mathcal{M}_j$ used in Steps 3 and 4.

□

5.2 Soundness of GBiInt1

The only difference between GBiInt0 and GBiInt1 is that GBiInt1 contains the extra transitional rules $(\to_{R2})$ and $(\prec_{L2})$. We now show that each of these rules is derivable in GBiInt0. In particular, instances of $(\to_{R2})$ and $(\prec_{L2})$ can be seen as absorbing certain instances of cut and weakening. Since GBiInt0 is sound, so are the extra derived rules.

The following lemma is crucial for showing the soundness of $(\to_{R2})$ and $(\prec_{L2})$ because it shows that the variables at the root of a refutation in fact contain the information required to turn the refutation into a derivation. More specifically, we will use the $\mathcal{S}$ variable to obtain a derivation from the refutation when we apply the $(\prec_{L2})$ rule, and will use the $\mathcal{P}$ variable when we apply the $(\to_{R2})$ rule. Reading GBiInt• trees top-down, we do not know which variable will be required at a lower sequent, so we keep both $\mathcal{S}$ and $\mathcal{P}$.

Lemma 20. *For all $\mathcal{S}, \Gamma, \Delta, \mathcal{P}$:*
if $\vdash \mathcal{S}\ \Gamma \triangleleft \Delta\ \mathcal{P}$, then $\vdash \emptyset \bigwedge \mathcal{P}, \Gamma \triangleright \Delta\ \emptyset$ and $\vdash \emptyset\ \Gamma \triangleright \Delta, \bigvee \mathcal{S}\ \emptyset$.

Proof. By induction on the height of the refutation of $\mathcal{S}\ \Gamma \triangleleft \Delta\ \mathcal{P}$.
Base Case: A refutation of height 1 must be an instance of (Ret):

(Ret) $\{\Gamma\}\ \Gamma \triangleleft \Delta\ \{\Delta\}$ where $\Gamma \triangleleft \Delta$ is strongly saturated

That is, $\mathcal{S} = \{\Gamma\}$ and hence $\emptyset\ \Gamma \triangleright \Delta, \bigvee \mathcal{S}\ \emptyset$ is $\emptyset\ \Gamma \triangleright \Delta, \bigvee\{\Gamma\}\ \emptyset$. Then the following is a derivation of $\emptyset\ \Gamma \triangleright \Delta, \bigvee\{\Gamma\}\ \emptyset$, where $\Gamma = \{\gamma_1, \cdots, \gamma_k\}$ for some $k \geq 1$:

$$\dfrac{\dfrac{}{\emptyset\ \Gamma \triangleright \Delta, \gamma_1\ \emptyset}\,(\text{Id}) \quad \cdots \quad \dfrac{}{\emptyset\ \Gamma \triangleright \Delta, \gamma_k\ \emptyset}\,(\text{Id})}{\emptyset\ \Gamma \triangleright \Delta, \bigvee\{\Gamma\}\ \emptyset}\,(\textstyle\bigvee_R)$$

Dually for $\bigwedge \mathcal{P}$ on the left.
IH: Assume the lemma holds for all refutations of height $\leq k$, and for all $\mathcal{S}, \Gamma, \Delta, \mathcal{P}$.
Induction step: Consider a refutation of height $k+1$, and the lowest rule application.

There are two cases:

Case 1: For all rules except (Refute), we can use the induction hypothesis for the premises to easily obtain the required derivation. For example, consider the $(\wedge_R)$ rule:

$$(\wedge_R)\ \dfrac{\mathcal{S}_1\ \Gamma \bowtie_1 \Delta, \varphi \wedge \psi, \varphi\ \mathcal{P}_1 \qquad \mathcal{S}_2\ \Gamma \bowtie_2 \Delta, \varphi \wedge \psi, \psi\ \mathcal{P}_2}{\mathcal{S}_1 \cup \mathcal{S}_2\ \Gamma \bowtie_0 \Delta, \varphi \wedge \psi\ \mathcal{P}_1 \cup \mathcal{P}_2}$$

Where $\bowtie_0 = \begin{cases} \triangleright & \text{if } \bowtie_1 = \triangleright \text{ and } \bowtie_2 = \triangleright \\ \triangleleft & \text{otherwise} \end{cases}$

Since the conclusion is refutable by assumption, we know that $\bowtie_0 = \triangleleft$, then by the condition of the rule one or both of $\bowtie_1$ and $\bowtie_2$ is also $\triangleleft$.

The cases when either $\bowtie_1 = \triangleleft$ or $\bowtie_2 = \triangleleft$ are straightforward. If both $\bowtie_1 = \triangleleft$ and $\bowtie_2 = \triangleleft$ then both premises are the roots of refutations of height $\leq k$. Then the induction hypothesis gives derivations δ_1 of $\emptyset \bigwedge \mathcal{P}_1, \Gamma \triangleright \Delta, \varphi\wedge\psi, \varphi\ \emptyset$ and δ_2 of $\emptyset \bigwedge \mathcal{P}_2, \Gamma \triangleright \Delta, \varphi\wedge\psi, \psi\ \emptyset$, from which we obtain the following derivation:

$$\dfrac{\dfrac{\dfrac{\delta_1}{\emptyset \bigwedge\mathcal{P}_1, \Gamma \ \triangleright\ \Delta, \varphi\wedge\psi, \varphi\ \emptyset}}{\emptyset \bigwedge\mathcal{P}_1, \bigwedge\mathcal{P}_2, \Gamma \ \triangleright\ \Delta, \varphi\wedge\psi, \varphi\ \emptyset}\ \text{LW} \quad \dfrac{\dfrac{\delta_2}{\emptyset \bigwedge\mathcal{P}_2, \Gamma \ \triangleright\ \Delta, \varphi\wedge\psi, \psi\ \emptyset}}{\emptyset \bigwedge\mathcal{P}_1, \bigwedge\mathcal{P}_2, \Gamma \ \triangleright\ \Delta, \varphi\wedge\psi, \psi\ \emptyset}\ \text{LW}}{\dfrac{\emptyset \bigwedge\mathcal{P}_1, \bigwedge\mathcal{P}_2, \Gamma \ \triangleright\ \Delta, \varphi\wedge\psi\ \emptyset}{\emptyset \bigwedge(\mathcal{P}_1\cup\mathcal{P}_2), \Gamma \ \triangleright\ \Delta, \varphi\wedge\psi\ \emptyset}\ (\bigwedge^{\cup}_L)}\ (\wedge_R)$$

Dually for $\bigvee\mathcal{S}$ on the right.

