A Theorem Prover for Boolean BI

Jonghyun Park Jeongbong Seo Sungwoo Park

July 2012
POSTECH-CSE-2012-7

Department of Computer Science & Engineering
Pohang University of Science and Technology
Republic of Korea, 790-784

Abstract

While separation logic is acknowledged as an enabling technology for large-scale program verification, most of the existing verification tools use only a fragment of separation logic that excludes separating implication. As the first step towards a verification tool using full separation logic, we develop a nested sequent calculus for Boolean BI (Bunched Implications), the underlying theory of separation logic, as well as a theorem prover based on it. A salient feature of our nested sequent calculus is that its sequent may have not only smaller child sequents but also multiple parent sequents, thus producing a graph structure of sequents instead of a tree structure. Our theorem prover is based on backward search in a refinement of the nested sequent calculus in which weakening and contraction are built into all the inference rules. We explain the details of designing our theorem prover and provide empirical evidence of its practicality.

Keywords: Separation logic, Boolean BI, Cut elimination, Sequent calculus, Nested sequent

1 Introduction

1.1 Separation logic

Separation logic [41] is an extension of Hoare logic which facilitates reasoning about programs using mutable data structures. As it is acknowledged as an enabling technology for large-scale program verification [44, 8, 33], researchers have developed automated verification tools that use separation logic as their foundational theory. Examples of such tools include Smallfoot [6], Space Invader [16, 13], THOR [32], SLAyer [4], HIP [38], VeriFast [28], jStar [17], and Xisa [14]. The active development of such tools attests to the importance of local reasoning in program verification, which is precisely the key feature that separation logic intends to support.

All the aforementioned tools, however, use not full separation logic but only a decidable fragment by Berdine *et al*. [5] or its extension. Specifically separation logic features two new logical connectives, separating conjunction $\star$ and separating implication $-\!\!\star$, but this decidable fragment includes only separating conjunction. Lack of separating implication implies that for any program performing heap mutation or allocation, there is no support for backward reasoning by weakest precondition generation, an essential requirement for any complete program verification system (see Ishtiaq and O'Hearn [27]). Thus, while very effective in their respective application domains, these tools allow only forward reasoning based on symbolic execution as in [7] and fail to demonstrate the full potential of separation logic in program verification.

Omitting separating implication is not a deliberate decision. Rather it is an inevitable decision due to the availability of no theorem prover for full separation logic. Berdine *et al.* [7] suggest that such a theorem prover is highly desirable and can evolve into a complete program verification system based on separation logic:

> *This incompleteness could be dealt with if we instead used the backwards-running weakest preconditions of Separation Logic. Unfortunately, there is no existing automatic theorem prover which can deal with the form of these assertions (which use quantification and the separating implication $-\!\!\star$). If there were such a prover, we would be eager consumers of it.*

Still, however, there is no practical theorem prover for full separation logic.

Our long-term goal is to develop a theorem prover for full separation logic and incorporate it into a program verification system supporting backward reasoning. The first step is then to study Boolean BI, the underlying theory of separation logic.

1.2 Boolean BI

Boolean BI is a substructural logic which belongs to the family of the logic of BI (Bunched Implications) of O'Hearn and Pym [39]. It inherits additive connectives from classical propositional logic, thus retaining the convenience of classical reasoning. It is also particularly suitable for reasoning about local resources because of multiplicative connectives inherited from intuitionistic linear logic. Like other members in the family, Boolean BI allows free combinations of these additive connectives and multiplicative connectives, giving rise to an unusual form of contexts called bunches: trees whose internal nodes specify whether subtrees are combined additively or multiplicatively. We obtain separation logic as a model for Boolean BI based on a monoid of heaps.

While theoretical work on Boolean BI is maturing with recent discoveries of its undecidability [31, 11], there is still no practical theorem prover for Boolean BI. The display calculus for Boolean BI by Brotherston [10], which draws on the framework of display logic by Belnap [2], has the cut elimination property and thus can be easily turned into a theorem prover, but developing a practical proof search strategy on top of it is far from easy because of the complexity due to its display rules [9]. In order to develop a practical theorem prover for Boolean BI and hence also for full separation logic, then, we may need another proof theory that directly reflects the characteristics of Boolean BI and better lends itself to proof search. This paper presents such a proof theory for Boolean BI as well as a theorem prover based on it.

1.3 Contribution

We present a *nested sequent calculus* $\mathbf{S_{BBI}}$ for Boolean BI. Unlike in typical nested sequent calculi [29, 12, 24, 25, 26], its sequent may have not only smaller child sequents but also multiple parent sequents, thus producing a graph structure of sequents instead of a tree structure. The use of nested sequents is necessary because of the presence of intuitionistic multiplicative conjunction in a classical setting. The use of a graph structure of sequents is necessary because of the interaction between multiplicative implication and classical negation. As in typical multi-conclusioned sequent calculi for classical logic, we use multisets of formulas not only for antecedents but also for succedents of a sequent. Thus, although $\mathbf{S_{BBI}}$ belongs to the family of the logic of BI, its sequents do not use bunches, which are supplanted by new structural connectives specifying a graph structure of sequents. $\mathbf{S_{BBI}}$ has the cut elimination property and is sound and complete with respect to the Kripke semantics for Boolean BI.

Our theorem prover for Boolean BI is based on backward proof search in another nested sequent calculus $\mathbf{CS_{BBI}}$ which is obtained from $\mathbf{S_{BBI}}$ by building weakening and contraction into all the inference rules. In conjunction with a graph structure of sequents, the structural rules in $\mathbf{CS_{BBI}}$ make it particularly challenging to devise a practical proof search strategy, even if it is based on backward proof search. We deal with an explosion in the search space due to the structural rules, which can be applied indefinitely and exponentially increase the search space, by prioritizing their applications. We find that our theorem prover is reasonably fast in proving typical formulas of Boolean BI. To the best of our knowledge, our theorem prover is the first theorem prover for Boolean BI.

1.4 Organization of the paper

Section 2 gives preliminaries on Boolean BI. Section 3 presents the nested sequent calculus $\mathbf{S_{BBI}}$ and the satisfaction relation for its sequents. Section 4 proves the cut elimination property of $\mathbf{S_{BBI}}$, and Section 5 proves the soundness and completeness of $\mathbf{S_{BBI}}$ with respect to the satisfaction relation as well as the Kripke semantics for Boolean BI. Section 6 reviews the display calculus for Boolean BI by Brotherston [10] and shows that $\mathbf{S_{BBI}}$ is an optimization of the display calculus. Section 7 presents the nested sequent calculus $\mathbf{CS_{BBI}}$. Section 8 describes the backward search strategy in our theorem prover and presents experimental results. Section 9 discusses related work and Section 10 concludes. Appendix shows selected proofs. Our theorem prover is available at `http://pl.postech.ac.kr/BBI/`.

2 Preliminaries on Boolean BI

Formulas in Boolean BI extend classical propositional logic with multiplicative connectives from linear logic:

$$\text{formula} \quad A \quad ::= \quad P \mid \bot \mid \neg A \mid A \vee A \mid \mathsf{I} \mid A \star A \mid A \mathrel{-\!\!\star} A$$

P denotes an atomic formula drawn from a set V. I is the multiplicative unit. $A \star B$ is a multiplicative conjunction and $A \mathrel{-\!\!\star} B$ is a multiplicative implication. We define $\top$ as $\neg\bot$, $A \wedge B$ as $\neg(\neg A \vee \neg B)$, and $A \to B$ as $\neg A \vee B$. We use conventional precedence rules for logical connectives: $\neg > \wedge, \star > \vee > \to, \mathrel{-\!\!\star}$.

The Kripke semantics of Boolean BI [19] uses a non-deterministic commutative monoid on a set U. Assume a binary operator $\circ : U \times U \to \mathcal{P}(U)$ and a unit element $e \in U$ (where $\mathcal{P}(U)$ denotes the power set of U). We extend $\circ$ to a binary operator on $\mathcal{P}(U)$ such that $U_1 \circ U_2 = \cup\{w_1 \circ w_2 \mid w_1 \in U_1, w_2 \in U_2\}$. A non-deterministic commutative monoid is a triple $\langle U, \circ, e\rangle$ which satisfies the following conditions:

$$\begin{array}{rl} \text{(neutrality)} & \forall w \in U.\ w \circ e = \{w\} \\ \text{(commutativity)} & \forall w_1, w_2 \in U.\ w_1 \circ w_2 = w_2 \circ w_1 \\ \text{(associativity)} & \forall w_1, w_2, w_3 \in U.\ w_1 \circ (w_2 \circ w_3) = (w_1 \circ w_2) \circ w_3 \end{array}$$

Given a non-deterministic commutative monoid $\langle U, \circ, e\rangle$ and a valuation $\rho : V \to \mathcal{P}(U)$ of atomic formulas, we obtain the Kripke semantics of Boolean BI from the satisfaction relation $w, \rho \models A$ for formulas given in Figure 1. The satisfaction relation $w, \rho \models A$ is defined inductively on the structure of formula A. A formula A is valid, written $\models A$, if $w, \rho \models A$ holds for any element w and valuation ρ.

The Hilbert system for Boolean BI [40] uses a judgment $\vdash A$ and is obtained by extending classical propositional logic with axioms and inference rules for multiplicative connectives given in Figure 2. An

$$
\begin{array}{rll}
w, \rho \models P & \text{iff.} & w \in \rho(P) \\
w, \rho \models \bot & \text{iff.} & \text{never} \\
w, \rho \models \neg A & \text{iff.} & w, \rho \not\models A \\
w, \rho \models A \vee B & \text{iff.} & w, \rho \models A \text{ or } w, \rho \models B \\
w, \rho \models \mathsf{I} & \text{iff.} & w = e \\
w, \rho \models A \star B & \text{iff.} & \exists w_1, w_2 \in U \text{ such that } w \in w_1 \circ w_2 \text{ and } w_1, \rho \models A \text{ and } w_2, \rho \models B \\
w, \rho \models A \mathrel{-\!\!\star} B & \text{iff.} & \forall w_1 \in U.\ w_1, \rho \models A \text{ implies } \forall w_2 \in w \circ w_1.\ w_2, \rho \models B
\end{array}
$$

Figure 1: Satisfaction relation $w, \rho \models A$ for formulas

$$
\begin{array}{ll}
\text{(Axiom 1)} & \vdash A \to \mathsf{I} \star A \\
\text{(Axiom 2)} & \vdash \mathsf{I} \star A \to A \\
\text{(Axiom 3)} & \vdash A \star B \to B \star A \\
\text{(Axiom 4)} & \vdash A \star (B \star C) \to (A \star B) \star C
\end{array}
$$

$$
\frac{\vdash A_1 \to A_2 \quad \vdash B_1 \to B_2}{\vdash (A_1 \star B_1) \to (A_2 \star B_2)}\ \star H
\qquad
\frac{\vdash (A \star B) \to C}{\vdash A \to (B \mathrel{-\!\!\star} C)}\ \mathrel{-\!\!\star} H_1
\qquad
\frac{\vdash A \to (B \mathrel{-\!\!\star} C)}{\vdash (A \star B) \to C}\ \mathrel{-\!\!\star} H_2
$$

Figure 2: Axioms and inference rules for multiplicative connectives in the Hilbert system for Boolean BI

induction on the structure of the proof of $\vdash A$ proves the soundness of the Hilbert system with respect to the Kripke semantics of Boolean BI. Galmiche and Larchey-Wendling [19] prove that the Hilbert system is also complete with respect to the Kripke semantics of Boolean BI.

