A Cut-free Sequent Calculus for Bi-Intuitionistic Logic

Linda Buisman[1] and Rajeev Goré[12]

[1] The Australian National University
Canberra ACT 0200, Australia
[2] Logic and Computation Programme
Canberra Research Laboratory, NICTA*, Australia
{Linda.Buisman|Rajeev.Gore}@anu.edu.au

Abstract. Bi-intuitionistic logic is the extension of intuitionistic logic with a connective dual to implication. Bi-intuitionistic logic was introduced by Rauszer as a Hilbert calculus with algebraic and Kripke semantics. But her subsequent "cut-free" sequent calculus for `BiInt` has recently been shown by Uustalu to fail cut-elimination. We present a new cut-free sequent calculus for `BiInt`, and prove it sound and complete with respect to its Kripke semantics. Ensuring completeness is complicated by the interaction between implication and its dual, similarly to future and past modalities in tense logic. Our calculus handles this interaction using extended sequents which pass information from premises to conclusions using variables instantiated at the leaves of failed derivation trees. Our simple termination argument allows our calculus to be used for automated deduction, although this is not its main purpose.

1 Introduction

Propositional intuitionistic logic (`Int`) has connectives $\rightarrow$, $\wedge$, $\vee$ and $\neg$, with $\neg\varphi$ definable as $\neg\varphi := \varphi \rightarrow \bot$. Propositional dual intuitionistic logic (`DualInt`) has connectives $-\!\!<$, $\wedge$, $\vee$ and $\sim$, with $\sim\varphi$ definable as $\sim\varphi := \top -\!\!< \varphi$. Bi-intuitionistic logic (`BiInt`) or subtractive logic or Heyting-Brouwer logic is the union of `Int` and `DualInt`. It is a conservative extension of both and was first studied by Rauszer [11, 12].

Rauszer's Kripke semantics for `BiInt` involve a reflexive and transitive binary relation $\mathcal{R}$, and its converse $\mathcal{R}^{-1}$, similar to the normal **tense** logic `Kt.S4`. Specifically, a world w makes $\varphi \rightarrow \psi$ true if every $\mathcal{R}$-successor v that makes φ true also makes ψ true, and a world w makes $\varphi -\!\!< \psi$ true if there exists an $\mathcal{R}$-predecessor v where φ holds but ψ does not. Thus, $\varphi -\!\!< \psi$ ("φ *excludes* ψ") is a natural dual to $\varphi \rightarrow \psi$ ("φ *implies* ψ").

While there are many cut-free sequent systems for `Int` (e.g., [15, 6, 5]) and `DualInt` (e.g., [16, 4]), the case for `BiInt` is less satisfactory. Rauszer presented

* National ICT Australia is funded by the Australian Government's Dept of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program.

a sequent calculus for BiInt in [11] and "proved" it cut-free, but Uustalu [17] has recently shown that the BiInt-valid formula $p \to (q \vee (r \to ((p -\!\!< q) \wedge r))$ cannot be derived in Rauszer's calculus without the cut rule. Uustalu's example also shows that Crolard's sequent calculus [3] for BiInt is not cut-free. Uustalu's example fails in these calculi because certain sequent rules are restricted to singleton succedents or antecedents in their conclusions, and these fail to capture the interaction between $\to$ and $-\!\!<$. Uustalu and Pinto have apparently given a cut-free sequent-calculus for BiInt [19, 18] using labelled formulae which use the Kripke semantics directly in the rules. But we have been unable to examine their rules or proofs, as only the abstract of their work has been published.

We present a new purely syntactic cut-free sequent calculus for BiInt. We avoid Rauszer's and Crolard's restrictions on the antecedents and succedents for certain rules by basing our rules on Dragalin's GHPC [5] which allows multiple formulae on both sides of sequents. To maintain intuitionistic soundness, we restrict the *premise* of the implication-right rule to a singleton in the succedent. Dually, the premise of our exclusion-left rule is restricted to a singleton in the antecedent. But using Dragalin's calculus and its dual does not give us BiInt completeness. We therefore follow Schwendimann [13], and use sequents which pass relevant information from premises to conclusions using variables instantiated at the leaves of failed derivation trees. We then recompute parts of our derivation trees using the new information, similarly to the restart technique of [9]. Our calculus thus uses a purely syntactic addition to traditional sequents, rather than resorting to a semantic mechanism such as labels. Our termination argument also relies on two new rules from Śvejdar [14].

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 [20], and using tableaux for description logics with inverse roles [9]. However, an embedding into Kt.S4 provides no proof-theoretic insights into BiInt itself. Moreover, the restart technique of Horrocks et al. [9] involves non-deterministic expansion of disjunctions, which is complicated by inverse roles. 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 formulae, neither of which have a semantic content. Our purely syntactic approach is preferable for proof-theoretic reasons, since models are never explicitly involved in the proof system: see Remark 2.

In Section 2, we define the syntax and semantics of BiInt. In Section 3, we introduce our sequent calculus **GBiInt** and give an example derivation of Uustalu's interaction formula. We prove the soundness and completeness of **GBiInt** in Sections 4 and 5 respectively. A version with full proofs can be found in [2].

2 Syntax and Semantics of BiInt

The formulae *Fml* of BiInt are built from a denumerable set of *Atoms* and the constants $\top$ and $\bot$ using the connectives $\wedge$, $\vee$, $\to$, $-\!\!<$, $\neg$, and $\sim$. The length of a formula χ is just the number of symbols it contains. We use classical first-order

$$
\begin{array}{lll}
w \vDash \varphi \vee \psi & \text{if} & w \vDash \varphi \text{ or } w \vDash \psi \\
w \vDash \varphi \wedge \psi & \text{if} & w \vDash \varphi \ \ \& \ \ w \vDash \psi \\
w \vDash \neg\varphi & \text{if} & \forall u \in \mathcal{W}.[w\mathcal{R}u \Rightarrow (u \nvDash \varphi)] \\
w \vDash \varphi \rightarrow \psi & \text{if} & \forall u \in \mathcal{W}.[w\mathcal{R}u \Rightarrow (u \nvDash \varphi \text{ or } u \vDash \psi)] \\
w \vDash \sim \varphi & \text{if} & \exists u \in \mathcal{W}.[u\mathcal{R}w \ \ \& \ \ u \nvDash \varphi] \\
w \vDash \varphi -\!\!< \psi & \text{if} & \exists u \in \mathcal{W}.[u\mathcal{R}w \ \ \& \ \ u \vDash \varphi \ \ \& \ \ u \nvDash \psi]
\end{array}
$$

Fig. 1. `BiInt` semantics

logic when reasoning about `BiInt` at the meta-level. A `BiInt` frame is a pair $\langle \mathcal{W}, \mathcal{R} \rangle$, where $\mathcal{W}$ is a non-empty set of worlds and $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$ is a binary reflexive transitive relation. A `BiInt` model is a triple $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, where $\langle \mathcal{W}, \mathcal{R} \rangle$ is a `BiInt` frame and the truth valuation ϑ is a function $\mathcal{W} \times \mathit{Atoms} \rightarrow \{\text{true}, \text{false}\}$ which obeys: $\forall w \in \mathcal{W}.\vartheta(w, \top) = \text{true}$; $\forall w \in \mathcal{W}.\vartheta(w, \bot) = \text{false}$; and which obeys persistence, also known as truth monotonicity:

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

Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, a world $w \in \mathcal{W}$ and an atom $p \in \mathit{Atoms}$, we write $w \vDash p$ if $\vartheta(w, p) = \text{true}$. We pronounce $\vDash$ as "forces", and we pronounce $\nvDash$ as "rejects". The forcing of compound formulae is defined in Fig. 1. Since $\neg$ and $\sim$ can be derived from $\rightarrow$ and $-\!\!<$ respectively, we restrict our attention to $\rightarrow$, $-\!\!<$, $\wedge$, $\vee$. We obtain persistence for compound formulae by induction on their length, and then reverse persistence for compound formulae follows from persistence because the truth valuation is binary:

$$\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle.\forall u, w \in \mathcal{W}.\forall \varphi \in \mathit{Fml}.(w \vDash \varphi \ \ \& \ \ w\mathcal{R}u \Rightarrow u \vDash \varphi)$$
$$\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle.\forall u, w \in \mathcal{W}.\forall \varphi \in \mathit{Fml}.(w \nvDash \varphi \ \ \& \ \ u\mathcal{R}w \Rightarrow u \nvDash \varphi).$$

We write ϵ to mean the empty set. Given a formula φ and two sets of formulae Δ and Γ, we write Δ, Γ for $\Delta \cup \Gamma$ and we write Δ, φ for $\Delta \cup \{\varphi\}$. Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and a world $w \in \mathcal{W}$, we write $w \vDash \Gamma$ (w forces Γ) if $\forall \varphi \in \Gamma.w \vDash \varphi$, and we write $w \rightleftharpoons \Delta$ (w rejects Δ) if $\forall \varphi \in \Delta.w \nvDash \varphi$. We deliberately use "$\rightleftharpoons$" for rejection of sets to emphasize that every member of the set is rejected, instead of "$\nvDash$", which could be seen as "some member is rejected".