Case 2: Consider the (Refute) rule. That is, $\mathcal{S} = \{\Gamma'\}$ and hence $\emptyset\ \Gamma' \triangleright \Delta', \bigvee\mathcal{S}\ \emptyset$ is $\emptyset\ \Gamma' \triangleright \Delta', \bigvee\{\Gamma'\}\ \emptyset$. Then a derivation of $\emptyset\ \Gamma' \triangleright \Delta', \bigvee\{\Gamma'\}\ \emptyset$, where $\Gamma' = \Gamma, \varphi_1 \prec \psi_1, \cdots, \varphi_n \prec \psi_n = \{\gamma_1, \cdots, \gamma_k\}$ for some $k \geq 1$ is:

$$\dfrac{\dfrac{}{\emptyset\ \Gamma' \ \triangleright\ \Delta', \gamma_1\ \emptyset}\ (\text{Id}) \quad \cdots \quad \dfrac{}{\emptyset\ \Gamma' \ \triangleright\ \Delta', \gamma_k\ \emptyset}\ (\text{Id})}{\emptyset\ \Gamma' \ \triangleright\ \Delta', \bigvee\{\Gamma'\}\ \emptyset}\ (\bigvee_R)$$

Dually for $\bigwedge\mathcal{P}$ on the left. □

Remark 21. *Case 1 of the induction step in the previous proof shows why we need to keep variables as sets of sets and form the union of variables from all premises. If we only kept, say $\mathcal{P}_1$, at the conclusion of the $(\wedge_R)$ rule, the above case would not go through since we would not be able to show that the right premise is derivable.*

We now show that the $(\to_{R2})$ and $(\prec_{L2})$ rules are sound by showing how they absorb certain instances of (cut). Intuitively, we show that instead of guessing the cut formula required for a derivation, we can combine the variables at the root of the refutation of the left premise with a derivation of the right premise to obtain the derivation of the conclusion.

Lemma 22. *The $(\to_{R2})$ and $(\prec_{L2})$ rules are sound.*

Proof. We show the case for $(\to_{R2})$, the case for $(\prec_{L2})$ is symmetric.

$$(\to_{R2})\ \dfrac{\mathcal{S}\ \Gamma, \varphi \ \triangleleft\ \psi\ \mathcal{P} \qquad \mathcal{S}_1\ \Gamma\ \bowtie_1\ \Pi_1, \Delta, \varphi\to\psi\ \mathcal{P}_1 \quad \cdots \quad \mathcal{S}_m\ \Gamma\ \bowtie_m\ \Pi_m, \Delta, \varphi\to\psi\ \mathcal{P}_m}{\bigcup_1^m \mathcal{S}_i\ \Gamma\ \bowtie_0\ \Delta, \varphi\to\psi\ \bigcup_1^m \mathcal{P}_i}$$

There are two cases: either all the right premises are derivable, or at least one is refutable.

1. If all the right premises are derivable, then $\bowtie_0 = \triangleright$, i.e. the conclusion is also derivable. We show how to replace an instance of $(\to_{R2})$ with a sound instance of the cut rule.

   The left-most premise $\mathcal{S}\ \Gamma, \varphi \ \triangleleft\ \psi\ \mathcal{P}$ of $(\to_{R2})$ is refutable. Then by Lemma 20, there is a derivation δ_1 of the sequent $\emptyset \bigwedge\mathcal{P}, \Gamma, \varphi \ \triangleright\ \psi\ \emptyset$.

   All the right premises of $(\to_{R2})$ are derivable, that is, $\mathcal{S}_i\ \Gamma \ \triangleright\ \Delta, \varphi\to\psi, \Pi_i\ \mathcal{P}_i$ has a derivation δ_2^i, for $1 \leq i \leq n$ and $n \geq 1$. By Lemma 14, we have that $\mathcal{S}_i = \emptyset$ and $\mathcal{P}_i = \emptyset$, thus each δ_2^i is a derivation of $\emptyset\ \Gamma \ \triangleright\ \Delta, \varphi\to\psi, \Pi_i\ \emptyset$. Then let δ_2 be a derivation of $\emptyset\ \Gamma \ \triangleright\ \Delta, \varphi\to\psi, \bigwedge\mathcal{P}\ \emptyset$, where $\mathcal{P} = \{\Pi_1, \cdots, \Pi_n\}$ for $n \geq 1$, as shown below:

$$\dfrac{\begin{array}{c}\delta_2^1\\ \emptyset\ \Gamma \rhd \Delta, \varphi \to \psi, \Pi_1\ \emptyset\end{array} \quad \cdots \quad \begin{array}{c}\delta_2^n\\ \emptyset\ \Gamma \rhd \Delta, \varphi \to \psi, \Pi_n\ \emptyset\end{array}}{\emptyset\ \Gamma \rhd \Delta, \varphi \to \psi, \bigwedge \mathcal{P}\ \emptyset}\ (\bigwedge_R)$$

Then a cut on $\bigwedge \mathcal{P}$ gives a sound derivation of the conclusion of $(\to_{R2})$ as follows:

$$\dfrac{\begin{array}{c}\delta_2\\ \emptyset\ \Gamma \rhd \Delta, \varphi \to \psi, \bigwedge \mathcal{P}\ \emptyset\end{array} \quad \dfrac{\begin{array}{c}\delta_1\\ \emptyset\ \bigwedge \mathcal{P}, \Gamma, \varphi \rhd \psi\ \emptyset\end{array}}{\emptyset\ \bigwedge \mathcal{P}, \Gamma \rhd \Delta, \varphi \to \psi\ \emptyset}\ (\to_{R1})}{\emptyset\ \Gamma \rhd \Delta, \varphi \to \psi\ \emptyset}\ (\text{cut})$$

Thus, if $\emptyset\ \Gamma \rhd \Delta, \varphi \to \psi\ \emptyset$ is the conclusion of $(\to_{R2})$ in `GBiInt1`, then there is a `GBiInt0` derivation of $\emptyset\ \Gamma \rhd \Delta, \varphi \to \psi\ \emptyset$. We have used only the part of Lemma 20 relating to $\mathcal{P}$. The symmetric case of $(-\!\!<_{L2})$ requires the part relating to $\mathcal{S}$.