Theorem 2.1. *$\models A$ if and only if $\vdash A$.*

3 Nested sequent calculus $\mathbf{S_{BBI}}$ for Boolean BI

This section presents the nested sequent calculus $\mathbf{S_{BBI}}$ for Boolean BI. We first explain the definition of sequents in $\mathbf{S_{BBI}}$. Then we present the satisfaction relation for sequents and the inference rules of $\mathbf{S_{BBI}}$.

3.1 Nested sequents

A sequent in $\mathbf{S_{BBI}}$ represents a graph structure whose nodes store sequents in classical logic. A node can have multiple parent nodes as well as multiple child nodes, but the following two relations should always hold:

1. A node can have multiple parent nodes, but each parent node determines a unique sibling node. Hence no node can have two parent nodes with the same sibling node.

2. A node can have multiple child nodes, but each child node determines another unique child node. Hence we can divide all child nodes into groups of two sibling nodes.

Formally a sequent W describes a graph structure with respect to a certain node in it, which we refer to as the *reference node.* It consists of a truth context and a falsehood context. A truth context contains *node states* which are either true formulas specific to the reference node or descriptions of its relation to child, sibling, and parent nodes; a falsehood context contains false formulas specific to the reference node:

$$
\begin{array}{rrcl}
\text{sequent} & W & = & \Gamma \Longrightarrow \Delta \\
\text{truth context} & \Gamma & ::= & \cdot \mid \Gamma; S \\
\text{falsehood context} & \Delta & ::= & \cdot \mid \Delta; A \\
\text{node state} & S & ::= & A \mid \emptyset_{\mathsf{m}} \mid W, W \mid W\langle W \rangle
\end{array}
$$

We use truth contexts and falsehood contexts as unordered sets and do not use an additive zero (like $\emptyset_{\mathsf{a}}$ in the definition of bunches). $\emptyset_{\mathsf{m}}$ is a special node state which corresponds to the unit element of the

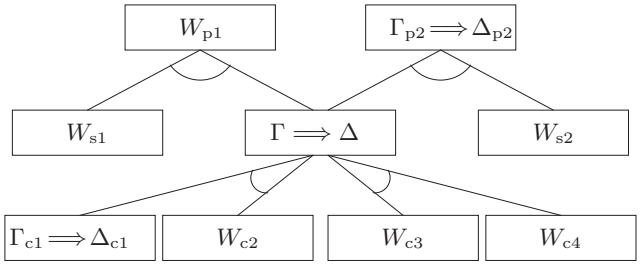


Figure 3: An example of a graph structure of nodes in $\mathbf{S_{BBI}}$ (p for parent, c for child, s for sibling)

monoid in the Kripke semantics of Boolean BI. A *multiplicative pair* W, W' asserts the existence of a pair of child nodes which are reference nodes of W and W'. An *adjoint pair* $W\langle W'\rangle$ asserts the existence of a sibling node and a common parent node which are reference nodes of W and W', respectively. Note that a sequent in $\mathbf{S_{BBI}}$ reverts to a sequent in classical logic if we leave only true formulas in its truth context.

The use of adjoint pairs implies that we can describe the same graph structure of nodes using different sequents by changing the reference node. As an example, consider the graph structure in Figure 3 where lines denote parent-child relations and arcs denote sibling relations. We let $\Gamma' = W_{\mathrm{s1}}\langle W_{\mathrm{p1}}\rangle; (W_{\mathrm{c3}}, W_{\mathrm{c4}})$. Then the following three sequents describe the same graph structure in Figure 3, but all use different reference nodes (top right, center, and bottom left):

$$\Gamma_{\mathrm{p2}}; (\Gamma; \Gamma'; (\Gamma_{\mathrm{c1}} \Longrightarrow \Delta_{\mathrm{c1}}, W_{\mathrm{c2}}) \Longrightarrow \Delta, W_{\mathrm{s2}}) \Longrightarrow \Delta_{\mathrm{p2}}$$
$$\Gamma; \Gamma'; (\Gamma_{\mathrm{c1}} \Longrightarrow \Delta_{\mathrm{c1}}, W_{\mathrm{c2}}); W_{\mathrm{s2}}\langle \Gamma_{\mathrm{p2}} \Longrightarrow \Delta_{\mathrm{p2}}\rangle \Longrightarrow \Delta$$
$$\Gamma_{\mathrm{c1}}; W_{\mathrm{c2}}\langle \Gamma; \Gamma'; W_{\mathrm{s2}}\langle \Gamma_{\mathrm{p2}} \Longrightarrow \Delta_{\mathrm{p2}}\rangle \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{\mathrm{c1}}$$

$\mathbf{S_{BBI}}$ provides two inference rules which convert a sequent into another equivalent sequent by changing the reference node.

Our definition of sequents in $\mathbf{S_{BBI}}$ embodies the principle of proof by contradiction from classical logic: a proof of a sequent means that its truth and falsehood contexts together lead to a logical contradiction. This departure from the standard interpretation of sequents for classical logic (in which the conjunction of antecedents implies the disjunction of succedents) is intentional, as the principle of proof by contradiction guides the development of both the satisfaction relation and the rules for $\mathbf{S_{BBI}}$.

3.2 Satisfaction relation for sequents

Given a non-deterministic commutative monoid $\langle U, \circ, e\rangle$ and a valuation $\rho : V \to \mathcal{P}(U)$ of atomic formulas, we can define the satisfaction relation $w, \rho \models_{\mathcal{W}} W$ for sequents. It uses another satisfaction relation $w, \rho \models_{\mathcal{S}} S$ for node states and the satisfaction relation $w, \rho \models A$ for formulas:

$$w, \rho \models_{\mathcal{W}} \Gamma \Longrightarrow \Delta \quad \text{iff.} \quad \begin{cases} \forall S \in \Gamma.\ w, \rho \models_{\mathcal{S}} S \\ \forall A \in \Delta.\ w, \rho \not\models A \end{cases}$$

The satisfaction relation $w, \rho \models_{\mathcal{S}} S$ for node states is defined as follows:

$$\begin{aligned}
w, \rho \models_{\mathcal{S}} A &\ \text{iff.}\ w, \rho \models A \\
w, \rho \models_{\mathcal{S}} \emptyset_{\mathsf{m}} &\ \text{iff.}\ w = e \\
w, \rho \models_{\mathcal{S}} W_1, W_2 &\ \text{iff.}\ \exists w_1, w_2 \in U \text{ such that } w \in w_1 \circ w_2 \text{ and } w_1, \rho \models_{\mathcal{W}} W_1 \text{ and } w_2, \rho \models_{\mathcal{W}} W_2 \\
w, \rho \models_{\mathcal{S}} W_1\langle W_2\rangle &\ \text{iff.}\ \exists w_1, w_2 \in U \text{ such that } w_2 \in w \circ w_1 \text{ and } w_1, \rho \models_{\mathcal{W}} W_1 \text{ and } w_2, \rho \models_{\mathcal{W}} W_2
\end{aligned}$$

Note that the satisfaction relation for multiplicative formulas can be rewritten in terms of the satisfaction relation for node states as follows:

- $w, \rho \models \mathsf{I}$ iff. $w, \rho \models_{\mathcal{S}} \emptyset_{\mathsf{m}}$.
- $w, \rho \models A \star B$ iff. $w, \rho \models_{\mathcal{S}} (A \Longrightarrow \cdot), (B \Longrightarrow \cdot)$.

Structural rules:

$$\frac{\Gamma \Longrightarrow \Delta}{\Gamma; S \Longrightarrow \Delta}\ WL_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; A}\ WR_{\mathcal{S}} \quad \frac{\Gamma; S; S \Longrightarrow \Delta}{\Gamma; S \Longrightarrow \Delta}\ CL_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta; A; A}{\Gamma \Longrightarrow \Delta; A}\ CR_{\mathcal{S}} \quad \frac{\Gamma; W', W \Longrightarrow \Delta}{\Gamma; W, W' \Longrightarrow \Delta}\ EC_{\mathcal{S}}$$

$$\frac{\Gamma; W_1, (W_2, W_3 \Longrightarrow \cdot) \Longrightarrow \Delta}{\Gamma; (W_1, W_2 \Longrightarrow \cdot), W_3 \Longrightarrow \Delta}\ EA_{\mathcal{S}}$$

$$\frac{\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta_1}{\Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2}\ \emptyset_{\mathsf{m}}U_{\mathcal{S}} \quad \frac{\Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2}{\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta_1}\ \emptyset_{\mathsf{m}}D_{\mathcal{S}}$$

Traverse rules (p for parent, c for child, s for sibling):

$$\frac{\Gamma_{\mathrm{c1}}; (\Gamma_{\mathrm{c2}} \Longrightarrow \Delta_{\mathrm{c2}})\langle \Gamma \Longrightarrow \Delta \rangle \Longrightarrow \Delta_{\mathrm{c1}}}{\Gamma; (\Gamma_{\mathrm{c1}} \Longrightarrow \Delta_{\mathrm{c1}}), (\Gamma_{\mathrm{c2}} \Longrightarrow \Delta_{\mathrm{c2}}) \Longrightarrow \Delta}\ TC_{\mathcal{S}} \quad \frac{\Gamma_{\mathrm{p}}; (\Gamma \Longrightarrow \Delta), (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}) \Longrightarrow \Delta_{\mathrm{p}}}{\Gamma; (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}})\langle \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \rangle \Longrightarrow \Delta}\ TP_{\mathcal{S}}$$