$\Gamma \Vdash_{\texttt{BiInt}} \Delta$ means: $\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle.\forall w \in \mathcal{W}.(w \vDash \Gamma \Rightarrow \exists \varphi \in \Delta.w \vDash \varphi)$
$\Gamma \nVdash_{\texttt{BiInt}} \Delta$ means: $\exists \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle.\exists w \in \mathcal{W}.(w \vDash \Gamma \ \ \& \ \ w \rightleftharpoons \Delta)$.

Thus $\Gamma \nVdash_{\texttt{BiInt}} \Delta$ means that $\Gamma \Vdash_{\texttt{BiInt}} \Delta$ is falsifiable. As usual, our sequent calculus has a semantic reading which assumes that there exists an initial world w_0 in a `BiInt`-model $\mathcal{M}$ where $w_0 \vDash \Gamma$ and $w_0 \rightleftharpoons \Delta$. We then systematically apply the sequent rules using backward proof-search to either construct $\mathcal{M}$ successfully, giving us $\Gamma \nVdash_{\texttt{BiInt}} \Delta$, or conclude that $\mathcal{M}$ cannot exist, giving us $\Gamma \Vdash_{\texttt{BiInt}} \Delta$.

3 Our Sequent Calculus GBiInt

We now present **GBiInt**, a Gentzen-style sequent calculus for `BiInt`. The sequents have a non-traditional component in the form of variables that are instan-

tiated at the leaves of the derivation tree, and passed back to lower sequents from *premises to conclusion*. Note that variables are not names for Kripke worlds.

We extend our syntax for presenting some of our sequent rules. The extended `BiInt` formulae are defined as: if φ is a `BiInt` formula, then φ is an extended `BiInt` formula, and if $\mathcal{S}/\mathcal{P}$ is a set $\{\{\varphi_0^0, \cdots, \varphi_0^n\}, \cdots, \{\varphi_m^0, \cdots, \varphi_m^k\}\}$ of sets of `BiInt` formulae, then $\bigvee \mathcal{S}$ and $\bigwedge \mathcal{P}$ are extended `BiInt` formulae with intended semantics

$$\bigvee \mathcal{S} \equiv (\varphi_0^0 \wedge \cdots \wedge \varphi_0^n) \vee \cdots \vee (\varphi_m^0 \wedge \cdots \wedge \varphi_m^k)$$
$$\bigwedge \mathcal{P} \equiv (\varphi_0^0 \vee \cdots \vee \varphi_0^n) \wedge \cdots \wedge (\varphi_m^0 \vee \cdots \vee \varphi_m^k).$$

From now on, we implicitly treat extended `BiInt` formulae as their `BiInt` equivalents. Given a `BiInt` model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, and a world $w \in \mathcal{W}$, the following semantics follows directly from their definition:

$$w \vDash \bigvee \mathcal{S} \text{ if } \exists \Gamma \in \mathcal{S}. w \vDash \Gamma \quad \text{and} \quad w \nvDash \bigwedge \mathcal{P} \text{ if } \exists \Delta \in \mathcal{P}. w \nvDash \Delta.$$

We can now extend the definition of forcing and rejecting to extended `BiInt` formulae in the obvious way. If Γ and Δ are sets of extended `BiInt` formulae, and φ is an extended `BiInt` formula, then $w \vDash \Gamma$ if $\forall \varphi \in \Gamma. w \vDash \varphi$, and $w \nvDash \Delta$ if $\forall \varphi \in \Delta. w \nvDash \varphi$.

A **GBiInt** sequent is an expression $\frac{\mathcal{S}}{\mathcal{P}} \big|\big| \Gamma \vdash \Delta$, where the left hand side (LHS) Γ is a set of extended `BiInt` formulae; the right hand side (RHS) Δ is a set of extended `BiInt` formulae; and the variables $\mathcal{S}$ and $\mathcal{P}$ are each a set of sets of formulae. We sometimes write just $\Gamma \vdash \Delta$, ignoring the variable values for readability, but only when the values of the variables are not important to the discussion. In terms of the counter-model under construction, we say that a sequent $\frac{\mathcal{S}}{\mathcal{P}} \big|\big| \Gamma \vdash \Delta$ is **falsifiable** [at w_0 in $\mathcal{M}$] iff there exists a `BiInt` model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and $\exists w_0 \in \mathcal{W}$ such that $w_0 \vDash \Gamma$ and $w_0 \nvDash \Delta$. Thus, a sequent $\Gamma \vdash \Delta$ is not falsifiable iff $\Gamma \Vdash_{\texttt{BiInt}} \Delta$. We say the **variable conditions** of a sequent $\gamma = \frac{\mathcal{S}}{\mathcal{P}} \big|\big| \Gamma \vdash \Delta$ hold iff γ is falsifiable at w_0 in some model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and the following **S**uccessor/**P**redecessor conditions hold:

$\mathcal{S}$-condition: $\exists \Sigma \in \mathcal{S}. \forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \vDash \Sigma$
$\mathcal{P}$-condition: $\exists \Pi \in \mathcal{P}. \forall w \in \mathcal{W}. w \mathcal{R} w_0 \Rightarrow w \nvDash \Pi.$

A sequent **rule** is an expression of one of the two forms below

$$(name)\ \frac{\gamma_1 \ \cdots \ \gamma_n}{\gamma} \qquad \text{side conditions}$$

$$(name)\ \frac{\gamma_0 \ \cdots \ \gamma_n}{\gamma} \qquad \text{side conditions}$$

where $n \geq 0$, and each γ_i is a sequent. The rule has a name, a **conclusion** γ, optional **premise(s)** $\gamma_1, \cdots, \gamma_n$, optional **side conditions**, and universal branching as indicated by a solid line or existential branching as indicated by a dashed line (explained shortly).

Our **traditional** rules (Fig. 2) are based on Dragalin's `GHPC` [5] for `Int` because we require multiple formulae in the succedents and antecedents of sequents for completeness; we have added symmetric rules for the `DualInt` connective $-\!\!<$. The main difference is that our $(\rightarrow_L)$ rule and the symmetric $(-\!\!<_R)$ carry their

$$(Id)\ \frac{}{\substack{\mathcal{S}:=\epsilon\\ \mathcal{P}:=\epsilon}\Big|\Big|\ \Gamma, \varphi \vdash \Delta, \varphi} \qquad (\bot_L)\ \frac{}{\substack{\mathcal{S}:=\epsilon\\ \mathcal{P}:=\epsilon}\Big|\Big|\ \Gamma, \bot \vdash \Delta} \qquad (\top_R)\ \frac{}{\substack{\mathcal{S}:=\epsilon\\ \mathcal{P}:=\epsilon}\Big|\Big|\ \Gamma \vdash \Delta, \top}$$

$$(\wedge_L)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi \wedge \psi, \varphi, \psi \vdash \Delta}{\substack{\mathcal{S}:=\mathcal{S}_1\\ \mathcal{P}:=\mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi \wedge \psi \vdash \Delta} \qquad (\wedge_R)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \wedge \psi, \varphi \qquad \substack{\mathcal{S}_2\\ \mathcal{P}_2}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \wedge \psi, \psi}{\substack{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2\\ \mathcal{P}:=\mathcal{P}_1 \cup \mathcal{P}_2}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \wedge \psi}$$

$$(\vee_R)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \vee \psi, \varphi, \psi}{\substack{\mathcal{S}:=\mathcal{S}_1\\ \mathcal{P}:=\mathcal{P}_1}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \vee \psi} \qquad (\vee_L)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi \vee \psi, \varphi \vdash \Delta \qquad \substack{\mathcal{S}_2\\ \mathcal{P}_2}\Big|\Big|\ \Gamma, \varphi \vee \psi, \psi \vdash \Delta}{\substack{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2\\ \mathcal{P}:=\mathcal{P}_1 \cup \mathcal{P}_2}\Big|\Big|\ \Gamma, \varphi \vee \psi \vdash \Delta}$$

$$(\rightarrow_L)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi \rightarrow \psi \vdash \varphi, \Delta \qquad \substack{\mathcal{S}_2\\ \mathcal{P}_2}\Big|\Big|\ \Gamma, \varphi \rightarrow \psi, \psi \vdash \Delta}{\substack{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2\\ \mathcal{P}:=\mathcal{P}_1 \cup \mathcal{P}_2}\Big|\Big|\ \Gamma, \varphi \rightarrow \psi \vdash \Delta}$$

$$(-\!\!<_R)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \psi \vdash \Delta, \varphi -\!\!< \psi \qquad \substack{\mathcal{S}_2\\ \mathcal{P}_2}\Big|\Big|\ \Gamma \vdash \Delta, \varphi -\!\!< \psi, \varphi}{\substack{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2\\ \mathcal{P}:=\mathcal{P}_1 \cup \mathcal{P}_2}\Big|\Big|\ \Gamma \vdash \Delta, \varphi -\!\!< \psi}$$