2. If any right premise is refutable, then $\bowtie_0 = \lhd$, i.e. the conclusion is also refutable. But the RHS of each right premise contains the RHS of the conclusion, while the LHSs are the same, so if any right premise is falsifiable, then the conclusion is also falsifiable.

□

We now illustrate the effect of the transformation in the previous lemma by showing the example derivation of Section 4.3 using an instance of cut instead of an instance of $(\to_{R2})$.

Example 23 (Derivation using cut)**.** *Below is a* `GBiInt1`*-derivation of Uustalu's [32] interaction formula* $p \to (q \vee (r \to ((p -\!\!< q) \wedge r)))$*, simplified to the sequent* $p \bowtie q, r \to ((p -\!\!< q) \wedge r)$*. Let* $X := (p -\!\!< q) \wedge r$ *and* $Y = r \to X$*. This derivation uses a cut on* $p -\!\!< q$ *instead of* $(\to_{R2})$ *and variables. All variables have a value of* $\emptyset$ *so we omit them to save space.*

$$\dfrac{\dfrac{\dfrac{}{q, p \rhd q, Y, p \prec q}\ Id \quad \dfrac{}{p \rhd p, q, Y, p \prec q}\ Id}{p \rhd q, Y, p \prec q}\ (\prec_R) \quad \dfrac{\dfrac{\dfrac{}{p \prec q, p, r \rhd X, p \prec q}\ Id \quad \dfrac{}{p \prec q, p, r \rhd X, r}\ Id}{p \prec q, p, r \rhd (p \prec q) \wedge r}\ (\wedge_R)}{p \prec q, p \rhd q, r \to ((p \prec q) \wedge r)}\ (\to_{R1})}{p \rhd q, r \to ((p \prec q) \wedge r)}\ cut$$

Comparing the derivation of Example 23 with that of Example 1 shows that their basic structure is the same. There are some notational differences since `GBiInt1` uses the "$\rhd$" turnstile and variables (which are empty in this case). Also, the cut rule in `GBiInt1` is additive rather than multiplicative to ensure an easy transformation of instances of $(\to_{R2})$ and $(-\!\!<_{L2})$ into cut. A more significant difference is the hidden contraction in $(\prec_R)$ and $(\wedge_R)$, where the principal formula is carried from the conclusion to the premises. As a consequence, the axioms of the derivation of Example 23 contain additional formulae to those found in Example 1. The additional formulae, and hence the contractions, are redundant in this case since the rules have produced a derivation, but would be essential otherwise.

5.3 Soundness and Completeness of GBiInt

The difference between GBiInt1 and GBiInt is that GBiInt omits the structural rules and the special rules for $\bigwedge$ and $\bigvee$. Thus GBiInt is sound, since it is a subset of GBiInt1, whose soundness we showed in the previous section. To show the completeness of GBiInt, we will use the fact that all the rules preserve the semantic judgement σ downwards, and that if a sequent is not derivable, then it is refutable. That is, a refutation gives a semantically correct counter-model, and we can obtain such a refutation whenever we cannot obtain a derivation. We now prove the precursor to the soundness and completeness corollaries.

Theorem 24. *For all $\mathcal{S}, \Gamma, \Delta, \mathcal{P}$:*

1. *If $\vdash \mathcal{S}\ \Gamma\ \triangleright\ \Delta\ \mathcal{P}$ then $\models_{\texttt{BiInt}} \tau(\mathcal{S}\ \Gamma\ \triangleright\ \Delta\ \mathcal{P})$.*
2. *If $\vdash \mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P}$ then $\Dashv_{\texttt{BiInt}} \tau(\mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P})$.*

Proof. We proceed by simultaneous induction on the height of the derivation or refutation. If the height is 1, we have a leaf node, so both cases 1 and 2 follow by Lemma 17. For all $\mathcal{S}, \Gamma, \Delta, \mathcal{P}$, assume the lemma holds for all derivations/refutations of height $\leq k$. Let ρ be the lowest rule application of a derivation/refutation of height $k+1$. The premises of ρ obey the IH. By Lemmas 16, 22 and 19, ρ preserves the semantic judgement σ downwards. Thus, cases 1 and 2 hold for the conclusion of ρ. ☐

In some sense, Theorem 24 is our main result, since it shows that derivability/refutability captures validity/falsifiability. But traditionally, we wish to obtain soundness, which says that derivability implies validity, and completeness, which says that validity implies derivability. The traditional soundness result easily follows from Theorem 24, as shown below:

Corollary 25 (Soundness)**.** *If $\vdash \mathcal{S}\ \Gamma\ \triangleright\ \Delta\ \mathcal{P}$ then $\models_{\texttt{BiInt}} \hat{\Gamma} \rightarrow \check{\Delta}$.*

Proof. By case 1 of Lemma 24, we have $\models_{\texttt{BiInt}} \tau(\mathcal{S}\ \Gamma\ \triangleright\ \Delta\ \mathcal{P})$. Then by Lemma 14, we have $\mathcal{S} = \emptyset = \mathcal{P}$ and thus $\models_{\texttt{BiInt}} \hat{\Gamma} \rightarrow \check{\Delta}$. ☐

Corollary 26 (Pre-completeness)**.** *If $\vdash \mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P}$ then $\Dashv_{\texttt{BiInt}} \hat{\Gamma} \rightarrow \check{\Delta}$.*

Proof. By case 2 of Lemma 24, if $\vdash \mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P}$ then $\Dashv_{\texttt{BiInt}} \tau(\mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P})$. That is, there exists a BiInt model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and a world $w \in \mathcal{W}$ such that $w \Vdash \bigvee \mathcal{S}, \Gamma$ and $w \dashv\!| \Delta, \bigwedge \mathcal{P}$. Then clearly $w \Vdash \Gamma$ and $w \dashv\!| \Delta$, that is, $\Dashv_{\texttt{BiInt}} \hat{\Gamma} \rightarrow \check{\Delta}$. ☐

For full completeness, we also need decidability, which we establish next.

6 Decision Procedure and Complexity

In this and the subsequent section, we concentrate only on GBiInt, that is, the sequent calculus without cut and special rules, but with the derived transitional rules $(\rightarrow_{R2})$ and $(\prec_{L2})$. We show that GBiInt is terminating and gives a decision procedure for BiInt. Indeed, we can even create GBiInt trees automatically, using backward search.