Logical rules:

$$\frac{}{A \Longrightarrow A}\ Init_{\mathcal{S}} \quad \frac{}{\bot \Longrightarrow \cdot}\ \bot L_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \bot}\ \bot R_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta; A}{\Gamma; \neg A \Longrightarrow \Delta}\ \neg L_{\mathcal{S}} \quad \frac{\Gamma; A \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \neg A}\ \neg R_{\mathcal{S}}$$

$$\frac{\Gamma_1; A \Longrightarrow \Delta_1 \quad \Gamma_2; B \Longrightarrow \Delta_2}{\Gamma_1; \Gamma_2; A \vee B \Longrightarrow \Delta_1; \Delta_2}\ \vee L_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta; A; B}{\Gamma \Longrightarrow \Delta; A \vee B}\ \vee R_{\mathcal{S}} \quad \frac{\Gamma; \emptyset_{\mathsf{m}} \Longrightarrow \Delta}{\Gamma; \mathsf{I} \Longrightarrow \Delta}\ \mathsf{I}L_{\mathcal{S}} \quad \frac{}{\emptyset_{\mathsf{m}} \Longrightarrow \mathsf{I}}\ \mathsf{I}R_{\mathcal{S}}$$

$$\frac{\Gamma; (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow \Delta}{\Gamma; A \star B \Longrightarrow \Delta}\ \star L_{\mathcal{S}} \quad \frac{\Gamma_1 \Longrightarrow \Delta_1; A \quad \Gamma_2 \Longrightarrow \Delta_2; B}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\ \star R_{\mathcal{S}}$$

$$\frac{\Gamma_1 \Longrightarrow \Delta_1; A \quad \Gamma_2; B \Longrightarrow \Delta_2}{(\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2 \rangle; A \mathrel{-\!\!\star} B \Longrightarrow \cdot}\ \mathrel{-\!\!\star} L_{\mathcal{S}} \quad \frac{\Gamma; (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow B \rangle \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; A \mathrel{-\!\!\star} B}\ \mathrel{-\!\!\star} R_{\mathcal{S}}$$

Figure 4: Nested sequent calculus $\mathbf{S_{BBI}}$ for Boolean BI

- $w, \rho \models A \mathrel{-\!\!\star} B$ iff. $w, \rho \not\models_{\mathcal{S}} (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow B \rangle$.

If $w, \rho \not\models_{\mathcal{W}} W$ holds for any element w and valuation ρ, we say that W is unsatisfiable and write $\not\models_{\mathcal{W}} W$.

3.3 Nested sequent calculus $\mathbf{S_{BBI}}$

Figure 4 shows the nested sequent calculus $\mathbf{S_{BBI}}$ for Boolean BI. The inference rules are divided into three groups: structural rules, traverse rules, and logical rules. We read every rule from the conclusion to the premise.

A structural rule makes a change to the sequent in the conclusion, but does not change the reference node. The rules $WL_{\mathcal{S}}$, $WR_{\mathcal{S}}$, $CL_{\mathcal{S}}$, and $CR_{\mathcal{S}}$ are weakening and contraction rules. The rules $EC_{\mathcal{S}}$ and $EA_{\mathcal{S}}$ rewrite a node state according to commutativity and associativity of sequents, respectively. Note that associativity of sequents does not use $(W_1, W_2), W_3$ and $W_1, (W_2, W_3)$, both of which are syntactically ill-formed. The rule $\emptyset_{\mathsf{m}}U_{\mathcal{S}}$ creates a new child node with a special form of sequent $\emptyset_{\mathsf{m}} \Longrightarrow \cdot$, which can be absorbed back into the parent node by the rule $\emptyset_{\mathsf{m}}D_{\mathcal{S}}$. Intuitively $\emptyset_{\mathsf{m}} \Longrightarrow \cdot$ describes an empty node whose sibling node can be identified with its parent node.

A traverse rule changes the reference node without changing parent-child or sibling relations between nodes. The rules $TC_{\mathcal{S}}$ and $TP_{\mathcal{S}}$ promote the left child node (corresponding to $\Gamma_{\mathrm{c1}} \Longrightarrow \Delta_{\mathrm{c1}}$) and the parent node (corresponding to $\Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}}$), respectively, as the new reference node. In conjunction with the rule $EC_{\mathcal{S}}$, the two traverse rules enable us to designate an arbitrary node as the reference node because every pair of nodes can be connected only via parent-child relations. The following example

shows how to promote the sibling node as the reference node:

$$\frac{\Gamma_{\mathrm{s}}; (\Gamma \Longrightarrow \Delta)\langle \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \rangle \Longrightarrow \Delta_{\mathrm{s}}}{\Gamma_{\mathrm{p}}; (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}), (\Gamma \Longrightarrow \Delta) \Longrightarrow \Delta_{\mathrm{p}}}\ TC_{\mathcal{S}} \qquad \frac{\Gamma_{\mathrm{p}}; (\Gamma \Longrightarrow \Delta), (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}) \Longrightarrow \Delta_{\mathrm{p}}}{\Gamma; (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}})\langle \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \rangle \Longrightarrow \Delta}\ EC_{\mathcal{S}}\ TP_{\mathcal{S}}$$

A logical rule focuses on a principal formula in the reference node. If the sequent already expresses a logical contradiction, it completes the proof without generating a premise. If we are to focus on a formula that is not in the reference node, we can always use the traverse rules to expose it in the reference node before applying a corresponding logical rule. All the rules from $Init_{\mathcal{S}}$ to $\vee R_{\mathcal{S}}$ are from classical propositional logic. The rule $Init_{\mathcal{S}}$ expresses the principle of proof by contradiction. The rules $\mathsf{I}L_{\mathcal{S}}$ and $\mathsf{I}R_{\mathcal{S}}$ use the fact that I is true only in an empty node which $\emptyset_{\mathsf{m}}$ describes. The rules $\star L_{\mathcal{S}}$ and $\star R_{\mathcal{S}}$ are based on the following interpretation of multiplicative conjunction $\star$:

$$\begin{array}{lll} w, \rho \models A \star B & \text{iff.} & \exists w_1, w_2 \in U \text{ such that } w \in w_1 \circ w_2 \text{ and } w_1, \rho \models A \text{ and } w_2, \rho \models B \\ w, \rho \not\models A \star B & \text{iff.} & \forall w_1, w_2 \in U.\ w \in w_1 \circ w_2 \text{ implies } w_1, \rho \not\models A \text{ or } w_2, \rho \not\models B \end{array}$$

The rule $\star L_{\mathcal{S}}$ creates ($\exists$) two fresh child nodes (corresponding to w_1 and w_2) where A and B are true, respectively, which explains why we need to use nested sequents. The rule $\star R_{\mathcal{S}}$ chooses ($\forall$) two existing child nodes (corresponding to w_1 and w_2) which are described by $\Gamma_1 \Longrightarrow \Delta_1$ and $\Gamma_2 \Longrightarrow \Delta_2$. The rules $\mathrel{-\!\!\star} L_{\mathcal{S}}$ and $\mathrel{-\!\!\star} R_{\mathcal{S}}$ are based on the following interpretation of multiplicative implication $\mathrel{-\!\!\star}$:

$$\begin{array}{lll} w, \rho \models A \mathrel{-\!\!\star} B & \text{iff.} & \forall w_1 \in U.\ w_1, \rho \models A \text{ implies } \forall w_2 \in w \circ w_1.\ w_2, \rho \models B \\ w, \rho \not\models A \mathrel{-\!\!\star} B & \text{iff.} & \exists w_1, w_2 \in U \text{ such that } w_2 \in w \circ w_1 \text{ and } w_1, \rho \models A \text{ and } w_2, \rho \not\models B \end{array}$$

The rule $\mathrel{-\!\!\star} L_{\mathcal{S}}$ chooses ($\forall$) existing sibling and parent nodes (corresponding to w_1 and w_2) which are described by $\Gamma_1 \Longrightarrow \Delta_1$ and $\Gamma_2 \Longrightarrow \Delta_2$. The rule $\mathrel{-\!\!\star} R_{\mathcal{S}}$ creates ($\exists$) a fresh sibling node (corresponding to w_1) where A is true and a fresh parent node (corresponding to w_2) where B is false, which explains why we need to allow multiple parent nodes.

In the presence of the traverse rules, we may replace adjoint pairs with multiplicative pairs in the rules $\mathrel{-\!\!\star} L_{\mathcal{S}}$ and $\mathrel{-\!\!\star} R_{\mathcal{S}}$, which are the only logical rules using adjoint pairs:

$$\frac{\Gamma_1 \Longrightarrow \Delta_1; A \quad \Gamma_2; B \Longrightarrow \Delta_2}{\Gamma_2; (\Gamma_1 \Longrightarrow \Delta_1), (A \mathrel{-\!\!\star} B \Longrightarrow \cdot) \Longrightarrow \Delta_2}\ \mathrel{-\!\!\star} L'_{\mathcal{S}} \qquad \frac{(\Gamma \Longrightarrow \Delta), (A \Longrightarrow \cdot) \Longrightarrow B}{\Gamma \Longrightarrow \Delta; A \mathrel{-\!\!\star} B}\ \mathrel{-\!\!\star} R'_{\mathcal{S}}$$

This, however, does not mean that the traverse rules can be discarded, in which case a node can never have multiple parent nodes and the cut elimination property fails.