For every rule with premises π_i and conlusion γ, apply the rule only if:
$\forall \pi_i.(LHS_{\pi_i} \not\subseteq LHS_\gamma$ or $RHS_{\pi_i} \not\subseteq RHS_\gamma)$

Fig. 2. GBiInt rules - traditional

principal formula and all side formulae into the premises. Our rules for $\wedge$ and $\vee$ also carry their principal formula into their premises to assist with termination. Note that there are other approaches to a terminating sequent calculus for `Int`, e.g., Dyckhoff's contraction-free calculi [6], or history methods by Heuerding et al. [8] and Howe [10]. 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 loop-checking method than with history-based methods since both [8] and [10] prove completeness using syntactic transformations of derivations. Consequently, while **GBiInt** is sound and complete for the `Int` (and `DualInt`) fragment of `BiInt`, it is unlikely to be as efficient on the fragment as these specific calculi.

Our rules for $\rightarrow$ on the right and $-\!\!<$ on the left (Fig. 3) are **non-traditional**. The $(\rightarrow_R)$ and $(-\!\!<_L)$ rules have *two* premises instead of one, and they are connected by **existential branching** as indicated by the dotted horizontal line. Existential branching means that the conclusion is derivable if *some* premise is derivable; thus it is dual to the conventional universal branching, where the conclusion is derivable if *all* premises are derivable. We chose existential branching rather than two separate non-invertible rules so the left premise can communicate information via variables to the right premise. This inter-premise communication and the use of variables is crucial to proving interaction formulae of `BiInt`, and it gives our calculus an **operational reading**.

When applying an existential branching rule during backward proof search, we first create the left premise. If the left premise is non-derivable, then it returns

$$(Ret)\ \frac{}{\substack{\mathcal{S}:=\{\Gamma\}\\ \mathcal{P}:=\{\Delta\}}\Big|\Big|\ \Gamma \vdash \Delta}$$

where no other rule is applicable

$$(\rightarrow^I_R)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \rightarrow \psi, \psi}{\substack{\mathcal{S}:=\mathcal{S}_1\\ \mathcal{P}:=\mathcal{P}_1}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \rightarrow \psi} \qquad (-<^I_L)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi, \varphi -< \psi \vdash \Delta}{\substack{\mathcal{S}:=\mathcal{S}_1\\ \mathcal{P}:=\mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi -< \psi \vdash \Delta}$$

$$(\rightarrow_R)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \varphi \vdash \psi \qquad \substack{\mathcal{S}_2\\ \mathcal{P}_2}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \rightarrow \psi, \bigwedge \mathcal{P}_1}{\mathcal{S}/\mathcal{P}:=\begin{cases}\mathcal{S}_1/\mathcal{P}_1 & \text{if } \mathcal{P}_1 = \epsilon\\ \mathcal{S}_2/\mathcal{P}_2 & \text{if right prem created}\\ \{\Gamma\}/\{\Delta, \varphi \rightarrow \psi\} & \text{otherwise}\end{cases}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \rightarrow \psi}$$

right prem created only if $\mathcal{P}_1 \neq \epsilon\ \ \&\ \ \forall \Pi_i \in \mathcal{P}_1.\Pi_i \not\subseteq \{\Delta, \varphi \rightarrow \psi\}$

$$(-<_L)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \varphi \vdash \Delta, \psi \qquad \substack{\mathcal{S}_2\\ \mathcal{P}_2}\Big|\Big|\ \Gamma, \varphi -< \psi, \bigvee \mathcal{S}_1 \vdash \Delta}{\mathcal{S}/\mathcal{P}:=\begin{cases}\mathcal{S}_1/\mathcal{P}_1 & \text{if } \mathcal{S}_1 = \epsilon\\ \mathcal{S}_2/\mathcal{P}_2 & \text{if right prem created}\\ \{\Gamma, \varphi -< \psi\}/\{\Delta\} & \text{otherwise}\end{cases}\Big|\Big|\ \Gamma, \varphi -< \psi \vdash \Delta}$$

right prem created only if $\mathcal{S}_1 \neq \epsilon\ \ \&\ \ \forall \Sigma_i \in \mathcal{S}_1.\Sigma_i \not\subseteq \{\Gamma, \varphi -< \psi\}$

$$(\bigwedge_R)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma \vdash \Delta, \Pi_1 \ \cdots\ \substack{\mathcal{S}_n\\ \mathcal{P}_n}\Big|\Big|\ \Gamma \vdash \Delta, \Pi_n}{\substack{\mathcal{S}:=\bigcup_1^n \mathcal{S}_i\\ \mathcal{P}:=\bigcup_1^n \mathcal{P}_i}\Big|\Big|\ \Gamma \vdash \Delta, \bigwedge \Pi} \qquad (\bigvee_L)\ \frac{\substack{\mathcal{S}_1\\ \mathcal{P}_1}\Big|\Big|\ \Gamma, \Sigma_1 \vdash \Delta \ \cdots\ \substack{\mathcal{S}_n\\ \mathcal{P}_n}\Big|\Big|\ \Gamma, \Sigma_n \vdash \Delta}{\substack{\mathcal{S}:=\bigcup_1^n \mathcal{S}_i\\ \mathcal{P}:=\bigcup_1^n \mathcal{P}_i}\Big|\Big|\ \Gamma, \bigvee \Sigma \vdash \Delta}$$

For every universally branching rule with premises π_i and conlusion γ, apply the rule only if: $\forall \pi_i.(LHS_{\pi_i} \not\subseteq LHS_\gamma$ or $RHS_{\pi_i} \not\subseteq RHS_\gamma)$

For every existentially branching rule with left premise π and conlusion γ, apply the rule only if: $LHS_\pi \not\subseteq LHS_\gamma$ or $RHS_\pi \not\subseteq RHS_\gamma$

Fig. 3. GBiInt rules - non-traditional

the variables $\mathcal{S}_1$ and $\mathcal{P}_1$. We then use these variables to create the right premise, which corresponds to the same world as the conclusion, but with updated information. Our existential branching rules work together with (Ret), which assigns the variables at non-derivable leaves of failed derivation trees, and $(\bigwedge_R)$ and $(\bigvee_L)$, which extract the different variable choices at existential branching rules.

The conclusion of each of our rules **assigns the variables** based on the variables returned from the premise(s), and 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 $(\wedge_L)$ in Fig. 2 assigns $\mathcal{S} := \mathcal{S}_1$, where $\mathcal{S}_1$ is the value of the variable at the premise. However, for rules with multiple universally branching premises, we take a union of the sets of sets corresponding to each falsifiable premise. For example, the conclusion of $(\bigwedge_R)$ in Fig. 3 assigns $\mathcal{S} := \bigcup_1^n \mathcal{S}_i$, where $\mathcal{S}_i$ is the value of the variable at the i-th premise.