We start with $\Gamma \bowtie \Delta$, where $\bowtie$ is unknown, and we want to determine whether $\bowtie = \triangleright$ or $\bowtie = \triangleleft$. We apply the rules of `GBiInt` backwards, using the systematic procedure outlined in Figure 3. When the recursive calls of the procedure return, they replace $\bowtie$ with either $\triangleright$ or $\triangleleft$, depending on the derivability/refutability of the subtrees and thus the appropriate rule form. At the end, our rules will deliver either $\emptyset\ \Gamma\ \triangleright\ \Delta\ \emptyset$ or $\mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P}$.

We first outline a simple blocking condition to ensure termination.

Definition 27 (Blocking condition)**.** *Let ρ be a rule with $n \geq 1$ premises π_i, for $1 \leq i \leq n$, and conclusion γ. Apply ρ backwards only if: $\forall \pi_i.(LHS_{\pi_i} \not\subseteq LHS_\gamma$ or $RHS_{\pi_i} \not\subseteq RHS_\gamma)$.*

Intuitively, the general blocking condition of Definition 27 states that we apply a rule backwards only if the application adds new formulae to the current sequent, for otherwise we would be in a loop. This simple condition, the persistence property of `BiInt` and the contractions built into our static rules ensures termination, as we show later in this section.

Remark 28. *Note that Definition 27 implicitly includes the following:*

$$\text{if } \rho = (\to_{R2}) \text{ then } \forall \Pi \in \mathcal{P}.\Pi \not\subseteq RHS_\gamma \tag{6.1}$$

$$\text{if } \rho = (-\!\!<_{L2}) \text{ then } \forall \Sigma \in \mathcal{S}.\Sigma \not\subseteq LHS_\gamma \tag{6.2}$$

From a backward search perspective, the only difference between $(\to_{R2})$, $(-\!\!<_{L2})$ and the other rules is that we must receive the variables from the left-most premise, before we can determine whether or not to create the right premises and fully apply the rule.

The intuition behind the classification of the logical rules in Section 4.2 is that backwards applications of static rules add formulae to the current world in the counter-model, transitional rules $(\to_{R1})$ and $(-\!\!<_{L1})$ create new worlds and add formulae to them, transitional rules $(\to_{R2})$ and $(-\!\!<_{L2})$ update existing worlds with new interaction formulae received from successors/predecessors, and the transitional rule (Refute) moves back towards the root of the counter-model when successors/predecessors do not return any new information. The classification justifies the search strategy defined in Figure 3.

Definition 29. *For a `BiInt`-formula φ, the subformulae $sf(\varphi)$ are defined as usual. The subformulae of an extended `BiInt`-formula $\bigvee \mathcal{Q}$ or $\bigwedge \mathcal{Q}$ are the subformulae of its members.*

To show that the procedure in Figure 3 gives us a decision procedure, we need to show that for all input $\Gamma \bowtie \Delta$, it terminates and returns either true, meaning $\vdash \emptyset\ \Gamma\ \triangleright\ \Delta\ \emptyset$, or false, meaning $\vdash \mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P}$. We show termination first. So let $m = |sf(\Gamma \cup \Delta)|$.

Definition 30 (LEN)**.** *Let $>_{len}$ be a lexicographic ordering of sequents:*
$(\Gamma_2 \bowtie \Delta_2) >_{len} (\Gamma_1 \bowtie \Delta_1)$ *iff* $[(|\Gamma_2| > |\Gamma_1|)$ *or* $(|\Gamma_2| = |\Gamma_1|$ *and* $|\Delta_2| > |\Delta_1|)]$.

Definition 31. *We use **successor rules** (resp. **predecessor rules**) to refer to $(\to_{R1})$, $(\to_{R2})$ and (Refute) (resp. $(-\!\!<_{L1})$, $(-\!\!<_{L2})$ and (Refute)). We use **successor premises** (resp. **predecessor premises**) to refer to the premise of $(\to_{R1})$, the left premise of $(\to_{R2})$, and the $\to$-premises of (Refute) (resp. the premise of $(-\!\!<_{L1})$, the left premise of $(-\!\!<_{L2})$, and the $-\!\!<$-premises of (Refute)). We use **transitional premises** to refer to both predecessor premises and successor premises.*

Function Decide
Input: sequent $\pi_0 = \Gamma \bowtie \Delta$
Output: true (meaning $\vdash \emptyset\ \Gamma\ \triangleright\ \Delta\ \emptyset$) or false (meaning $\vdash \mathcal{S}\ \Gamma\ \triangleleft\ \Delta\ \mathcal{P}$)

1. If $\rho \in \{(\mathrm{Id}), (\bot_L), (\top_R)\}$ is applicable to π_0 then return true
2. Else if $\rho = (\mathrm{Ret})$ is applicable to π_0 then return false
3. Else if ρ is a static rule that is applicable to π_0 then
   (a) Let $\pi_1, \cdots, \pi_n$ be the premises of ρ obtained from π_0
   (b) Return $\bigwedge_{i=1}^n \mathit{Decide}(\pi_i)$
4. Else if $\mathit{Decide}(\pi_1) =$ true for some premise instance π_1 obtained from π_0 via $\rho = (\rightarrow_{R1})$ then return true
5. Else if $\mathit{Decide}(\pi_1) =$ true for some premise instance π_1 obtained from π_0 via $\rho = (\prec_{L1})$ then return true
6. Else if $\mathit{Decide}(\pi) =$ false for some left premise instance π obtained from π_0 via $\rho = (\rightarrow_{R2})$ and condition 6.1 is met then
   (a) Let π_i for $1 \leq i \leq n$ and $n \geq 1$ be the right premises of ρ obtained from π_0
   (b) Return $\bigwedge_{i=1}^n \mathit{Decide}(\pi_i)$
7. Else if $\mathit{Decide}(\pi) =$ false for some left premise instance π of obtained from π_0 via $\rho = (\prec_{L2})$ and condition 6.2 is met then
   (a) Let π_i for $1 \leq i \leq n$ and $n \geq 1$ be the right premises of ρ obtained from π_0
   (b) Return $\bigwedge_{i=1}^n \mathit{Decide}(\pi_i)$
8. Else $\rho = (\mathrm{Refute})$ must be applicable to π_0. Apply ρ and return false.
9. Endif

We have left out the variables for simplicity, but in each return statement it is implicit that the variables are returned as specified in the conclusion of the rules defined in Section 4.2. Also, $\bigwedge_{i=1}^n \mathit{Decide}(\pi_i)$ is true iff $\mathit{Decide}(\pi_i)$ is true for all premises π_i for $1 \leq i \leq n$.