Figure 5 shows an example of proving $A \to (A \star B) \vee (A \star \neg B)$ in $\mathbf{S_{BBI}}$. The formula means that every node can have an adjacent node in which either B or $\neg B$ is true. First we apply the rule $\emptyset_{\mathsf{m}} U_{\mathcal{S}}$ to create an empty node, described by $\emptyset_{\mathsf{m}} \Longrightarrow \cdot$, in which we later mix assumptions of B and $\neg B$ to produce a logical contradiction:

$$(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B$$

Then we extend the truth context of the empty node with a node state S describing the current relation with its sibling and parent nodes:

$$(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B$$

Here we promote the empty node as the reference node to generate S and apply the contraction rule $CL_{\mathcal{S}}$ to duplicate S. After isolating the sequent for the empty node and adding B to its falsehood context, we consume S in $\emptyset_{\mathsf{m}}; S \Longrightarrow B$ (by the rule $TP_{\mathcal{S}}$) to restore the previous relation between the empty node and its sibling and parent nodes:

$$(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow B) \Longrightarrow A \star B; A \star \neg B$$

Finally we add B to the truth context and produce a logical contradiction:

$$\emptyset_{\mathsf{m}}; B \Longrightarrow B$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,Init_{\mathcal{S}}{}^{8}}{(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot) \Longrightarrow A \star B}\,\star R_{\mathcal{S}}{}^{7}}{(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B}\,WR_{\mathcal{S}}{}^{6}}{(\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot), (A \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B}\,\left.\begin{array}{l} EC_{\mathcal{S}} \\ TP_{\mathcal{S}} \end{array}\right\}{}^{5}}{\dfrac{\emptyset_{\mathsf{m}}; S; S \Longrightarrow \cdot}{\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot}\,CL_{\mathcal{S}}{}^{4}}}{(\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (A \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B}\,\left.\begin{array}{l} TC_{\mathcal{S}} \\ EC_{\mathcal{S}} \end{array}\right\}{}^{3}}{(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B}\,\emptyset_{\mathsf{m}}U_{\mathcal{S}}{}^{2}}{A \Longrightarrow A \star B; A \star \neg B}}{A \Longrightarrow (A \star B) \vee (A \star \neg B)}\,\vee R_{\mathcal{S}}{}^{1}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,Init_{\mathcal{S}}{}^{12}}{(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow B) \Longrightarrow A \star \neg B}\,\star R_{\mathcal{S}}{}^{11}}{(A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow B) \Longrightarrow A \star B; A \star \neg B}\,WR_{\mathcal{S}}{}^{10}}{(\emptyset_{\mathsf{m}} \Longrightarrow B), (A \Longrightarrow \cdot) \Longrightarrow A \star B; A \star \neg B}\,\left.\begin{array}{l} EC_{\mathcal{S}} \\ TP_{\mathcal{S}} \end{array}\right\}{}^{9}}{\emptyset_{\mathsf{m}}; S \Longrightarrow B}\,\neg R_{\mathcal{S}}{}^{13}}{\emptyset_{\mathsf{m}}; B \Longrightarrow B}\,WL_{\mathcal{S}}{}^{14}}{B \Longrightarrow B}\,Init_{\mathcal{S}}{}^{15}
$$

where $S = (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow A \star B; A \star \neg B\rangle$

Figure 5: A proof of $A \Longrightarrow (A \star B) \vee (A \star \neg B)$ in $\mathbf{S_{BBI}}$. We number all proof steps for comparison with Figure 9.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{I} \Longrightarrow \mathsf{I}}\,\mathsf{I}R_{\mathcal{S}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,Init_{\mathcal{S}}}{\cdot \Longrightarrow \neg A; A}\,\neg L_{\mathcal{S}} \qquad \dfrac{}{\emptyset_{\mathsf{m}} \Longrightarrow \mathsf{I}}\,\mathsf{I}R_{\mathcal{S}}}{(\cdot \Longrightarrow A), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \neg A \star \mathsf{I}}\,\star R_{\mathcal{S}}}{\cdot \Longrightarrow \neg A \star \mathsf{I}; A}\,\emptyset_{\mathsf{m}}U_{\mathcal{S}}}{\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I}); (\emptyset_{\mathsf{m}} \Longrightarrow \cdot)\langle \cdot \Longrightarrow A\rangle \Longrightarrow \cdot}\,\mathrel{-\!\!\star} L_{\mathcal{S}}}{(\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I}) \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow A}\,TC_{\mathcal{S}}}{(\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I})) \Longrightarrow A}\,\emptyset_{\mathsf{m}}U_{\mathcal{S}}}{\cdot \Longrightarrow (\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I})) \to A}\,\to R_{\mathcal{S}}
$$

Figure 6: A proof of $\cdot \Longrightarrow (\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I})) \to A$ in $\mathbf{S_{BBI}}$

Figure 6 proves the truth of $(\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I})) \to A$ in $\mathbf{S_{BBI}}$. The formula means that if $\neg A \star \mathsf{I}$ can never be true for a pair of a given node and its empty adjacent node, A must be true at the given node. It is a special instance of a more general formula $(B \mathrel{-\!\!\star} \neg(\neg A \star B)) \to A$. Note that we apply the rule $\emptyset_{\mathsf{m}}U_{\mathcal{S}}$ to create an empty node twice: first to add $\neg(\neg A \star \mathsf{I})$ to the truth context of the original node and second to add $\neg A$ to the falsehood context of the original node.

Figure 7 shows that if A is true at a node where I is simultaneously true, the node can be split into two child nodes in each of which A is true. The proof starts with a sequent

$$\mathsf{I} \wedge A \Longrightarrow A \star A,$$

which is extended with an empty node:

$$(\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow A \star A$$

Then we generate a node state $S = (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle \cdot \Longrightarrow A \star A\rangle$ to remember the relation between nodes with respect to the new empty node:

$$(\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow A \star A$$

$$
\begin{array}{ll}
\hline A \Longrightarrow A & \mathit{Init}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; A \Longrightarrow A & \mathit{WL}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; \emptyset_{\mathsf{m}}; A \Longrightarrow A & \mathit{WL}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; A; (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \cdot & \emptyset_{\mathsf{m}}D_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot); (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \cdot & \emptyset_{\mathsf{m}}D_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot); (\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow A) \Longrightarrow \cdot & \mathit{EC}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; (\emptyset_{\mathsf{m}} \Longrightarrow A)\langle(\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \cdot\rangle \Longrightarrow \cdot & \mathit{TP}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; A; (\emptyset_{\mathsf{m}} \Longrightarrow A)\langle(\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \cdot\rangle \Longrightarrow \cdot & \mathit{WL}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot); (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow A) \Longrightarrow \cdot & \mathit{TC}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot); (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot & \mathit{EC}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot); (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot & \mathit{EC}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle(\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot\rangle \Longrightarrow \cdot & \mathit{TP}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle(\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot\rangle \Longrightarrow A & \mathit{WR}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot); (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot & \mathit{TC}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot & \mathit{CL}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow A \star A & \mathit{WL}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; S \Longrightarrow A & \mathit{TP}_{\mathcal{S}}
\end{array}
\qquad
\begin{array}{ll}
\hline A \Longrightarrow A & \mathit{Init}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; A \Longrightarrow A & \mathit{WL}_{\mathcal{S}}
\end{array}
$$

$$
\begin{array}{ll}
\hline (\emptyset_{\mathsf{m}}; S \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow A \star A & \star R_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; S; S \Longrightarrow \cdot & \mathit{TP}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; S \Longrightarrow \cdot & \mathit{CL}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow A \star A & \mathit{TC}_{\mathcal{S}} \\
\hline (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow A \star A & \mathit{EC}_{\mathcal{S}} \\
\hline \emptyset_{\mathsf{m}}; A \Longrightarrow A \star A & \emptyset_{\mathsf{m}}U_{\mathcal{S}} \\
\hline \mathsf{I}; A \Longrightarrow A \star A & \mathsf{I}L_{\mathcal{S}} \\
\hline \mathsf{I} \wedge A \Longrightarrow A \star A & \wedge L_{\mathcal{S}}
\end{array}
$$

where $S = (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle \cdot \Longrightarrow A \star A\rangle$

Figure 7: A proof of $\mathsf{I} \wedge A \Longrightarrow A \star A$ in $\mathbf{S_{BBI}}$

Then we apply the rule $\star R_{\mathcal{S}}$ to prove $\emptyset_{\mathsf{m}}; S \Longrightarrow A$ and $\emptyset_{\mathsf{m}}; A \Longrightarrow A$. The first sequent eventually leads to $(\emptyset_{\mathsf{m}} \Longrightarrow A), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow \cdot$ which expresses a contradictory state in which A is both true and false at empty nodes. The second sequent is immediately provable.

4 Cut elimination in $\mathbf{S_{BBI}}$

We state the cut elimination property of $\mathbf{S_{BBI}}$ as follows:

Theorem 4.1 (Cut elimination)**.** *If* $\Gamma \Longrightarrow \Delta; C$ *and* $\Gamma'; C \Longrightarrow \Delta'$, *then* $\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$.

Section 6 gives an indirect proof of Theorem 4.1 which exploits the cut elimination property of the display calculus for Boolean BI [10]. Here we give a sketch of a direct proof which is inspired by the proof of cut elimination in original display logic [2].

The main complication in proving Theorem 4.1 is that the two contraction rules $\mathit{CL}_{\mathcal{S}}$ and $\mathit{CR}_{\mathcal{S}}$ duplicate a node state or a formula in their premise. In conjunction with the traverse rules, these contraction rules can produce copies of the cut formula C in different (smaller) sequents within the same sequent, as in:

$$\Gamma; (\Gamma_1; C^{n_1} \Longrightarrow \Delta_1), W_1; (\Gamma_2; C^{n_2} \Longrightarrow \Delta_2), W_2 \Longrightarrow \Delta$$

Here C^n means a truth context containing n copies of C. To represent such a sequent containing smaller

sequents with copies of the cut formula, we introduce the following definitions:

$$\begin{array}{rcll} \text{partial sequent} & \omega & ::= & [] \mid [;\gamma \Longrightarrow] \mid \Gamma;\gamma \Longrightarrow \Delta \\ \text{partial truth context} & \gamma & ::= & \sigma \mid \sigma;\gamma \\ \text{partial node state} & \sigma & ::= & \omega, W \mid W, \omega \mid \omega, \omega \mid \\ & & & \omega\langle W\rangle \mid W\langle\omega\rangle \mid \omega\langle\omega\rangle \end{array}$$

A partial sequent is a sequent with one or more holes in it; similarly partial truth contexts and partial node states contain one or more holes. We write $\omega[W_1]\cdots[W_n]$ for the sequent obtained by filling holes in ω with sequents $W_1, \cdots, W_n$ in that order; we define $\gamma[W_1]\cdots[W_n]$ and $\sigma[W_1]\cdots[W_n]$ in a similar way:

$$\omega[W_1]\cdots[W_n] = \begin{cases} W_1 & \text{where } \omega = [], n = 1 \\ \Gamma_1;\gamma[W_2]\cdots[W_n] \Longrightarrow \Delta_1 & \text{where } \omega = [;\gamma \Longrightarrow], \\ & \quad n > 1, W_1 = \Gamma_1 \Longrightarrow \Delta_1 \\ \Gamma;\gamma[W_1]\cdots[W_n] \Longrightarrow \Delta & \text{where } \omega = \Gamma;\gamma \Longrightarrow \Delta \end{cases}$$

Note that $[;\gamma \Longrightarrow]$ uses the first sequent to describe the reference node and remaining sequents to fill the holes in γ.