This way, the sets of sets stored in our variables **determinise** the return of formulae to lower sequents – each non-derivable 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 Example 2 in [2]. Then the intuition behind adding $\bigwedge \mathcal{P}$ to the right premise of $(\rightarrow_R)$ is that the subsequent application of $(\bigwedge_R)$ will create one or more premises, depending on the cardinality of $\mathcal{P}$. Since $\mathcal{P}$ is a set of sets representing all the open branches, all of the premises of $(\bigwedge_R)$ have to be derivable in order to obtain a derivation. On the other hand, if some premises of $(\bigwedge_R)$ are non-derivable (open), we form the set that consists of the union of the variables returned by those premises, and pass the union back to lower sequents, and so on. The premises that are derivable contribute only ϵ and are thus ignored by the union operator. Also, we only create the right premise of $(\rightarrow_R)$ if **every** member of $\mathcal{P}$ introduces new formulae to the current world. Otherwise, the current world already contains one of the open branches, which would still remain open after an application of $(\bigwedge_R)$. To summarise, the sets-of-sets concept of variables is critical to the soundness of **GBiInt**, as it allows us to remember the required choices arising further up the tree.

The **extended syntax** allows us to syntactically encode the variable choices described above. While the variables $\mathcal{S}$ and $\mathcal{P}$ are sets of sets when we pass them down the tree and combine them using set union, we use $\bigvee \mathcal{S}$ on the left and $\bigwedge \mathcal{P}$ on the right of the sequent to reflect these choices when we add $\bigvee \mathcal{S}$ or $\bigwedge \mathcal{P}$ to the right premise of an existentially branching rule. Then the $(\bigvee_L)$ and $(\bigwedge_R)$ rules break down the extended formulae $\bigvee \mathcal{S}$ and $\bigwedge \mathcal{P}$ to yield several premises, each corresponding to one variable choice. Thus the extended syntax allows us to give an intuitive syntactic representation of the variable choices.

We have also added the rule $(\rightarrow^I_R)$ for implication on the right (and dually, $(-\!\!<^I_L)$) originally given by Švejdar [14]. Rather than immediately creating the successor for a rejected $\varphi \rightarrow \psi$, the $(\rightarrow^I_R)$ rule first pre-emptively adds ψ to the right hand side of the sequent. Although Švejdar himself does not give the semantics behind this rule, and is unable to explain the precise role it plays in his calculus, it is very useful in our termination proof. 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, for if w forces ψ, then by forward persistence so does v, thus giving a contradiction.

The **side condition** on each of our rules is a general **blocking** condition, where we only explore the premise(s), if they are different from the conclusion. For example, in the $(\wedge_R)$ case, the blocking condition means that we apply the rule in backward proof search only if $\varphi \notin \Delta$ and $\psi \notin \Delta$, since otherwise some premise would be equal to the conclusion.

GBiInt also has the **subformula property**. This is obvious for all rules, except $(\rightarrow_R)$ and the dual $(-\!\!<_L)$. For these, the right premise "constructs" the formulae $\bigwedge \mathcal{P}$ and $\bigvee \mathcal{S}$. However, since $\mathcal{P}$ and $\mathcal{S}$ are sets of sets of subformulae of the conclusion that are again extracted by $(\bigwedge_R)$ and $(\bigvee_L)$, the right premise of $(\rightarrow_R)$ and $(-\!\!<_L)$ effectively only contains subformulae of the conclusion.

A **GBiInt** tree for $\frac{\mathcal{S}}{\mathcal{P}} \big|\big| \Gamma \vdash \Delta$ is a tree rooted at $\frac{\mathcal{S}}{\mathcal{P}} \big|\big| \Gamma \vdash \Delta$ where each child is obtained by a backwards application of a **GBiInt** rule and each leaf is an instance of $(\bot_L)$, $(\top_R)$, (Id) or (Ret).

Definition 1. *A* **GBiInt** *tree for* $\gamma = \frac{\mathcal{S}}{\mathcal{P}} \big|\big| \Gamma \vdash \Delta$ *is a derivation if:* γ *is the conclusion of a* $(\bot_L)$, $(\top_R)$ *or* (Id) *rule application; OR* γ *is the conclusion of a* ***universal branching*** *rule application and* ***all*** *its* ***premises*** *are derivations; OR* γ *is the conclusion of an* ***existential branching*** *rule application and* ***some premise*** *is a derivation.*

In the following example, we use a simplified version of the $(\wedge_R)$ rule, which discards the principal formula from the premises, merely to save horizontal space. Also, we only show non-empty variable values.

Example 1. The following is a derivation tree of Uustalu's interaction formula $p \to (q \vee (r \to ((p \multimap q) \wedge r))$, simplified to the sequent $p \vdash q, r \to ((p \multimap q) \wedge r)$. Let $X := r \to ((p \multimap q) \wedge r)$. The tree should be read bottom-up while ignoring the variables $\mathcal{S}$ and $\mathcal{P}$. At the leaves, the variables are assigned and transmit information down to parents and across to some siblings. The top left application of (Ret) occurs because an application of $(\multimap_R)$ to the bolded $p \multimap q$ is blocked, since its left premise would not be different from its conclusion. The key to finding the contradiction is the bolded $\mathbf{p} \multimap \mathbf{q}$ formula that is passed from the left-most leaf back to the right premise (1) of $(\to_R)$. The $(\bigwedge_R)$ in (1) is unary here, since the $\mathcal{P}$ variable contains only one set of formulae.

$$
\dfrac{
\dfrac{
\dfrac{
(Ret)\ \dfrac{}{\substack{\mathcal{S}:=\{\{p,r,q\}\} \\ \mathcal{P}:=\{\{\mathbf{p}\multimap\mathbf{q}\}\}} \big|\big| p, r, q \vdash \mathbf{p} \multimap \mathbf{q}}
\qquad
(Id)\ \dfrac{}{p, r \vdash p \multimap q, p}
}{(\multimap_R)\ \dfrac{\substack{\mathcal{S}:=\{\{p,r,q\}\} \\ \mathcal{P}:=\{\{\mathbf{p}\multimap\mathbf{q}\}\}} \big|\big| p, r \vdash p \multimap q}{}}
\qquad
(Id)\ \dfrac{}{p, r \vdash r}
}{(\wedge_R)\ \dfrac{}{\substack{\mathcal{S}:=\{\{p,r,q\}\} \\ \mathcal{P}:=\{\{\mathbf{p}\multimap\mathbf{q}\}\}} \big|\big| p, r \vdash (p \multimap q) \wedge r}}
}{(\to_R)\ \dfrac{}{p \vdash q, r \to ((p \multimap q) \wedge r)}}
\qquad (1)
$$

where (1) is:

$$
\dfrac{
\dfrac{
(Id)\ \dfrac{}{p, \mathbf{q} \vdash q, X, p \multimap q}
\qquad
(Id)\ \dfrac{}{p \vdash q, X, p \multimap q, \mathbf{p}}
}{(\multimap_R)\ \dfrac{}{p \vdash q, X, \mathbf{p} \multimap \mathbf{q}}}
}{(\bigwedge_R)\ \dfrac{}{p \vdash q, X, \bigwedge\{\{\mathbf{p} \multimap \mathbf{q}\}\}}}
$$

We now show that proof search in **GBiInt** terminates because our soundness proof relies on the left premises of existentially branching rules to deliver variables to their right premises. The (Ret) rule is an *operational rule*. All other rules are *logical rules* and are categorised as follows: the *static rules* (Id), $(\bot_L)$, $(\top_R)$, $(\wedge_L)$, $(\vee_L)$, $(\wedge_R)$, $(\vee_R)$, $(\to_L)$, $(\multimap_R)$, $(\to_R^I)$ add formulae to the current world in the counter-model; the *transitional rules* $(\to_R)$, $(\multimap_L)$ create new worlds and

Function Prove
Input: sequent γ_0
Output: Derivable ($true$ or $false$)

1. If $\rho \in \{(Id), (\bot_L), (\top_R)\}$ is applicable to γ_0 then return $true$
2. Else if any special or static rule ρ is applicable to γ_0 then
   (a) Let $\gamma_1, \cdots, \gamma_n$ be the (universally branching) premises of ρ
   (b) Return $\bigwedge_{i=1}^{n} Prove(\gamma_i)$
3. Else for each transitional rule ρ applicable to γ_0 do
   (a) Let γ_1 and γ_2 be the (existentially branching) premises of ρ
   (b) If $\bigvee_{i\in\{1,2\}} Prove(\gamma_i) = true$ then return $true$
4. Endif
5. Return $false$.