Figure 3: Search strategy.

Definition 32. *Given a* `GBiInt`*-tree $\mathcal{T}$ and a branch or part thereof $\mathcal{B}$ in $\mathcal{T}$, we say that $\mathcal{B}$ is* ***successor-only*** *if $\mathcal{B}$ contains only applications of static rules, $(\rightarrow_{R1})$, $(\rightarrow_{R2})$, and successor premises. Similarly, $\mathcal{B}$ is* ***predecessor-only*** *if $\mathcal{B}$ contains only applications of static rules, $(\prec_{L1})$, $(\prec_{L2})$, and predecessor premises. A branch is* ***single-directional*** *if it is successor-only or predecessor-only. Finally, a branch contains a* ***direction switch*** *if* *it is not single-directional.*

Lemma 33. *Every single-directional branch of every* GBiInt*-tree is $\mathcal{O}(m^2)$ long.*

Proof. We prove only the successor-only case since the predecessor-only case is symmetric.

We show that on every successor-only branch, the length of a sequent defined via $>_{len}$ increases with every rule application, and that it can increase $\mathcal{O}(m^2)$ times.

Consider a rule ρ, and a backwards application of ρ to some $\Gamma \bowtie \Delta$, which yields n premises $\Gamma_i \bowtie \Delta_i$, where $1 \leq i \leq n$. If ρ is a static rule, then for all premises i, we have $(\Gamma_i \bowtie \Delta_i) >_{len} (\Gamma \bowtie \Delta)$ from the generalised blocking condition (Definition 27). We only show the case for $\rho = (\rightarrow_{R1})$ since the other cases are similar:

Case $\rho = (\rightarrow_{R1})$: The principal formula is $\varphi \rightarrow \psi$. Consider the premise $\Gamma_1 \bowtie \Delta_1$. According to our strategy, the $(\rightarrow^I_R)$ rule has already been applied and thus $\psi \in \Delta$, so $(\rightarrow_{R1})$ is applied only if $\varphi \notin \Gamma$. Therefore, for the premise, we have $|\Gamma_1| > |\Gamma|$.

The length of a sequent can increase either by adding a subformula to the LHS, or by keeping the LHS unchanged and adding a subformula to the RHS. We can add a subformula to the LHS at most m times. After each such addition, the length of the RHS either remains the same (if a static rule was applied) or decreases to 1 (if $(\rightarrow_{R1})$, $(\rightarrow_{R2})$ or (Refute) was applied). In the latter case, we can again add a subformula to the RHS at most $m-1$ times. Hence the length can increase $\mathcal{O}(m^2)$ times. □

Definition 34 (Degree)**.** *The degree of a* BiInt *formula φ is the number of $\rightarrow$ and $\prec$ connectives in φ. The degree of a sequent $\Gamma \bowtie \Delta$ is defined as: $deg(\Gamma \bowtie \Delta) = \sum_{\varphi \in sf(\Gamma \cup \Delta)} deg(\varphi)$.*

Corollary 35. *By the subformula property, the degree of a sequent cannot increase in backward search. For any sequents γ_1 and γ_2, $deg(\gamma_2) < deg(\gamma_1)$ if $sf(\gamma_2) \subsetneq sf(\gamma_1)$.*

That is, removing some formula φ from a sequent during backward search decreases the degree of the sequent if φ is not a subformula of any other formula in the sequent since φ no longer contributes to the sum of degrees of subformulae.

Lemma 36. *Every rule of* GBiInt *has a finite number of premises.*

Proof. Obvious for all rules except $(\rightarrow_{R2})$ and $(\prec_{L2})$. For $(\rightarrow_{R2})$ and $(\prec_{L2})$, the number of premises is $1+n$, where n is the number of sets in the variable $\mathcal{S}$ or $\mathcal{P}$ of the left premise. But both $\mathcal{S}$ and $\mathcal{P}$ are subsets of the powerset of $sf(\Gamma \cup \Delta)$ of the end sequent $\Gamma \bowtie \Delta$. Therefore, each of $\mathcal{S}$ and $\mathcal{P}$ are of finite size $\mathcal{O}(2^m)$, where $m = |sf(\Gamma \cup \Delta)|$. □

Lemma 37. *Let $\mathcal{B}$ be any branch of any* GBiInt *tree that contains a direction switch, and let π_0 be the conclusion of a successor (resp. predecessor) rule and let π_1 be the premise of a predecessor (resp. successor) rule. Then $deg(\pi_1) < deg(\pi_0)$.*

Proof. We do the case where $\mathcal{B}$ contains $(\rightarrow_{R1})$ and $(\prec_{L1})$: see Figure 4. An inspection of the rules in Section 4.2 shows that expanding π_0 during backward search using $(\rightarrow_{R2})$ with the principal formula $\varphi_0 \rightarrow \psi_0$ yields the same successor premise $\Gamma_0, \varphi_0 \bowtie \psi_0$ as

$$
\begin{array}{rc}
 & \vdots \\
\cline{2-2}
 & \pi_1 = (\varphi_1 \bowtie \psi_1, \Delta_1) \\
(-\!\!<_{L1}) & \\
\cline{2-2}
 & \Gamma_1, \varphi_1 -\!\!< \psi_1 \bowtie \Delta_1 \\
\cline{2-2}
 & \vdots \\
\cline{2-2}
 & \Gamma_0, \varphi_0 \bowtie \psi_0 \\
(\to_{R1}) & \\
\cline{2-2}
 & \pi_0 = (\Gamma_0 \bowtie \Delta_0, \varphi_0 \to \psi_0) \\
\cline{2-2}
 & \vdots
\end{array}
$$

Figure 4: Switching premises

using $(\to_{R1})$. Similarly for the corresponding $\to$-premise of (Refute), and symmetrically for predecessor premises. Thus all other cases of direction switches are equivalent from a backward search perspective.

Let $\chi \in sf(\pi_0)$ be some formula such that $deg(\chi) = max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$: that is, χ is one of the subformulae with the maximum degree. In particular, this means that χ is not a subformula of any formula with a larger degree. We shall now show that $\chi \notin sf(\pi_1)$.