The proof of Theorem 4.1, which is inspired by the proof of cut elimination in display logic [2], proceeds by proving the following three lemmas. Here we say that $/C/$ holds if $\Gamma \Longrightarrow \Delta; A$ and $\Gamma'; A \Longrightarrow \Delta'$ implies $\Gamma;\Gamma' \Longrightarrow \Delta;\Delta'$ for any proper subformula A of C. We also write $\Gamma;\boxed{C} \Longrightarrow \Delta$ and $\Gamma \Longrightarrow \Delta;\boxed{C}$ to indicate that C is the principal formula of the last inference rule in their proofs. The proof of Lemma 4.3 uses Lemma 4.2, and the proof of Lemma 4.4 uses Lemma 4.3.

Lemma 4.2. *Suppose that $/C/$ holds. Then*
$\Gamma \Longrightarrow \Delta;\boxed{C}$ *and* $\Gamma';\boxed{C} \Longrightarrow \Delta'$ *imply* $\Gamma;\Gamma' \Longrightarrow \Delta;\Delta'$.

Proof. By case analysis on the cut formula C. □

Lemma 4.3. *Suppose that $/C/$ holds. Then*
$\omega[\Gamma_1 \Longrightarrow \Delta_1; C^{n_1}]\cdots[\Gamma_k \Longrightarrow \Delta_k; C^{n_k}]$ *and* $\Gamma';\boxed{C} \Longrightarrow \Delta'$
imply $\omega[\Gamma_1;\Gamma' \Longrightarrow \Delta_1;\Delta']\cdots[\Gamma_k;\Gamma' \Longrightarrow \Delta_k;\Delta']$.

Proof. By induction on the structure of proof $\mathcal{D}$ of $\omega[\Gamma_1 \Longrightarrow \Delta_1; C^{n_1}]\cdots[\Gamma_k \Longrightarrow \Delta_k; C^{n_k}]$. □

Lemma 4.4. *Suppose that $/C/$ holds. Then*
$\Gamma \Longrightarrow \Delta; C$ *and* $\omega[\Gamma'_1; C^{n_1} \Longrightarrow \Delta'_1]\cdots[\Gamma'_k; C^{n_k} \Longrightarrow \Delta'_k]$
imply $\omega[\Gamma;\Gamma'_1 \Longrightarrow \Delta;\Delta'_1]\cdots[\Gamma;\Gamma'_k \Longrightarrow \Delta;\Delta'_k]$.

Proof. By induction on the structure of proof $\mathcal{E}$ of $\omega'[\Gamma'_1; C^{n_1} \Longrightarrow \Delta'_1]\cdots[\Gamma'_k; C^{n_k} \Longrightarrow \Delta'_k]$. □

Then we complete the proof of Theorem 4.1 by induction on the structure of the cut formula C.

Corollary 4.5 (Consistency)**.** *There is no proof of* $\cdot \Longrightarrow \cdot$.

5 Soundness and completeness of $\mathbf{S_{BBI}}$

This section proves the soundness and completeness of the nested sequent calculus $\mathbf{S_{BBI}}$ with respect to the satisfaction relation in Section 3.2 (Theorems 5.1 and 5.2). It means that the syntactic provability of a sequent coincides with its semantic unsatisfiability, confirming the principle of proof by contradiction embodied in the definition of sequents.

Theorem 5.1 (Soundness)**.** *If* $\Gamma \Longrightarrow \Delta$, *then* $\not\models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$.

Theorem 5.2 (Completeness)**.** *If* $\not\models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$, *then* $\Gamma \Longrightarrow \Delta$.

The proof of soundness proceeds by induction on the structure of the proof of $\Gamma \Longrightarrow \Delta$. The proof of completeness uses a translation of a sequent W into a formula $[\![W]\!]_{\mathrm{w}}$ in Boolean BI defined as follows:

$$[\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}} \;=\; [\![\Gamma]\!]_{\mathrm{g}} \wedge \neg [\![\Delta]\!]_{\mathrm{d}}$$

$$\begin{array}{rclcrcl} [\![\cdot]\!]_{\mathrm{g}} & = & \top & & [\![\cdot]\!]_{\mathrm{d}} & = & \bot \\ [\![\Gamma; S]\!]_{\mathrm{g}} & = & [\![\Gamma]\!]_{\mathrm{g}} \wedge [\![S]\!]_{\mathrm{s}} & & [\![\Delta; A]\!]_{\mathrm{d}} & = & [\![\Delta]\!]_{\mathrm{d}} \vee A \end{array}$$

$$\begin{array}{rcl} [\![A]\!]_{\mathrm{s}} & = & A \\ [\![\emptyset_{\mathsf{m}}]\!]_{\mathrm{s}} & = & \mathsf{I} \\ [\![W_1, W_2]\!]_{\mathrm{s}} & = & [\![W_1]\!]_{\mathrm{w}} \star [\![W_2]\!]_{\mathrm{w}} \\ [\![W_1\langle W_2\rangle]\!]_{\mathrm{s}} & = & \neg([\![W_1]\!]_{\mathrm{w}} \mathrel{-\!\!\star} \neg [\![W_2]\!]_{\mathrm{w}}) \end{array}$$

Recall that a sequent W is a description of a set of nodes with respect to a reference node. $[\![W]\!]_{\mathrm{w}}$ is essentially the same description as W, except that it specifies the relationship between nodes through the use of multiplicative connectives $\star$ and $\mathrel{-\!\!\star}$ and negation $\neg$. The translation is characterized by Propositions 5.4 and 5.6 (whose proofs use Lemmas 5.3 and 5.5, respectively). We prove Proposition 5.4 by showing that the following two relations hold for any element w and valuation ρ:

Lemma 5.3.
$w, \rho \models_{\mathcal{W}} W$ if and only if $w, \rho \models [\![W]\!]_{\mathrm{w}}$.
$w, \rho \models_{\mathcal{S}} S$ if and only if $w, \rho \models [\![S]\!]_{\mathrm{s}}$.

Proposition 5.4. *$\not\models_{\mathcal{W}} W$ if and only if $\models \neg [\![W]\!]_{\mathrm{w}}$.*

Lemma 5.5.
$\Gamma; [\![\Gamma' \Longrightarrow \Delta']\!]_{\mathrm{w}} \Longrightarrow \Delta$ if and only if $\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$.
$\Gamma; [\![S]\!]_{\mathrm{s}} \Longrightarrow \Delta$ if and only if $\Gamma; S \Longrightarrow \Delta$.

Proposition 5.6. *If $\cdot \Longrightarrow \neg [\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}}$, then $\Gamma \Longrightarrow \Delta$.*

Propositions 5.4 and 5.6 allow us to complete the proof of completeness if we additionally show that $\models A$ implies $\cdot \Longrightarrow A$ (for $A = \neg [\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}}$). Since $\models A$ implies $\vdash A$ by Theorem 2.1, it suffices to prove the following lemma, whose proof exploits the cut elimination property of $\mathbf{S_{BBI}}$ (Theorem 4.1):

Lemma 5.7. *If $\vdash A$, then $\cdot \Longrightarrow A$.*

Proof. By induction on the structure of proof of $\vdash A$. The proof exploits the cut elimination property of $\mathbf{S_{BBI}}$, as illustrated in the following case:

Case $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}' \\ \vdash A \to (B \mathrel{-\!\!\star} C)\end{array}}{\vdash (A \star B) \to C} \mathrel{-\!\!\star}H_2$

$B \mathrel{-\!\!\star} C; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot$ by $\dfrac{\dfrac{}{B \Longrightarrow B}\, Init_{\mathcal{S}}' \quad \dfrac{}{C \Longrightarrow C}\, Init_{\mathcal{S}}'}{B \mathrel{-\!\!\star} C; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot} \mathrel{-\!\!\star}L_{\mathcal{S}}$

$A; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle; A \to (B \mathrel{-\!\!\star} C) \Longrightarrow \cdot$ by the rule $\to L_{\mathcal{S}}$

$\cdot \Longrightarrow A \to (B \mathrel{-\!\!\star} C)$ by induction hypothesis on $\mathcal{D}'$

$A; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot$ by Theorem 4.1

$\cdot \Longrightarrow A \star B \to C$ by $\dfrac{\dfrac{\dfrac{A; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}{(A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow C}\, TC_{\mathcal{S}}}{A \star B \Longrightarrow C} \star\mathsf{L}}{\cdot \Longrightarrow (A \star B) \to C} \to\mathsf{R}$

□

The following corollary shows that $\mathbf{S_{BBI}}$ is sound and complete with respect to the Kripke semantics in Section 2:

Corollary 5.8. *$\cdot \Longrightarrow A$ if and only if $\models A$.*

Proof. By Theorem 5.1, $\cdot \Longrightarrow A$ implies $\not\models_{\mathcal{W}} (\cdot \Longrightarrow A)$, which is equivalent to $\models A$ by definition. By Theorem 2.1, $\models A$ implies $\vdash A$, which implies $\cdot \Longrightarrow A$ by Lemma 5.7. □

Structural rules:

$$\frac{X \vdash_{\mathcal{D}} Y}{X; X' \vdash_{\mathcal{D}} Y}\ WL_{\mathcal{D}} \quad \frac{X \vdash_{\mathcal{D}} Y}{X \vdash_{\mathcal{D}} Y; Y'}\ WR_{\mathcal{D}} \quad \frac{X; X \vdash_{\mathcal{D}} Y}{X \vdash_{\mathcal{D}} Y}\ CL_{\mathcal{D}} \quad \frac{X \vdash_{\mathcal{D}} Y; Y}{X \vdash_{\mathcal{D}} Y}\ CR_{\mathcal{D}}$$

$$\frac{\frac{X; \emptyset_{\mathsf{a}} \vdash_{\mathcal{D}} Y}{X \vdash_{\mathcal{D}} Y}\ \emptyset_{\mathsf{a}}L_{\mathcal{D}}}{}\quad \frac{\frac{X \vdash_{\mathcal{D}} Y; \emptyset_{\mathsf{a}}}{X \vdash_{\mathcal{D}} Y}\ \emptyset_{\mathsf{a}}R_{\mathcal{D}}}{}$$

$$\frac{\frac{X_1; (X_2; X_3) \vdash_{\mathcal{D}} Y}{(X_1; X_2); X_3 \vdash_{\mathcal{D}} Y}\ AAL_{\mathcal{D}}}{} \quad \frac{\frac{X \vdash_{\mathcal{D}} Y_1; (Y_2; Y_3)}{X \vdash_{\mathcal{D}} (Y_1; Y_2); Y_3}\ AAR_{\mathcal{D}}}{} \quad \frac{\frac{X_1, (X_2, X_3) \vdash_{\mathcal{D}} Y}{(X_1, X_2), X_3 \vdash_{\mathcal{D}} Y}\ MAL_{\mathcal{D}}}{} \quad \frac{\frac{X, \emptyset_{\mathsf{m}} \vdash_{\mathcal{D}} Y}{X \vdash_{\mathcal{D}} Y}\ \emptyset_{\mathsf{m}}{}_{\mathcal{D}}}{}$$