Fig. 4. Proof search strategy. Note that we have left out the variables for simplicity. $\bigwedge_{i=1}^{n} Prove(\gamma_i)$ is true iff $Prove(\gamma_i)$ is true for all premises γ_i for $1 \leq i \leq n$, and $\bigvee_{i\in\{1,2\}} Prove(\gamma_i)$ is true iff $Prove(\gamma_i)$ is true for some premise γ_i for $i \in \{1, 2\}$.

add formulae to them; and the *special rules* $(\bigvee_L)$, $(\bigwedge_R)$ decompose variables returned from non-derivable leaves. This classification justifies our backward proof search strategy (Fig. 4).

For a `BiInt` formula φ, the subformulae $sf(\varphi)$ are defined as usual with $\varphi \in sf(\varphi)$. For extended `BiInt` formulae $\bigvee \mathcal{S}$, $\bigwedge \mathcal{P}$, and a set Γ of extended `BiInt` formulae let:

$$sf(\bigvee \mathcal{S}) = \bigcup_{\Sigma \in \mathcal{S}} sf(\Sigma) \qquad sf(\bigwedge \mathcal{P}) = \bigcup_{\Pi \in \mathcal{P}} sf(\Pi) \qquad sf(\Gamma) = \bigcup_{\chi \in \Gamma} sf(\chi).$$

Note that the subformulae of $\bigvee \mathcal{S}$ and $\bigwedge \mathcal{P}$ do not include the conjunctions and disjunctions implicit in their `BiInt` equivalents.

Given a **GBiInt**-tree $\mathcal{T}$ and a branch $\mathcal{B}$ in $\mathcal{T}$, we say that $\mathcal{B}$ is **forward-only** if $\mathcal{B}$ contains only applications of static and special rules, $(\rightarrow_R)$ and the right premises of $(-\!\!<_L)$. Similarly, $\mathcal{B}$ is **backward-only** if $\mathcal{B}$ contains only applications of static and special rules, $(-\!\!<_L)$ and the right premises of $(\rightarrow_R)$. A branch is **single-directional** if it is either forward-only or backward-only. A branch contains **interleaved** left premises of transitional rules if it contains a sequence $\langle \cdots, \gamma_i, \cdots, \gamma_j, \cdots, \gamma_k, \cdots \rangle$ s.t. γ_i is the left premise of $(\rightarrow_R)$, γ_j is the left premise of $(-\!\!<_L)$, and γ_k is the left premise of $(\rightarrow_R)$.

Lemma 1. *Every forward/backward only branch of any* **GBiInt***-tree is finite.*

Proof. We prove the lemma for forward-only branches, the one for backward-only branches is similar. Let $>_{len}$ be a lexicographic ordering of sequents: $(\Gamma_2 \vdash \Delta_2) >_{len} (\Gamma_1 \vdash \Delta_1)$ iff $|\Gamma_2| > |\Gamma_1|$, or ($|\Gamma_2| = |\Gamma_1|$ and $|\Delta_2| > |\Delta_1|$). Then from the blocking conditions of the rules, the length of a sequent according to $>_{len}$ increases on every forward-only branch: see [2] for details. Since **GBiInt** has the subformula property, eventually no more formulae can be added to a sequent on a forward-only branch, and the branch will terminate.

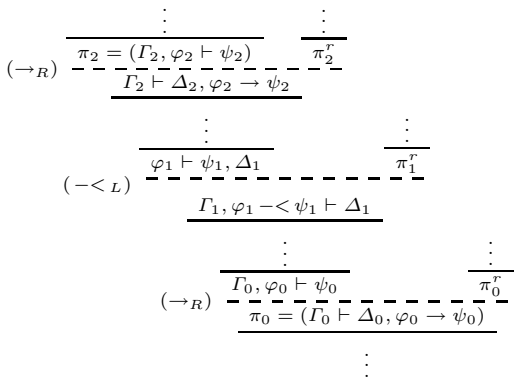

Fig. 5. Interleaved left premises of transitional rules.

Lemma 2. *If a* **GBiInt***-tree has an infinite branch, then the branch has an infinite number of interleaved left premises of transitional rules.*

Proof. By Lemma 1, single-directional branches must terminate. Thus, an infinite branch must involve an infinite number of interleaved left premises of transitional rules.

The degree of a `BiInt` formula φ is the number of $\rightarrow$ and $-\!\!<$ connectives in φ. The degree of a sequent $\Gamma \vdash \Delta$ is: $deg(\Gamma \vdash \Delta) = \sum_{\varphi \in sf(\Gamma \cup \Delta)} deg(\varphi)$. The following corollaries directly follow from the definition of the degree of a sequent.

Corollary 1. *By the subformula property of* **GBiInt***, the degree of a sequent can never increase in backward proof search.*

Corollary 2. *Given two sequents γ_1 and γ_2, if $sf(\gamma_2) \subsetneq sf(\gamma_1)$, then $deg(\gamma_2) < deg(\gamma_1)$. That is, removing some formula φ from a sequent during backward proof search decreases the sequent degree if φ is not a subformula of any other formula in the sequent.*

Theorem 1 (Termination). *Every* **GBiInt***-tree constructed according to the strategy of Fig. 4 is finite.*

Proof. Suppose for a contradiction that there exists an infinite **GBiInt**-tree $\mathcal{T}$. Since every rule has a finite number of premises, then by König's lemma $\mathcal{T}$ contains a branch $\mathcal{B}$ of infinite length. By Lemma 2, $\mathcal{B}$ contains an infinite number of interleaved left premises of transitional rules as shown in Fig. 5.

Let $\chi \in sf(\pi_0)$ be 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. Thus χ is not a subformula of any formula with a larger degree. We show that $\chi \notin sf(\pi_2)$. There are two cases:

$\chi \notin sf(\Gamma_0)$: Then $\chi \in sf(\Delta_0)$ or $\chi = \varphi_0 \rightarrow \psi_0$. In both cases, $\chi \notin sf(\pi_2)$.

$\chi \in sf(\Gamma_0)$**:** Then it cannot be the case that $\chi \in sf(\varphi_1)$ or $\chi \in sf(\psi_1)$, since then $deg(\varphi_1 -\!\!< \psi_1) > deg(\chi)$, contradicting $deg(\chi) = max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$. Therefore, either:

- χ and all its occurrences in subformulae disappear from the sequent at the premise of $(-\!\!<_L)$, in which case $\chi \notin sf(\pi_2)$, or
- χ is moved to the RHS by applying $(\rightarrow_L)$ to some $\chi \rightarrow \tau$. But then $deg(\chi \rightarrow \tau) > deg(\chi)$, contradicting $deg(\chi) = max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$.

Thus we have $\chi \in sf(\pi_0)$ and $\chi \notin sf(\pi_2)$. Also, the subformula property of **GBiInt** gives $sf(\pi_2) \subseteq sf(\pi_0)$. Together with $\chi \in sf(\pi_0)$ and $\chi \notin sf(\pi_2)$, this means $sf(\pi_2) \subsetneq sf(\pi_0)$. Then by Corollary 2 we have $deg(\pi_2) < deg(\pi_0)$.

Note that the steps indicated by vertical ellipses ($\vdots$) in Fig. 5 are arbitrary, since by Corollary 1 no rule can increase the degree of a sequent. Since we have $deg(\pi_2) < deg(\pi_0)$, then every sequence of interleaved transitional rule applications must decrease the degree of the sequent. This can only happen a finite number of times, until no more transitional rules are applicable. Therefore our assumption was wrong, and no branch can be infinite. Hence every **GBiInt**-tree is finite.

4 Soundness

Instead of showing that each rule application preserves validity downwards, we show that each rule application preserves falsifiability upwards. Since variables introduce a two-way flow of information in the **GBiInt** trees, we separate the notion of soundness into two: *local soundness*, applicable to a single rule application, and *global soundness*, which considers the propagation of variables down the tree. Note that locality here refers to locality in the **GBiInt** trees, not locality in the underlying Kripke models.

Definition 2. *A logical rule in* **GBiInt** *is locally sound iff: if the conclusion is falsifiable, then some* ***universally*** *branching premise is falsifiable, or all* ***existentially*** *branching premises are falsifiable.*

Lemma 3. *Each static and special rule of* **GBiInt** *is locally sound.*

Proof. We assume the conclusion is falsifiable at w_0 in $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and easily show that some premise is falsifiable at w_0 in $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$: see [2] for details.

We now show global soundness, which relies on the notion of variable conditions because of the operational nature of **GBiInt**. Since our transitional rules use the variables returned from proof search of the left premise to instantiate the right premise, we need to show that the variables are instantiated and propagated soundly.