There are two cases:

$\chi \notin sf(\Gamma_0)$: Then $\chi \in sf(\Delta_0)$ or $\chi = \varphi_0 \to \psi_0$. In both cases, $\chi \notin sf(\pi_1)$.

$\chi \in sf(\Gamma_0)$: Then $\chi \in sf(\varphi_1)$ or $\chi \in sf(\psi_1)$ implies $deg(\varphi_1 -\!\!< \psi_1) > deg(\chi)$, contradicting our assumption that $deg(\chi) = max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$. Therefore, either:

- χ and all its occurrences in subformulae disappear from the sequent at the premise of $(-\!\!<_L)$, in which case $\chi \notin sf(\pi_1)$, or
- χ is moved to the RHS of the sequent by applying the $(\to_L)$ rule to some formula $\chi \to \tau \in sf(\Gamma_0)$. However, since $deg(\chi \to \tau) > deg(\chi)$, this again contradicts our assumption that $deg(\chi) = max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$.

We have shown that for some formula χ we have $\chi \in sf(\pi_0)$ and $\chi \notin sf(\pi_1)$. Also, by the subformula property of `GBiInt` we have $sf(\pi_1) \subseteq sf(\pi_0)$. Together with $\chi \in sf(\pi_0)$ and $\chi \notin sf(\pi_1)$, this means $sf(\pi_1) \subsetneq sf(\pi_0)$. Then by Corollary 35 we have $deg(\pi_1) < deg(\pi_0)$. Note that the steps indicated by vertical ellipses (dots) in Figure 4 are arbitrary, since by Corollary 35 no rule can increase the degree of a sequent. Since $deg(\pi_1) < deg(\pi_0)$, every direction switch must decrease the degree of the sequent. □

Lemma 38. *Any branch in any* `GBiInt` *tree built via the strategy of Fig. 3 is* $\mathcal{O}(m^3)$ *long.*

Proof. By Lemma 33, we can move in one direction $\mathcal{O}(m^2)$ times, before we must stop or change direction. By Lemma 37, every direction change decreases the degree of the sequent. We can change direction $\mathcal{O}(m)$ times since the degree of the end sequent is $\left(\sum_{\varphi \in sf(\Gamma \cup \Delta)} deg(\varphi)\right) = \mathcal{O}(m)$. Thus, every branch has length $\mathcal{O}(m^2) \times \mathcal{O}(m) = \mathcal{O}(m^3)$. □

Theorem 39 (Termination)**.** *Every* GBiInt*-tree built via the strategy of Figure 3 is finite.*

Proof. By Lemmas 36 and 38, every tree is finitely branching, and every branch is finite. □

Note that the strategy of Figure 3 is required for both completeness and termination. In particular, transitional rules must be applied only to saturated sequents as this blocks the transitional rules from creating an infinite branch by repeatedly using the same formula as the principal formula. The other aspects of the strategy are required for completeness.

Theorem 40 (Decision procedure)**.** *For every* $\Gamma \bowtie \Delta$*, there is an effective decision procedure to decide whether* $\vdash \emptyset\ \Gamma \triangleright \Delta\ \emptyset$ *or* $\vdash \mathcal{S}\ \Gamma \triangleleft \Delta\ \mathcal{P}$*, where* $\mathcal{S} \subseteq 2^{sf(\Gamma \cup \Delta)}$ *and* $\mathcal{P} \subseteq 2^{sf(\Gamma \cup \Delta)}$*.*

Proof. By Theorem 39, the backward search procedure of Figure 3 terminates for all Γ, Δ. It is clear that cases 1 to 8 of Figure 3 are exhaustive, thus it is fully deterministic and always returns an answer of either true ($\vdash \emptyset\ \Gamma \triangleright \Delta\ \emptyset$) or false ($\vdash \mathcal{S}\ \Gamma \triangleleft \Delta\ \mathcal{P}$). □

We can now obtain traditional completeness and its "dual" as corollaries:

Corollary 41 (Completeness)**.** *If* $\models_{\texttt{BiInt}} \hat{\Gamma} \to \check{\Delta}$ *then* $\vdash \emptyset\ \Gamma \triangleright \Delta\ \emptyset$*.*

Proof. Suppose $\models_{\texttt{BiInt}} \hat{\Gamma} \to \check{\Delta}$. Run our procedure on $\Gamma \bowtie \Delta$, and obtain by Theorem 40 that either $\vdash \emptyset\ \Gamma \triangleright \Delta\ \emptyset$ or $\vdash \mathcal{S}\ \Gamma \triangleleft \Delta\ \mathcal{P}$. In the first case we are done, since we have shown what was required. In the second, Theorem 26 gives us that $\Dashv_{\texttt{BiInt}} \hat{\Gamma} \to \check{\Delta}$. But this contradicts our assumption that $\models_{\texttt{BiInt}} \hat{\Gamma} \to \check{\Delta}$. Hence the second case is impossible. □

Corollary 42. *If* $\Dashv_{\texttt{BiInt}} \hat{\Gamma} \to \check{\Delta}$ *then* $\vdash \mathcal{S}\ \Gamma \triangleleft \Delta\ \mathcal{P}$ *for some* $\mathcal{S}$ *and* $\mathcal{P}$*.*

Proof. Symmetric to the proof of Corollary 41. □

Lemma 43. *A* GBiInt *sequent takes* $\mathcal{O}(2^m)$ *space.*

Proof. A GBiInt sequent $\mathcal{S}\ \Gamma \bowtie \Delta\ \mathcal{P}$ consists of 4 components. Each of Γ and Δ are of size $\mathcal{O}(m)$, and each of the variables $\mathcal{S}$ and $\mathcal{P}$ are subsets of the powerset of $sf(\Gamma \cup \Delta)$. Therefore, each of $\mathcal{P}$ and $\mathcal{S}$ are of size $\mathcal{O}(2^m)$, and the overall sequent is of size $\mathcal{O}(2^m)$. □

Theorem 44 (Complexity)**.** *Our decision procedure* GBiInt *takes* $\mathcal{O}(2^m)$ *space.*

Proof. Since our decision procedure performs depth-first construction/traversal of GBiInt trees, it suffices to show that any path of a GBiInt tree takes $\mathcal{O}(2^m)$ space. By Lemma 38, any path is at most of polynomial length, and by Lemma 43, each sequent on such a path uses at most exponential space. Therefore, any path of a GBiInt tree takes $\mathcal{O}(2^m)$ space. □

Given a graph, a cluster is a set of nodes which form a strongly connected component. A cluster is proper if it contains more than one node. A BiInt frame is rooted if there exists a root world w such that every world u can be reached from w by following $\mathcal{R}$-edges or $\mathcal{R}^{-1}$-edges. The next corollary follows directly from termination and from our construction in the proof of Lemma 19 since we never create proper clusters, i.e., we do not reuse worlds.