Display rules:

$$\frac{\frac{X_2; X_1 \vdash_{\mathcal{D}} Y}{X_1 \vdash_{\mathcal{D}} \sharp X_2; Y}\ AD1a_{\mathcal{D}}}{X_1; X_2 \vdash_{\mathcal{D}} Y\ AD1b_{\mathcal{D}}} \quad \frac{\frac{X \vdash_{\mathcal{D}} Y_2; Y_1}{X; \sharp Y_1 \vdash_{\mathcal{D}} Y_2}\ AD2a_{\mathcal{D}}}{X \vdash_{\mathcal{D}} Y_1; Y_2\ AD2b_{\mathcal{D}}} \quad \frac{\frac{\sharp\sharp X \vdash_{\mathcal{D}} Y}{\sharp Y \vdash_{\mathcal{D}} \sharp X}\ AD3a_{\mathcal{D}}}{X \vdash_{\mathcal{D}} Y\ AD3b_{\mathcal{D}}} \quad \frac{\frac{X_2, X_1 \vdash_{\mathcal{D}} Y}{X_1 \vdash_{\mathcal{D}} X_2 \multimap Y}\ MD1a_{\mathcal{D}}}{X_1, X_2 \vdash_{\mathcal{D}} Y\ MD1b_{\mathcal{D}}}$$

Logical rules:

$$\frac{}{A \vdash_{\mathcal{D}} A}\ Init_{\mathcal{D}} \quad \frac{\emptyset_{\mathsf{a}} \vdash_{\mathcal{D}} Y}{\top \vdash_{\mathcal{D}} Y}\ \top L_{\mathcal{D}} \quad \frac{}{\emptyset_{\mathsf{a}} \vdash_{\mathcal{D}} \top}\ \top R_{\mathcal{D}} \quad \frac{}{\bot \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}}\ \bot L_{\mathcal{D}} \quad \frac{X \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}}{X \vdash_{\mathcal{D}} \bot}\ \bot R_{\mathcal{D}} \quad \frac{\sharp A \vdash_{\mathcal{D}} Y}{\neg A \vdash_{\mathcal{D}} Y}\ \neg L_{\mathcal{D}}$$

$$\frac{X \vdash_{\mathcal{D}} \sharp A}{X \vdash_{\mathcal{D}} \neg A}\ \neg R_{\mathcal{D}} \quad \frac{A; B \vdash_{\mathcal{D}} Y}{A \wedge B \vdash_{\mathcal{D}} Y}\ \wedge L_{\mathcal{D}}$$

$$\frac{X_1 \vdash_{\mathcal{D}} A \quad X_2 \vdash_{\mathcal{D}} B}{X_1; X_2 \vdash_{\mathcal{D}} A \wedge B}\ \wedge R_{\mathcal{D}} \quad \frac{A \vdash_{\mathcal{D}} Y_1 \quad B \vdash_{\mathcal{D}} Y_2}{A \vee B \vdash_{\mathcal{D}} Y_1; Y_2}\ \vee L_{\mathcal{D}} \quad \frac{X \vdash_{\mathcal{D}} A; B}{X \vdash_{\mathcal{D}} A \vee B}\ \vee R_{\mathcal{D}}$$

$$\frac{X \vdash_{\mathcal{D}} A \quad B \vdash_{\mathcal{D}} Y}{A \to B \vdash_{\mathcal{D}} \sharp X; Y}\ \to L_{\mathcal{D}} \quad \frac{X; A \vdash_{\mathcal{D}} B}{X \vdash_{\mathcal{D}} A \to B}\ \to R_{\mathcal{D}}$$

$$\frac{\emptyset_{\mathsf{m}} \vdash_{\mathcal{D}} Y}{\mathsf{I} \vdash_{\mathcal{D}} Y}\ \mathsf{I}L_{\mathcal{D}} \quad \frac{}{\emptyset_{\mathsf{m}} \vdash_{\mathcal{D}} \mathsf{I}}\ \mathsf{I}R_{\mathcal{D}} \quad \frac{A, B \vdash_{\mathcal{D}} Y}{A \star B \vdash_{\mathcal{D}} Y}\ \star L_{\mathcal{D}} \quad \frac{X_1 \vdash_{\mathcal{D}} A \quad X_2 \vdash_{\mathcal{D}} B}{X_1, X_2 \vdash_{\mathcal{D}} A \star B}\ \star R_{\mathcal{D}}$$

$$\frac{X \vdash_{\mathcal{D}} A \quad B \vdash_{\mathcal{D}} Y}{A \mathrel{-\!\!\star} B \vdash_{\mathcal{D}} X \multimap Y}\ \mathrel{-\!\!\star} L_{\mathcal{D}} \quad \frac{X, A \vdash_{\mathcal{D}} B}{X \vdash_{\mathcal{D}} A \mathrel{-\!\!\star} B}\ \mathrel{-\!\!\star} R_{\mathcal{D}}$$

Figure 8: Display calculus $\mathbf{DL_{BBI}}$ for Boolean BI by Brotherston [10]

6 Display calculus for Boolean BI

This section reviews the display calculus $\mathbf{DL_{BBI}}$ for Boolean BI by Brotherston [10]. We establish the equivalence between $\mathbf{S_{BBI}}$ and $\mathbf{DL_{BBI}}$, and show that $\mathbf{S_{BBI}}$ is an optimization of $\mathbf{DL_{BBI}}$.

6.1 Definition and properties of $\mathbf{DL_{BBI}}$

The display calculus $\mathbf{DL_{BBI}}$ uses a judgment $X \vdash_{\mathcal{D}} Y$, called a *consecution*, in its inference rules; X is called an A-structure and Y a C-structure:

$$\begin{array}{lrcl} A\text{-structure} & X & ::= & A \mid \emptyset_{\mathsf{a}} \mid \emptyset_{\mathsf{m}} \mid \sharp Y \mid X; X \mid X, X \\ C\text{-structure} & Y & ::= & A \mid \emptyset_{\mathsf{a}} \mid \sharp X \mid Y; Y \mid X \multimap Y \end{array}$$

A-structures are essentially an extension of bunches in Boolean BI with negative structures $\sharp Y$. C-structures do not use the multiplicative unit $\emptyset_{\mathsf{m}}$ and the multiplicative structural connective ,, but introduce a negative structural connective $\sharp$ and a multiplicative structural connective $\multimap$ which is originally from the display calculus for linear logic [3].

Figure 8 shows the display calculus $\mathbf{DL_{BBI}}$. A rule with a double line between premise and conclusion is invertible, *i.e.*, the premise implies the conclusion and vice versa. The inference rules of $\mathbf{DL_{BBI}}$ are divided into three groups: structural rules, display rules, and logical rules. Structural rules deal with the structural properties of consecutions. Display rules introduce or eliminate $\sharp Y$, $\sharp X$, and $X \multimap Y$

as necessary in order to "display" a target A-structure or C-structure as the sole element in the left or right side of a consecution. A logical rule focuses on a single formula that has already been "displayed" in the left or right side of a consecution by the display rules. We may generalize the rule $Init_{\mathcal{D}}$ to the following derivable rule:

$$\frac{}{A \vdash_{\mathcal{D}} A}\ Init_{\mathcal{D}}$$

Two consecutions $X \vdash_{\mathcal{D}} Y$ and $X' \vdash_{\mathcal{D}} Y'$ are display-equivalent, written $X \vdash_{\mathcal{D}} Y \equiv_{\mathcal{D}} X' \vdash_{\mathcal{D}} Y'$, if either consecution can be derived from the other consecution by the display rules. As proof search in $\mathbf{DL_{BBI}}$ often converts a consecution into another display-equivalent consecution, we use the following derived rule to simplify representations of proofs:

$$\frac{X' \vdash_{\mathcal{D}} Y' \quad X \vdash_{\mathcal{D}} Y \equiv_{\mathcal{D}} X' \vdash_{\mathcal{D}} Y'}{X \vdash_{\mathcal{D}} Y}\ E^*_{\mathcal{D}}$$

Brotherston [10] presents the following results on $\mathbf{DL_{BBI}}$:

Theorem 6.1 (Cut elimination)**.** *If $X \vdash_{\mathcal{D}} A$ and $A \vdash_{\mathcal{D}} Y$, then $X \vdash_{\mathcal{D}} Y$.*

Theorem 6.2 (Soundness and completeness)**.** *$\emptyset_{\mathsf{a}} \vdash_{\mathcal{D}} A$ if and only if $\models A$.*

6.2 Equivalence between $\mathbf{S_{BBI}}$ and $\mathbf{DL_{BBI}}$

Although the equivalence between $\mathbf{S_{BBI}}$ and $\mathbf{DL_{BBI}}$ is obvious from Corollary 5.8 and Theorem 6.2, it does not illuminate how sequents and consecutions are related. Here we present direct translations between the two calculi and study their similarities and differences.