Lemma 4. *In any* **GBiInt** *tree $\mathcal{T}$, for every sequent $\gamma_0 \in \mathcal{T}$: if γ_0 is falsifiable, then some universally branching, or all existentially branching, premises are falsifiable, and the variable conditions hold at γ_0.*

Proof. By induction on the length $h(\gamma_0)$ of the longest branch rooted at γ_0.

Base case: $h(\gamma_0) = 0$. So γ_0 is an instance of (Id), $(\bot_L)$, $(\top_R)$, or (Ret).

(Id), $(\bot_L)$, $(\top_R)$**:** The conclusion is never falsifiable, so there is nothing to show.

(Ret)**:** We show that the conclusion $\Gamma \vdash \Delta$ is always falsifiable, and that the variable conditions hold at $\Gamma \vdash \Delta$. We create a model with a single world w_0, and for every atom $p \in \Gamma$, let $\vartheta(w_0, p) = true$, and for every atom $q \in \Delta$, let $\vartheta(w_0, q) = false$. An atom cannot be both in Γ and Δ, since (Id) is not applicable to $\Gamma \vdash \Delta$.

To show that $\Gamma \vdash \Delta$ is falsifiable at w_0, we need to show that $w_0 \vDash \Gamma$ and $w_0 \nvDash \Delta$. For every atom in Γ and Δ, the valuation ensures this. For every composite formula φ, we do a simple induction on its length. Since (Ret) is only applied when no other rules are applicable, the required subformula ψ is already in Γ or Δ as appropriate, and ψ falls under the induction hypothesis. Thus we know that: (i) $w_0 \vDash \Gamma$ and (ii) $w_0 \nvDash \Delta$. Then (i) and the persistence property of `BiInt` give us that $\forall w \in \mathcal{W} : w_0 \mathcal{R} w \Rightarrow w \vDash \Gamma$. Similarly, (ii) and the reverse persistence property of `BiInt` give us that $\forall w \in \mathcal{W}. w \mathcal{R} w_0 \Rightarrow w \nvDash \Delta$. That is, the variable conditions hold at the conclusion of the (Ret) rule.

Induction step: We assume that the lemma holds for all γ_0 with $h(\gamma_0) \leq k$, and show that it holds for all γ_0 with $h(\gamma_0) \leq k + 1$. Consider the rule application ρ such that γ_0 is the conclusion of ρ. By the assumption of the lemma, the conclusion γ_0 of ρ is falsifiable at some w_0 in some model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$. If ρ is a static or a special rule (universally branching), then the lemma easily follows from the induction hypothesis. Otherwise, ρ is a transitional rule (existentially branching). We show the case for $(\rightarrow_R)$, the case for $(-\!\!\prec_L)$ is symmetric:

$$(\rightarrow_R)\ \frac{\begin{array}{c}\mathcal{S}_1 \\ \mathcal{P}_1\end{array}\Big|\Big|\ \Gamma, \varphi \vdash \psi \qquad \begin{array}{c}\mathcal{S}_2 \\ \mathcal{P}_2\end{array}\Big|\Big|\ \Gamma \vdash \Delta, \varphi \to \psi, \bigwedge \mathcal{P}_1}{\mathcal{S}/\mathcal{P} := \begin{cases} \mathcal{S}_1/\mathcal{P}_1 & \text{if } \mathcal{P}_1 = \epsilon \\ \mathcal{S}_2/\mathcal{P}_2 & \text{if right prem created} \\ \{\Gamma\}/\{\Delta, \varphi \to \psi\} & \text{otherwise} \end{cases} \Big|\Big|\ \Gamma \vdash \Delta, \varphi \to \psi}$$

$$\text{right prem created only if } \mathcal{P}_1 \neq \epsilon \ \&\ \forall \Pi_i \in \mathcal{P}_1. \Pi_i \not\subseteq \{\Delta, \varphi \to \psi\}$$

Since the conclusion is falsifiable, there is a world w_0 such that (i) $w_0 \vDash \Gamma$ and, (ii) $w_0 \nvDash \Delta, \varphi \to \psi$. From the `BiInt`-semantics of $\rightarrow$, (ii) implies that there is a successor w_1 such that: (iii) $w_0 \mathcal{R} w_1$, (iv) $w_1 \vDash \varphi$ and (v) $w_1 \nvDash \psi$.

1. To show that the left premise γ_1 of $(\rightarrow_R)$ is falsifiable, we need to show that there exists a world w' s. t. γ_1 is falsifiable at w'. We let $w' = w_1$. Then (i), (iv) and (v) give us that γ_1 is falsifiable.

   Now, γ_1 has $height(\gamma_1) \leq k$, therefore the induction hypothesis applies to γ_1. By the induction hypothesis, since γ_1 is falsifiable at w_1, we have that the variable conditions hold at γ_1. In particular, the $\mathcal{P}$ condition holds, giving:

   $$\exists \Pi \in \mathcal{P}_1. \forall w \in \mathcal{W}. w \mathcal{R} w_1 \Rightarrow w \nvDash \Pi \tag{4.1}$$

   Now there are two cases: either the right premise γ_2 was created, or it was not (and there is nothing to show). If it was created, then we need

to show that it is falsifiable by exhibiting a world w'' such that γ_2 is falsifiable at w''. We let $w'' = w_0$. Then, since $w_0 \mathcal{R} w_1$, we have $w_0 \nVdash \Pi$ by (4.1). Since $\Pi \in \mathcal{P}_1$, then by the semantics of extended formulae, we have that $w_0 \nVdash \bigwedge \mathcal{P}_1$. Together with (i) and (ii), this means that γ_2 is falsifiable at w_0. Moreover, the variable conditions hold at γ_2, since it also is falsifiable, and has $height(\gamma_2) \leq k$, so the induction hypothesis applies to it.

2. To show that the variable conditions hold at the conclusion γ_0, we do the case for $\mathcal{S}$; the case for $\mathcal{P}$ is dual. We need to show:

$$\exists \Sigma \in \mathcal{S}. \forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \vDash \Sigma \tag{4.2}$$

Since we have shown that the variable conditions hold at the left premise, we know that in particular $\mathcal{P}_1 \neq \epsilon$. Then there are two cases: either the right premise was created, or it was not:

- If the right premise γ_2 was created, then its variable conditions hold, since γ_2 falls under the induction hypothesis. This gives us:

$$\exists \Sigma_2 \in \mathcal{S}_2. \forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \vDash \Sigma_2.$$

  Thus $\mathcal{S} := \mathcal{S}_2$ obeys (4.2).
- If the right premise was not created, then we need to show that the variable conditions hold at the conclusion for $S := \{\Gamma\}$. Now, we have $w_0 \vDash \Gamma$ by (i), and then the persistence property tells us that $\forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \vDash \Gamma$. Thus $\mathcal{S} := \{\Gamma\}$ obeys (4.2).

Theorem 2 (Soundness). *If $\Gamma \vdash \Delta$ is derivable then $\Gamma \nVdash_{\texttt{BiInt}} \Delta$.*

Proof. We assume that $\Gamma \vdash \Delta$ is derivable and prove $\Gamma \vdash \Delta$ is not falsifiable. Then $\Gamma \nVdash_{\texttt{BiInt}} \Delta$ follows by definition. By induction on the height k of the derivation. **Base case:** A derivation of height 1 can only be an instance of $(\bot_L)$, $(\top_R)$ or (Id). In each case, γ is not falsifiable. **Inductive step:** We assume that if there is a derivation for γ of height $\leq k$, then γ is not falsifiable. Using Definition 1 and Lemma 4, it is easy to show by contradiction that if γ has a derivation of height $\leq k+1$, then γ is not falsifiable.

5 Completeness

We prove completeness via model graphs following [7]. We say that $\Gamma \vdash \Delta$ is **consistent** if $\bot \notin \Gamma$, $\top \notin \Delta$ and $\Gamma \cap \Delta = \epsilon$. We say that $\Gamma \vdash \Delta$ is **closed** w.r.t. a **GBiInt** rule ρ if either ρ is not applicable to $\Gamma \vdash \Delta$, or whenever $\Gamma \vdash \Delta$ matches the conclusion of an instance of ρ, then for some premise $\Gamma_1 \vdash \Delta_1$, we have $\Gamma_1 \subseteq \Gamma$ and $\Delta_1 \subseteq \Delta$. We say that $\Gamma \vdash \Delta$ is **saturated** if it is consistent and closed w.r.t. the static rules of **GBiInt**.