Corollary 45. `BiInt` *is characterised by finite rooted reflexive and transitive frames with no proper clusters.*

Theorem 46. *The decision problem for bi-intuitionistic logic is in* PSPACE.

Proof. We can easily extend the polynomial Gödel translation of intuitionistic logic into the modal logic `S4` [13], to a translation of bi-intuitionistic logic into the tense modal logic $\mathtt{K_tS4}$. Since $\mathtt{K_tS4}$ is a fragment of $ALCI_{R^+}$, a description logic with transitive and inverse roles, and $ALCI_{R^+}$ is in PSPACE [19], we know that bi-intuitionistic logic is also in PSPACE. □

An exponential-space decision procedure for a PSPACE problem may seem suboptimal, especially since a PSPACE decision procedure for a very similar logic [19] already exists. But the situation is not that simple as we show next. First, our main aim is developing a cut-free sequent calculus and a proof-theoretic perspective on bi-intuitionistic logic. This is important if we later want to extend our work to type theory and give a computational interpretation in the "proofs as types" paradigm, or extend our work to other non-classical logics. Second, while the initial algorithm given by Horrocks et al. is indeed in PSPACE , it is too inefficient in practice due to the many restarts: *"the technique is not used in practice as rebuilding the discarded parts of the completion tree can be very costly"* [18]. In fact, the usual approach is to implement decision procedures for logics with inverse roles without the depth-first strategy, but this makes it necessary to *"save the state of the whole completion tree at each* $\vee$*-rule application"* [18], which is similar to our encoding of the open branches using sets-of-sets. Thus practical decision procedures do not necessarily have to be optimal, meaning that our suboptimal approach for `BiInt` is not as bad as it may appear.

Finally, the space usage of our algorithm is amenable to optimisation. We can make a number of observations about how the formulae in the variables are passed down from leaves towards the root, and store them in more efficient data structures than sets of sets. For example, an approach like frequent-pattern trees [10] can be used to efficiently store sets containing overlapping elements. Additionally, it is likely that some of the traditional optimisations for tableau calculi [20] are still applicable in the intuitionistic case.

7 Related Work

Other sequent calculi for `BiInt`. As mentioned before, Uustalu has recently given a counter-example [32] to Rauszer's cut-elimination theorem [26]. Uustalu's counterexample also shows that Crolard's sequent calculus [4] for `BiInt` is not cut-free. Uustalu's counterexample fails in both Rauszer's and Crolard's calculi because they limit certain sequent rules to singleton succedents or antecedents in the conclusion, and the rules do not capture the "forward" and "backward" interaction between implication and exclusion.

Uustalu and Pinto [33] have also given a cut-free sequent-calculus for `BiInt`. Since only the abstract of this work has been published so far, we have not been able to examine their sequent rules, or verify their proofs. According to the abstract [33] and personal communication with Uustalu [32], their calculus uses labelled formulae, thereby utilising some semantic aspects, such as explicit worlds and accessibility, directly in the rules. On the other hand, our calculus GBiInt is purely syntactic and our variables $\mathcal{S}$ and $\mathcal{P}$ have no semantic content, although they clearly have some proof/refutation search content.

If we were interested only in decision procedures, we could obtain a decision procedure for BiInt by embedding it into the tense logic Kt.S4 [34], and using tableaux for description logics with inverse roles [19]. However, an embedding into Kt.S4 provides no proof-theoretic insights into BiInt itself. Moreover, the restart technique of Horrocks et al. [19] involves non-deterministic expansion of disjunctions, which is complicated by inverse roles. As mentioned previously, their actual implementation avoids this non-determinism by keeping a global view of the whole counter-model under construction. In contrast, we handle this non-determinism by syntactically encoding it using variables and extended sequents.

Comparison with other calculi for Int. Recall that LJT [9] is a traditional sequent calculus for Int, and CRIP [25] is a refutation calculus for Int. Although we developed GBiInt independently from these calculi, we now compare the three calculi.

In LJT and in other traditional sequent calculi, one relates the syntactic judgement of derivability to the semantic judgement of validity by showing that the rules preserve validity downwards (if the premises are valid, then the conclusion is valid). Similarly, in CRIP, one shows that the rules preserve falsifiability downwards (if the premises are falsifiable then the conclusion is falsifiable). Both are local notions referring to a single rule. For completeness or sufficiency in traditional sequent calculi, one typically shows that a counter-model can be constructed from a failed proof search, referring to a global notion of failure.

Because GBiInt contains both derivations and refutations, we show that GBiInt rules locally preserve the generalised semantic judgement (either validity or falsifiability) downwards. In addition, the side conditions in some of our refutation rules incrementally encode aspects of proof search failure that allow us to directly construct a counter-model from a refutation; thus proof search and refutation search are interleaved in our decision procedure. Then all we need to show for completeness or sufficiency is that for every input sequent, our calculus will produce either a derivation or a refutation.

Since the (Refute) rule of GBiInt serves a similar function to rule (11) in CRIP [25], our proof of Lemma 19 bears similarities to a part of the counter-model construction for CRIP. Indeed, if we were to restrict GBiInt to the Int (or the DualInt) fragment of BiInt, we would obtain a calculus whose refutation part is similar to CRIP, with the major difference being the termination mechanism. CRIP and LJT achieve termination using four separate contraction-free implication left rules each of which inspects the structure of the formula φ in $\varphi \to \psi$. A previous attempt to extend this idea to the case where φ is of the form $\chi \prec \xi$ was unsuccessful [7]. Because BiInt requires contraction in the implication left and exclusion right rules for completeness, our termination mechanism relies on the implicit contractions in all our static rules and the generalised blocking conditions.