Given a consecution $X \vdash_{\mathcal{D}} Y$, we translate A-structure X to a sequent $[\![X]\!]_{\mathcal{X}}$ and C-structure Y to another sequent $[\![Y]\!]_{\mathcal{Y}}$. Then we combine $[\![X]\!]_{\mathcal{X}}$ and $[\![Y]\!]_{\mathcal{Y}}$ to build a single sequent $[\![X \vdash_{\mathcal{D}} Y]\!]_{\mathcal{C}}$. Let us write $(\Gamma \Longrightarrow \Delta) \uplus (\Gamma' \Longrightarrow \Delta')$ for $\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$. We define $[\![X \vdash_{\mathcal{D}} Y]\!]_{\mathcal{C}}$ as follows:

$$\begin{array}{rcl}
[\![X \vdash_{\mathcal{D}} Y]\!]_{\mathcal{C}} & = & [\![X]\!]_{\mathcal{X}} \uplus [\![Y]\!]_{\mathcal{Y}} \\
[\![A]\!]_{\mathcal{X}} & = & A \Longrightarrow \cdot \\
[\![\emptyset_{\mathsf{a}}]\!]_{\mathcal{X}} & = & \cdot \Longrightarrow \cdot \\
[\![\emptyset_{\mathsf{m}}]\!]_{\mathcal{X}} & = & \emptyset_{\mathsf{m}} \Longrightarrow \cdot \\
[\![\sharp Y]\!]_{\mathcal{X}} & = & [\![Y]\!]_{\mathcal{Y}} \\
[\![X_1; X_2]\!]_{\mathcal{X}} & = & [\![X_1]\!]_{\mathcal{X}} \uplus [\![X_2]\!]_{\mathcal{X}} \\
[\![X_1, X_2]\!]_{\mathcal{X}} & = & [\![X_1]\!]_{\mathcal{X}}, [\![X_2]\!]_{\mathcal{X}} \Longrightarrow \cdot \\
[\![A]\!]_{\mathcal{Y}} & = & \cdot \Longrightarrow A \\
[\![\emptyset_{\mathsf{a}}]\!]_{\mathcal{Y}} & = & \cdot \Longrightarrow \cdot \\
[\![\sharp X]\!]_{\mathcal{Y}} & = & [\![X]\!]_{\mathcal{X}} \\
[\![Y_1; Y_2]\!]_{\mathcal{Y}} & = & [\![Y_1]\!]_{\mathcal{Y}} \uplus [\![Y_2]\!]_{\mathcal{Y}} \\
[\![X \multimap Y]\!]_{\mathcal{Y}} & = & [\![X]\!]_{\mathcal{X}} \langle [\![Y]\!]_{\mathcal{Y}} \rangle \Longrightarrow \cdot
\end{array}$$

Like sequents in $\mathbf{S_{BBI}}$, both A-structure X and C-structure Y are essentially descriptions of a set of nodes, but formulas in X are regarded as true whereas formulas in Y as false in the reference node. We also observe that multiplicative structures X_1, X_2 and $X \multimap Y$ correspond to multiplicative pairs and adjoint pairs in $\mathbf{S_{BBI}}$.

Lemma 6.3 shows that $\mathbf{S_{BBI}}$ is as expressive as $\mathbf{DL_{BBI}}$:

Lemma 6.3. *If $X \vdash_{\mathcal{D}} Y$ holds in $\mathbf{DL_{BBI}}$, then $[\![X \vdash_{\mathcal{D}} Y]\!]_{\mathcal{C}}$ holds in $\mathbf{S_{BBI}}$.*

Proof. By induction on the structure of proof of $X \vdash_{\mathcal{D}} Y$. □

Given a sequent $\Gamma \Longrightarrow \Delta$, we translate Γ to an A-structure $[\![\Gamma]\!]_{\mathcal{G}}$ and Δ to a C-structure $[\![\Delta]\!]_{\mathcal{D}}$. Then we combine $[\![\Gamma]\!]_{\mathcal{G}}$ and $[\![\Delta]\!]_{\mathcal{D}}$ to another A-structure $[\![\Gamma \Longrightarrow \Delta]\!]_{\mathcal{W}}$ defined as follows:

$$[\![\Gamma \Longrightarrow \Delta]\!]_{\mathcal{W}} \quad = \quad [\![\Gamma]\!]_{\mathcal{G}}; \sharp [\![\Delta]\!]_{\mathcal{D}}$$

$$\begin{array}{rcl}
[\![\cdot]\!]_{\mathcal{G}} & = & \emptyset_{\mathsf{a}} \\
[\![\Gamma; S]\!]_{\mathcal{G}} & = & [\![\Gamma]\!]_{\mathcal{G}}; [\![S]\!]_{\mathcal{S}}
\end{array}
\qquad
\begin{array}{rcl}
[\![\cdot]\!]_{\mathcal{D}} & = & \emptyset_{\mathsf{a}} \\
[\![\Delta; A]\!]_{\mathcal{D}} & = & [\![\Delta]\!]_{\mathcal{D}}; A
\end{array}$$

$$\begin{array}{rcl} [\![A]\!]_{\mathcal{S}} & = & A \\ [\![\emptyset_{\mathsf{m}}]\!]_{\mathcal{S}} & = & \emptyset_{\mathsf{m}} \\ [\![W_1, W_2]\!]_{\mathcal{S}} & = & [\![W_1]\!]_{\mathcal{W}}, [\![W_2]\!]_{\mathcal{W}} \\ [\![W_1\langle W_2\rangle]\!]_{\mathcal{S}} & = & \sharp([\![W_1]\!]_{\mathcal{W}} \multimap \sharp[\![W_2]\!]_{\mathcal{W}}) \end{array}$$

$[\![W]\!]_{\mathcal{W}}$ is defined in a similar way to $[\![W]\!]_{\mathrm{w}}$ given in Section 5. For example, $[\![W_1, W_2]\!]_{\mathcal{S}}$ coincides with $[\![W_1, W_2]\!]_{\mathrm{s}}$ if we write , as $\star$, and $[\![W_1\langle W_2\rangle]\!]_{\mathcal{S}}$ coincides with $[\![W_1\langle W_2\rangle]\!]_{\mathrm{s}}$ if we write $\sharp$ as $\neg$ and $\multimap$ as $-\!\!\star$. The comparison between $[\![W]\!]_{\mathcal{W}}$ and $[\![W]\!]_{\mathrm{w}}$ reveals a correspondence between structural connectives in $\mathbf{DL_{BBI}}$ and logical connectives in Boolean BI that complies with the translation from $\mathbf{DL_{BBI}}$ to Boolean BI given in [10].

Lemma 6.4 shows that $\mathbf{DL_{BBI}}$ is as expressive as $\mathbf{S_{BBI}}$. In conjunction with Lemma 6.3, it proves the equivalence between $\mathbf{S_{BBI}}$ and $\mathbf{DL_{BBI}}$. We also remark that with Lemmas 6.3 and 6.4, we can give another indirect proof of Theorem 4.1 by exploiting Theorem 6.1.

Lemma 6.4. *If* $\Gamma \Longrightarrow \Delta$, *then* $[\![\Gamma \Longrightarrow \Delta]\!]_{\mathcal{W}} \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}$.

Proof. By induction on the structure of proof of $\Gamma \Longrightarrow \Delta$. □

With Lemmas 6.3 and 6.4, we can now give another indirect proof of Theorem 4.1 by exploiting Theorem 6.1. We need an additional lemma relating the two translations:

Lemma 6.5. $[\![[\![\Gamma]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} = \Gamma \Longrightarrow \cdot$ *and* $[\![[\![\Delta]\!]_{\mathcal{D}}]\!]_{\mathcal{Y}} = \cdot \Longrightarrow \Delta$.

Proof of Theorem 4.1. Suppose $\Gamma \Longrightarrow \Delta; C$ and $\Gamma'; C \Longrightarrow \Delta'$.
$[\![\Gamma \Longrightarrow \Delta; C]\!]_{\mathcal{W}} \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}$ and $[\![\Gamma'; C \Longrightarrow \Delta']\!]_{\mathcal{W}} \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}$ by Lemma 6.4
$[\![\Gamma]\!]_{\mathcal{G}}; \sharp[\![\Delta]\!]_{\mathcal{D}} \vdash_{\mathcal{D}} C$ and $C \vdash_{\mathcal{D}} \sharp[\![\Gamma']\!]_{\mathcal{G}}; [\![\Delta']\!]_{\mathcal{D}}$ by the display rules and the rule $\emptyset_{\mathsf{a}} R_{\mathcal{D}}$
$[\![\Gamma]\!]_{\mathcal{G}}; \sharp[\![\Delta]\!]_{\mathcal{D}} \vdash_{\mathcal{D}} \sharp[\![\Gamma']\!]_{\mathcal{G}}; [\![\Delta']\!]_{\mathcal{D}}$ by Theorem 6.1
$[\![\Gamma; \Gamma']\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta; \Delta']\!]_{\mathcal{D}}$ by the display rules
$[\![[\![\Gamma; \Gamma']\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta; \Delta']\!]_{\mathcal{D}}]\!]_{\mathcal{C}}$ in $\mathbf{S_{BBI}}$ by Lemma 6.3
$\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$ by Lemma 6.5 □

6.3 $\mathbf{S_{BBI}}$ as an optimization of $\mathbf{DL_{BBI}}$

The two translations in Section 6.2 suggest that sequents in $\mathbf{S_{BBI}}$ essentially represent a normal form of consecutions in $\mathbf{DL_{BBI}}$. We say that a consecution is of the normal form if it permits a negative structure $\sharp X$ only in the right side of $\multimap$ according to the revised definition of C-structures:

$$C\text{-structure} \quad Y \quad ::= \quad A \mid \emptyset_{\mathsf{a}} \mid Y; Y \mid X \multimap \sharp X$$

It turns out that every consecution $X \vdash_{\mathcal{D}} Y$ can be converted by the structural rules (for associativity and $\emptyset_{\mathsf{a}}$) and the display rules to its normal form $[\![[\![X \vdash_{\mathcal{D}} Y]\!]_{\mathcal{C}}]\!]_{\mathcal{W}} \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}$, whose formulas form the same syntactic structure as the sequent $[\![X \vdash_{\mathcal{D}} Y]\!]_{\mathcal{C}}$. Thus we may think of sequents as representing consecutions of the normal form and $\mathbf{S_{BBI}}$ as a sequent calculus that directly manipulates such consecutions.

Note that consecutions of the normal form in $\mathbf{DL_{BBI}}$ still require the negative structural connective $\sharp$ whereas $\mathbf{S_{BBI}}$ requires no such negative structural connective. Hence the first three pairs of display rules in Figure 8 (from $AD1a_{\mathcal{D}}$ to $AD3b_{\mathcal{D}}$) have no counterparts in $\mathbf{S_{BBI}}$, which implies that proof searches in $\mathbf{S_{BBI}}$ are always simpler than in $\mathbf{DL_{BBI}}$ (except in trivial cases) because of the extra cost of applying such display rules in $\mathbf{DL_{BBI}}$.[1]

Figure 9 shows an example of proving in $\mathbf{DL_{BBI}}$ the same formula as in Figure 5. The proof search proceeds in a similar manner: first creating $\emptyset_{\mathsf{m}}$, next applying a contraction rule to duplicate a C-structure, then consuming the C-structure, and finally applying the rule $Init_{\mathcal{D}}$. We number each proof step to mark the correspondence between proof steps in Figures 5 and 9. Note that the display rule $MD1a_{\mathcal{D}}$ expands to a pair of a traverse rule ($TC_{\mathcal{S}}$ or $TP_{\mathcal{S}}$) and the rule $EC_{\mathcal{S}}$ (at proof steps 3, 5, and 9). We observe that $\mathbf{DL_{BBI}}$ takes extra six proof steps all of which apply display rules (marked in rectangles). This example illustrates that $\mathbf{S_{BBI}}$ is a formal system which can be obtained from an optimization of $\mathbf{DL_{BBI}}$ that dispenses with those display rules dealing with the negative structural connective $\sharp$ and revises all the logical rules accordingly.