Corollary 3. *If $\mathcal{S} \,\big|\big|\, \mathcal{P}$ $\Gamma \vdash \Delta$ is not derivable, then $\Gamma \vdash \Delta$ is consistent for all $\mathcal{S}$ and $\mathcal{P}$.*

Remark 1. As usual, every sequent has a set of one or more "saturations" due to the branching of $(\wedge_R)$, $(\vee_L)$, etc., rules. The usual approach is to non-deterministically choose one of the non-derivable premises of each such rule. However, in the presence of the inverse relation, a branch that appears open may close once we return variables to a lower sequent. Therefore, we need to temporarily keep all the non-derivable premises, since we do not know which of the open branches will stay open when we return to a lower sequent.

Lemma 5. *For each finite non-derivable sequent $\Gamma \vdash \Delta$, there is an effective procedure to construct a finite set $\zeta = \{\alpha_1, \cdots, \alpha_n\}$ of finite saturated sequents, with $\Gamma \cup \Delta \subseteq LHS(\alpha_j) \cup RHS(\alpha_j) \subseteq sf(\Gamma) \cup sf(\Delta)$ for all $1 \leq j \leq n$.*

Proof. Let $\mathcal{T} = \Gamma \vdash \Delta$. Repeatedly apply static rules to the leaves of $\mathcal{T}$ to obtain new leaves. Keep the non-derivable leaves only; by Corollary 3 they are consistent. By Theorem 1, the saturation process will terminate; let $\zeta = \{\alpha_1, \cdots, \alpha_n\}$ be the final leaves of $\mathcal{T}$. By the subformula property, $LHS(\alpha_j) \cup RHS(\alpha_j) \subseteq sf(\Gamma) \cup sf(\Delta)$ for all $1 \leq j \leq n$.

Definition 3. *A model graph for a sequent $\Gamma \vdash \Delta$ is a finite* `BiInt` *frame $\langle \mathcal{W}, \mathcal{R} \rangle$ such that all $w \in \mathcal{W}$ are saturated sequents $\Gamma_w \vdash \Delta_w$ and:*

1. *$\Gamma \subseteq \Gamma_{w_0}$ and $\Delta \subseteq \Delta_{w_0}$ for some $w_0 \in \mathcal{W}$, where $w_0 = \Gamma_{w_0} \vdash \Delta_{w_0}$;*
2. *if $\varphi \to \psi \in \Delta_w$ then $\exists v \in \mathcal{W}$ with $w\mathcal{R}v$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$;*
3. *if $\varphi -\!\!< \psi \in \Gamma_w$ then $\exists v \in \mathcal{W}$ with $v\mathcal{R}w$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$;*
4. *if $w\mathcal{R}v$ and $\varphi \to \psi \in \Gamma_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$;*
5. *if $v\mathcal{R}w$ and $\varphi -\!\!< \psi \in \Delta_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_{w'}$;*
6. *if $w\mathcal{R}v$ and $\varphi \in \Gamma_w$ then $\varphi \in \Gamma_v$;*
7. *if $v\mathcal{R}w$ and $\varphi \in \Delta_w$ then $\varphi \in \Delta_v$.*

Lemma 6. *If there exists a model graph $\langle \mathcal{W}, \mathcal{R} \rangle$ for $\Gamma \vdash \Delta$, then there exists a* `BiInt` *model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ such that for some $w_0 \in \mathcal{W}$, we have $w_0 \Vdash \Gamma$ and $w_0 \nVdash \Delta$. We call $\mathcal{M}$ the counter-model for $\Gamma \nvdash_{\texttt{BiInt}} \Delta$.*

Proof. Follows from Definition 3 by induction on the length of $\Gamma \vdash \Delta$.

We now show how to construct a model graph for $\Gamma \vdash \Delta$ from a consistent $\Gamma \vdash \Delta$. Recall from Remark 1 that we need to keep a number of independent versions of worlds because of the choices arising due to disjunctive non-determinism. We do this by storing one or more independent connected-components $\langle \mathcal{W}_1, \mathcal{R}_1 \rangle, \cdots, \langle \mathcal{W}_n, \mathcal{R}_n \rangle$ in the constructed model graph $\langle \mathcal{W}, \mathcal{R} \rangle$, and the indices (sorts) of worlds and relations tell us the connected-component of the graph to which they belong. We write $\langle \mathcal{W}_j, \mathcal{R}_j \rangle[j := i]$ to relabel the connected component $\langle \mathcal{W}_j, \mathcal{R}_j \rangle$ with sort j to a connected component $\langle \mathcal{W}_i, \mathcal{R}_i \rangle$ with sort i. Similarly, we also label each member of the variables $\mathcal{P}$ and $\mathcal{S}$, so we can later extract the member with sort i, corresponding to the component of $\langle \mathcal{W}, \mathcal{R} \rangle$ with sort i. We write $\mathcal{R}$-neighbour to mean $\mathcal{R}$-predecessor or $\mathcal{R}$-successor.

Our algorithm in Fig. 6 starts by saturating the root world to obtain one or more saturated "states". For each "state" α_i, it recursively creates all the

Procedure MGC
Input: sequent $\Gamma \vdash \Delta$
Output: model graph $\langle \mathcal{W}^f, \mathcal{R}^f \rangle$, variables $\mathcal{S}^f$ and $\mathcal{P}^f$

1. Let $\zeta = \{\alpha_1, \cdots, \alpha_n\}$ be the result of saturating $\Gamma \vdash \Delta$ using Lemma 5;
2. For each $\alpha_i \in \zeta$ do
   (a) Let $\langle \mathcal{W}_i, \mathcal{R}_i \rangle = \langle \{\alpha_i\}, \{(\alpha_i, \alpha_i)\} \rangle$; let $recompute := false$;
   (b) For each non-blocked $\varphi \to \psi \in \Delta_{\alpha_i}$ and while $recompute = false$ do
      i. Apply $(\to_R)$ to $\varphi \to \psi$ and obtain a left premise $\pi_1 = \Gamma_{\alpha_i}, \varphi \vdash \psi$;
      ii. Let $\langle \mathcal{W}, \mathcal{R} \rangle, \mathcal{S}, \mathcal{P} := MGC(\pi_1)$;
      iii. If $\exists \Pi_j \in \mathcal{P}. \Pi_j \subseteq \Delta_{\alpha_i}$ then
         A. Let $u_j \in \mathcal{W}_j$ be the root of the connected component $\mathcal{W}_j$ from $\mathcal{W}$;
         B. Let $G = \langle \mathcal{W}_j, \mathcal{R}_j \rangle[j := i]$; add G to $\langle \mathcal{W}_i, \mathcal{R}_i \rangle$, and put $\alpha_i \mathcal{R}_i u_i$.
      iv. else
         A. Let $\langle \mathcal{W}_i, \mathcal{R}_i \rangle = \langle \epsilon, \epsilon \rangle$; let $recompute := true$;
         B. Invoke the right premise of $(\to_R)$ to obtain $\pi_2 = \Gamma_{\alpha_i} \vdash \Delta_{\alpha_i}, \bigwedge \mathcal{P}$;
         C. Apply $(\bigwedge_R)$ to π_2 to obtain $m \geq 1$ non-derivable premises $\gamma_1, \cdots, \gamma_m$;
         D. For each γ_k, $1 \leq k \leq m$, let $\langle \mathcal{W}_k, \mathcal{R}_k \rangle, \mathcal{S}_k, \mathcal{P}_k := MGC(\gamma_k)$;
         E. Let $\langle \mathcal{W}_i, \mathcal{R}_i \rangle := \langle \bigcup \mathcal{W}_k, \bigcup \mathcal{R}_k \rangle$, and $\mathcal{S}_i := \bigcup \mathcal{S}_{\gamma_k}$ and $\mathcal{P}_i := \bigcup \mathcal{P}_{\gamma_k}$;
   (c) For each non-blocked $\varphi -< \psi \in \Gamma_{\alpha_i}$ and while $recompute = false$ do
      i. Perform a symmetric procedure to Steps 2(b)i to 2(b)ivE.
   (d) If $recompute = false$ then let $\mathcal{S}_i := \{\Gamma_{\alpha_i}\}$ and $\mathcal{P}_i := \{\Delta_{\alpha_i}\}$.
3. Return $\langle \bigcup \mathcal{W}_i, \bigcup \mathcal{R}_i \rangle, \bigcup \mathcal{S}_i, \bigcup \mathcal{P}_i$

Fig. 6. Model Graph Construction Procedure

$\mathcal{R}$-neighbours and saturates them, and so on. If during the construction of any $\mathcal{R}$-neighbour, new information is returned from the higher sequents (Step 2(b)iv), then we delete the entire subtree (connected component of sort i) rooted at α_i, and recreate α_i using the new information (Step 2(b)ivB). This re-creates all the $\mathcal{R}$-neighbours of α_i. Otherwise, if none of the $\mathcal{R}$-neighbours of α_i return any new information, or there are no $\mathcal{R}$-neighbours for α_i, then Step 2d instantiates the variables and returns from the recursion. In the latter case, the "state" α_i already has all the required information it can possibly receive from any $\mathcal{R}$-neighbours, thus α_i is final. Note the duality: new information *from* a single $\mathcal{R}$-neighbour means that all of the members of a variable were new, while new information *at* a "state" α_i means that some $\mathcal{R}$-neighbour returned new information.