Other termination mechanisms. There are other ways to obtain a terminating sequent calculus for Int using contraction-free calculi [9] or history methods [17, 21]. However, these methods are less suitable when the interaction between Int and DualInt formulae needs to be considered, since they erase potentially relevant formulae too soon during backward proof search. Moreover, we found it easier to prove semantic completeness with our method than with history-based methods since both Heuerding et al. [17] and Howe [21] prove completeness using syntactic transformations of derivations. Consequently, while `GBiInt` is sound and complete for the `Int` (and the `DualInt`) fragment of `BiInt`, it is unlikely to be as efficient on the fragment as these specific calculi. Dynamic blocking [19] is another possible solution for termination, but it also requires histories in a sequent calculus setting.

Other extended sequent calculi. Many other extended sequent mechanisms give cut-free sequent calculi for complicated logics where traditional sequent calculi fail, and some either have been applied, or could easily be applied, to `BiInt`: e.g. display calculi [15], deep sequent calculi [2] and hypersequents [1]. However, all involve a large degree of non-determinism, which is problematic for proof search.

8 Conclusions and Further Work

We have given a sequent calculus where derivations and refutations interact as first class citizens, and shown that this interaction gives us a cut-free sequent calculus for bi-intuitionistic logic, where more traditional sequent calculi have failed. Our rules preserve the generalised judgement (validity or falsifiability) downwards. Together with our terminating decision procedure, this gives four corollaries: traditional soundness (derivability implies validity) and completeness (validity implies derivability), as well as refutation correctness (refutability implies falsifiability) and refutation completeness (falsifiability implies refutability).

We have shown the soundness of our interaction rules by showing that they absorb certain cuts, but it would be useful to show that they can absorb all cuts. We already know that the cut rule is not required to achieve completeness, but a syntactic proof of cut-elimination would provide more insights into how refutations and derivations interact. Cut-elimination may be complicated by the "contractions-above-cut" problem since some of our rules contain implicit contractions. We are also interested in other by-products of cut-elimination like a Curry-Howard correspondence with a strongly normalising λ-calculus.

A simple implementation is available at `http://users.rsise.anu.edu.au/~linda`. An efficient implementation of our decision procedure is our next goal.

Acknowledgements. We would like to thank Alwen Tiu for providing useful comments and suggestions, and pointing out errors in a draft of this paper.

References

[1] A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In W. Hodges et al., editor, *Logic: from foundations to applications. Proc. Logic Colloquium, Keele, UK, 1993*, pages 1–32. OUP, New York, 1996.

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

[3] L. Buisman and R. Goré. A cut-free sequent calculus for bi-intuitionistic logic. In N. Olivetti, editor, *Proc. TABLEAUX 2007*, LNAI 4548:90–106. Springer, 2007.

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

[5] T. Crolard. A formulae-as-types interpretation of Subtractive Logic. *Journal of Logic and Computation*, 14(4):529–570, August 2004.

[6] J. Czermak. A remark on Gentzen's calculus of sequents. *NDJFL*, 18(3):471–474, 1977.

[7] E. Donovan. *Automated Proof Search in Bi-Intuitionistic Logic using Sequent Calculi*, Honours Thesis, Dept. of Computer Science, The Australian National University, 2005.

[8] A. Dragalin. *Mathematical Intuitionism: Introduction to Proof Theory*, volume 68 of *Translations of Mathematical Monographs*. Cambridge Univ. Press, 1988.

[9] R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, September 1992.

[10] J. Han et al. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

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

[12] G. Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1935. English translation in [30].

[13] K. Gödel. Eine interpretation des intuitionistischen aussagenkalkuls. *Ergebnisse eines Mathematischen Kolloquiums.*, 4:39–40, 1933. English translation in [11].

[14] V. Goranko. Refutation systems in modal logic. *Studia Logica*, 53(2):299–324, 1994.

[15] R. Goré. Substructural logics on display. *Logic J. of the IGPL*, 6(3):451–504, 1998.

[16] R. Goré. Tableau methods for modal and temporal logics. In D'Agostino at al, editor, *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.

[17] A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Proc. TABLEAUX 1996*, LNAI: 1071:210–225, 1996.

[18] I. Horrocks, U. Hustadt, U. Sattler, and R. Schmidt. Computational modal logic. In P. Blackburn et al., editor, *Handbook of Modal Logic*, pages 181–245. Elsevier, 2007.

[19] I. Horrocks, U. Sattler, and S. Tobies. A PSpace-algorithm for deciding $ALCNI_{R^+}$-satisfiability. Technical Report LTCS-98-08, LuFG TCS, RWTH Aachen, 1998.

[20] I. Horrocks and P. Patel-Schneider. Optimizing description logic subsumption. Journal of Logic and Computation 9(3):267–293, 1999.

[21] J. M. Howe. *Proof search issues in some non-classical logics.* PhD thesis, University of St Andrews, 1998.

[22] D. Larchey-Wendling. Combining proof-search and counter-model construction for deciding Gödel-Dummett logic. In *Proc. CADE-18*, LNCS:2392:94–110, 2002.

[23] J. Łukasiewicz. *Aristotle's syllogistic from the standpoint of modern formal logic.* Clarendon Press, Oxford, 2nd edition, 1957.

[24] S. Maehara. Eine Darstellung intuitionistic Logik und der Klassischen. Nagoya Math. Journal, Vol. 7, pp. 45-64, 1954.

[25] L. Pinto and R. Dyckhoff. Loop-free construction of counter-models for intuitionistic propositional logic. In *Symposia Gaussiana*, pp 225–232, 1995.

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

[27] C. Rauszer. An algebraic and Kripke-style approach to a certain extension of intuitionistic logic. *Dissertationes Mathematicae*, 168, 1980. Institute of Mathematics, Polish Academy of Sciences.

[28] S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Proc. TABLEAUX 1998*, LNAI 1397:277–292, Springer 1998.

[29] V. Śvejdar. On sequent calculi for intuitionistic propositional logic. *Commentationes Mathematicae Universitatis Carolinae*, 47(1):159–173, 2006.

[30] M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen.* Studies in Logic and the foundations of Mathematics. North-Holland, Amsterdam, 1969.

[31] I. Urbas. Dual-intuitionistic logic. *NDJFL*, 37(3):440–451, 1996.

[32] T. Uustalu. Personal communication. Via email, 2004 & 2006.

[33] 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, 2006.

[34] F. Wolter. On logics with coimplication. *J. of Philosophical Logic*, 27(4):353–387, 1998.