When we return from MGC, we form the union of the components of the model graph and the variables from the different "states", so that the caller of MGC can extract the appropriate component at Step 2(b)iiiA.

Remark 2. Note that while the counter-model construction procedure keeps the whole counter-model in memory, this procedure is only used to prove the completeness of **GBiInt**. Our procedure for checking the validity of `BiInt` formulae (Fig. 4) does not need the whole counter-model, and explores one branch at a time, as is usual for sequent/tableaux calculi.

Theorem 3 (Completeness). GBiInt *is complete: if* $\Gamma \vdash \Delta$ *is not derivable, then there exists a counter-model for* $\Gamma \Vdash_{\texttt{BiInt}} \Delta$.

Proof. If $\Gamma \vdash \Delta$ is not derivable, then $\Gamma \vdash \Delta$ is consistent by Corollary 3. We construct a model graph for $\Gamma \vdash \Delta$ using the procedure of Fig. 6, and obtain $\langle \mathcal{W}^f, \mathcal{R}^f \rangle$. Let $\langle \mathcal{W}, \mathcal{R} \rangle$ be any connected component of $\langle \mathcal{W}^f, \mathcal{R}^f \rangle$. To show that $\langle \mathcal{W}, \mathcal{R} \rangle$ satisfies the properties of a model graph, we give the cases for properties 1, 2 and 4, the others are similar:

1. $\Gamma \subseteq \Gamma_{w_0}$ and $\Delta \subseteq \Delta_{w_0}$ for some $w_0 \in \mathcal{W}$: This holds because w_0 is one of the saturated sequents obtained from $\Gamma \vdash \Delta$. Moreover, if we delete the original w_0 at Step 2(b)ivA, a final version of w_0 is created at Step 2(b)iiiB which is never deleted.
2. if $\varphi \to \psi \in \Delta_w$ then $\exists v \in \mathcal{W}$ with $w\mathcal{R}v$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$: This holds because we have either created v using $(\to_R)$ at Step 2(b)iiiB, or had w fulfill the role of this successor by reflexivity if $(\to_R)$ was blocked.
4. if $w\mathcal{R}v$ and $\varphi \to \psi \in \Gamma_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$: In our construction, there are three ways of obtaining $w\mathcal{R}v$, so we need to show that for each case, the property holds. We first show that $\varphi \to \psi \in \Gamma_v$:
   1. v was created by applying $(\to_R)$ to w on some $\alpha \to \beta \in \Delta_w$. Then Γ_v also contains $\varphi \to \psi$.
   2. w was created by applying $(-\!\!< _L)$ to some $\alpha -\!\!< \beta \in \Gamma_v$. Then, when the final version of Γ_v was created, $\varphi \to \psi \in \Gamma_w$ was added to the $\mathcal{S}$ variable at Step 2d. There are two cases:
      - The right premise π_2 of $(-\!\!< _L)$ was invoked at v. Then $\mathcal{S}$ was added to π_2 at v by the symmetric process to Step 2(b)ivB. Thus the updated Γ_v also contains $\varphi \to \psi$.
      - The right premise of $(-\!\!< _L)$ was not invoked at v. This means that $\exists \Sigma_j \in \mathcal{S}. \Sigma_j \subseteq \Gamma_v$, and the j-th version of v's predecessor w is chosen at the symmetric process to Step 2(b)iiiA. But since Step 2d at w assigns $\Sigma_j := \Gamma_w$, then we have $\Gamma_w \subseteq \Gamma_v$ and thus $\varphi \to \psi \in \Gamma_v$.
   3. $v = w$, and $w\mathcal{R}w$ by reflexivity. Then $\Gamma_v = \Gamma_w$, so $\varphi \to \psi \in \Gamma_v$.

   In all cases, saturation for v will then ensure that $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$.

We can obtain a counter-model for $\Gamma \Vdash_{\texttt{BiInt}} \Delta$ from $\langle \mathcal{W}, \mathcal{R} \rangle$ via Lemma 6.

We use di-tree to mean a directed graph such that if the direction of the edges is ignored, it is a tree. The following corollary follows directly from our procedure since it never creates proper clusters: see [2].

Corollary 4. `BiInt` *is characterised by finite rooted reflexive and transitive di-trees.*

6 Conclusions and Future Work

Our cut-free calculus for `BiInt` enjoys terminating backward proof-search and is sound and complete w.r.t Kripke semantics. A simple Java implementation

of **GBiInt** is available at `http://users.rsise.anu.edu.au/~linda`. The next step is to add a cut rule to **GBiInt**, and prove cut elimination syntactically. We are also extending our work to the modal logic `S5`, and the tense logic `Kt.S4`. Our approach of existential branching and inter-premise communication bears some similarities to hypersequents of Pottinger and Avron [1]. It would be interesting to investigate this correspondence further. From an automated deduction perspective, **GBiInt** is the first step towards an efficient decision procedure for `BiInt`. The next task is to analyse the computational complexity of **GBiInt** and investigate which of the traditional optimisations for tableaux systems are still applicable in the intuitionistic case.

We would like to thank the anonymous reviewers for their suggestions.

References

1. A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In *Proc. Logic Colloquium, Keele, UK, 1993*, 1–32, OUP, 1996.
2. L. Buisman and R. Goré. A cut-free sequent calculus for bi-intuitionistic logic: extended version. `http://arxiv.org/abs/0704.1707`, 2007.
3. T. Crolard. Subtractive logic. *Theor. Comp. Sci.*, 254(1–2):151–185, March 2001.
4. J. Czermak. A remark on Gentzen's calculus of sequents. *NDJFL*, 18(3), 1977.
5. A. Dragalin. *Mathematical Intuitionism: Introduction to Proof Theory*, volume 68 of *Translations of Mathematical Monographs*. Cambridge Univ. Press, 1988.
6. R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, September 1992.
7. R. Goré. Tableau methods for modal and temporal logics. In D'Agostino at al, editor, *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.
8. A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. TABLEAUX'96, 1996.
9. I. Horrocks, U. Sattler, and S. Tobies. A PSpace-algorithm for deciding $ALCNI_{R^+}$-satisfiability. LTCS-98-08, LuFG Theor. Comp. Sci, RWTH Aachen, 1998.
10. J. M. Howe. *Proof search issues in some non-classical logics*. PhD thesis, University of St Andrews, 1998.
11. C. Rauszer. A formalization of the propositional calculus of H-B logic. *Studia Logica*, 33:23–34, 1974.
12. C. Rauszer. An algebraic and Kripke-style approach to a certain extension of intuitionistic logic. *Dissertationes Mathematicae*, 168, 1980. Inst. of Math, Polish Academy of Sciences.
13. S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Proc. TABLEAUX'98*, pages 277–292, 1998.
14. V. Švejdar. On sequent calculi for intuitionistic propositional logic. *Commentationes Mathematicae Universitatis Carolinae*, 47(1):159–173, 2006.
15. M. E. Szabo, ed. *The Collected Papers of Gerhard Gentzen*. Studies in Logic and the foundations of Mathematics. North-Holland, Amsterdam, 1969.
16. I. Urbas. Dual-intuitionistic logic. *NDJFL*, 37(3):440–451, Summer 1996.
17. T. Uustalu. Personal communication. via email, 2004.
18. T. Uustalu. Personal communication. via email, 2006.
19. T. Uustalu and L. Pinto. Days in logic '06 conference abstract. At `http://www.mat.uc.pt/~kahle/dl06/tarmo-uustalu.pdf`, accessed on 27th October 2006.
20. F. Wolter. On logics with coimplication. *JPL*, 27(4):353–387, 1998.