[1] The last pair of display rules $MD1a_{\mathcal{D}}$ and $MD1b_{\mathcal{D}}$ correspond to four combinations of the traverse rules and the structural rule $EC_{\mathcal{S}}$ in $\mathbf{S_{BBI}}$, which is the reason that we refer to the rules $TC_{\mathcal{S}}$ and $TP_{\mathcal{S}}$ in Figure 4 not as display rules but as traverse rules.

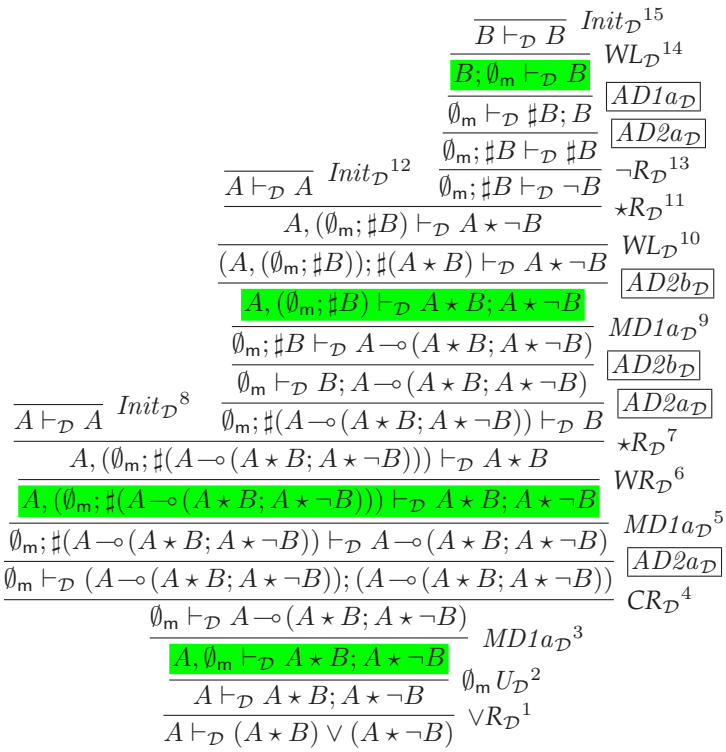


Figure 9: A proof of $A \vdash_{\mathcal{D}} (A \star B) \vee (A \star \neg B)$ in $\mathbf{DL_{BBI}}$ (adapted from Brotherston [9])

7 Nested sequent calculus $\mathbf{CS_{BBI}}$

While the presence of multiplicative connectives from intuitionistic linear logic may suggest the inverse method for implementing a theorem prover for Boolean BI, the contraction property alone makes the inverse method not so ideal as it seems, as already observed in previous work on intuitionistic BI by Donnelly *et al*. [18]. This is especially the case for $\mathbf{S_{BBI}}$, which, unlike sequent calculi for intuitionistic BI, needs to use a graph structure of sequents instead of a tree structure. For example, it is not clear how to generate a minimal graph structure that weakens to a given pair of graph structures (for those inference rules with two sequents in the premise). Thus we choose to use a backward search strategy in our theorem prover.

As the first step, we obtain the nested sequent calculus $\mathbf{CS_{BBI}}$ (Contraction-free $\mathbf{S_{BBI}}$), shown in Figure 10, by embedding the weakening and contraction rules ($WL_{\mathcal{S}}$, $WR_{\mathcal{S}}$, $CL_{\mathcal{S}}$, and $CR_{\mathcal{S}}$) into all the other rules of $\mathbf{S_{BBI}}$. Similarly to $\mathbf{S_{BBI}}$, the inference rules are divided into structural rules, traverse rules, and logical rules. Since contraction is built into all the inference rules in $\mathbf{CS_{BBI}}$, the premise of every structural rule subsumes the graph structure of nodes represented by the conclusion. Except the rule $EC_{\mathcal{C}}$ which rewrites a node state according to commutativity of sequents, every structural rule has a premise that strictly extends its conclusion with new relations between nodes. Below we further describe the two structural rules $EA_{\mathcal{C}}$ and $\emptyset_{\mathsf{m}}U_{\mathcal{C}}$, in which the premise restructures nodes by changing parent-child and sibling relations but is also able to recover the original structure of nodes expressed in the conclusion.

In the rule $EA_{\mathcal{C}}$, each sequent $W_i \oplus S_i$ ($i = 1, 2, 3$) in the premise represents the same graph structure as the conclusion, except that its reference node is now described by W_i. Extending W_i with S_i in this way is a part of building contraction into the rule $EA_{\mathcal{C}}$ and is thus necessary for the completeness of $\mathbf{CS_{BBI}}$ with respect to Boolean BI. As an example, consider the proof of a sequent

$$(W_1, W_2 \Longrightarrow B), W_3 \Longrightarrow \cdot \text{ with } \begin{cases} W_1 & = & A \Longrightarrow \cdot \\ W_2 & = & C \mathrel{-\!\!\star} \neg(\neg(A \mathrel{-\!\!\star} B) \star C) \Longrightarrow \cdot \\ W_3 & = & C \Longrightarrow \cdot \end{cases} .$$

Structural rules:

$$\frac{\Gamma;(\Gamma';W_1,W_2\Longrightarrow\Delta'),W_3;W_1\oplus S_1,(W_2\oplus S_2,W_3\oplus S_3\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;(\Gamma';W_1,W_2\Longrightarrow\Delta'),W_3\Longrightarrow\Delta}\ EA_{\mathcal{C}}$$

$$\text{where}\ \begin{cases} S_1 = W_2\langle\Gamma';W_3\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta'\rangle \\ S_2 = W_1\langle\Gamma';W_3\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta'\rangle \\ S_3 = (\Gamma';W_1,W_2\Longrightarrow\Delta')\langle\Gamma\Longrightarrow\Delta\rangle \end{cases}$$

$$\frac{\Gamma;W_2,W_1\Longrightarrow\Delta}{\Gamma;W_1,W_2\Longrightarrow\Delta}\ EC_{\mathcal{C}}\qquad\frac{\Gamma;(\Gamma\Longrightarrow\Delta),(\emptyset_{\mathsf{m}}\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma\Longrightarrow\Delta}\ \emptyset_{\mathsf{m}}U_{\mathcal{C}}$$

$$\frac{\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2;\emptyset_{\mathsf{m}}\Longrightarrow\Delta_2);\Gamma_1;S\Longrightarrow\Delta;\Delta_1}{\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2;\emptyset_{\mathsf{m}}\Longrightarrow\Delta_2)\Longrightarrow\Delta}\ \emptyset_{\mathsf{m}}D_{\mathcal{C}}\quad\text{where } S=(\Gamma_2;\emptyset_{\mathsf{m}}\Longrightarrow\Delta_2)\langle\Gamma\Longrightarrow\Delta\rangle$$

Traverse rules (p for parent, c for child, s for sibling):

$$\frac{\Gamma_{\mathrm{c1}};(\Gamma_{\mathrm{c2}}\Longrightarrow\Delta_{\mathrm{c2}})\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta_{\mathrm{c1}}}{\Gamma;(\Gamma_{\mathrm{c1}}\Longrightarrow\Delta_{\mathrm{c1}}),(\Gamma_{\mathrm{c2}}\Longrightarrow\Delta_{\mathrm{c2}})\Longrightarrow\Delta}\ TC_{\mathcal{C}}\qquad\frac{\Gamma_{\mathrm{p}};(\Gamma\Longrightarrow\Delta),(\Gamma_{\mathrm{s}}\Longrightarrow\Delta_{\mathrm{s}})\Longrightarrow\Delta_{\mathrm{p}}}{\Gamma;(\Gamma_{\mathrm{s}}\Longrightarrow\Delta_{\mathrm{s}})\langle\Gamma_{\mathrm{p}}\Longrightarrow\Delta_{\mathrm{p}}\rangle\Longrightarrow\Delta}\ TP_{\mathcal{C}}$$

Logical rules:

$$\frac{}{\Gamma;A\Longrightarrow\Delta;A}\ Init_{\mathcal{C}}\qquad\frac{}{\Gamma;\bot\Longrightarrow\Delta}\ \bot L_{\mathcal{C}}\qquad\frac{\Gamma\Longrightarrow\Delta;A}{\Gamma;\neg A\Longrightarrow\Delta}\ \neg L_{\mathcal{C}}\qquad\frac{\Gamma;A\Longrightarrow\Delta}{\Gamma\Longrightarrow\Delta;\neg A}\ \neg R_{\mathcal{C}}$$

$$\frac{\Gamma;A\Longrightarrow\Delta\quad\Gamma;B\Longrightarrow\Delta}{\Gamma;A\vee B\Longrightarrow\Delta}\ \vee L_{\mathcal{C}}\qquad\frac{\Gamma\Longrightarrow\Delta;A;B}{\Gamma\Longrightarrow\Delta;A\vee B}\ \vee R_{\mathcal{C}}\qquad\frac{\Gamma;\emptyset_{\mathsf{m}}\Longrightarrow\Delta}{\Gamma;\mathsf{I}\Longrightarrow\Delta}\ \mathsf{I}L_{\mathcal{C}}\qquad\frac{}{\Gamma;\emptyset_{\mathsf{m}}\Longrightarrow\Delta;\mathsf{I}}\ \mathsf{I}R_{\mathcal{C}}$$

$$\frac{\Gamma;(A\Longrightarrow\cdot),(B\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;A\star B\Longrightarrow\Delta}\ \star L_{\mathcal{C}}$$

$$\frac{\Gamma;(\Gamma_1\Longrightarrow\Delta_1;A),(\Gamma_2\Longrightarrow\Delta_2)\Longrightarrow\Delta;A\star B\quad\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2\Longrightarrow\Delta_2;B)\Longrightarrow\Delta;A\star B}{\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2\Longrightarrow\Delta_2)\Longrightarrow\Delta;A\star B}\ \star R_{\mathcal{C}}$$

$$\frac{\Gamma;(\Gamma_1\Longrightarrow\Delta_1;A)\langle\Gamma_2\Longrightarrow\Delta_2\rangle;A\mathrel{-\!\!\star}B\Longrightarrow\Delta\quad\Gamma;(\Gamma_1\Longrightarrow\Delta_1)\langle\Gamma_2;B\Longrightarrow\Delta_2\rangle;A\mathrel{-\!\!\star}B\Longrightarrow\Delta}{\Gamma;(\Gamma_1\Longrightarrow\Delta_1)\langle\Gamma_2\Longrightarrow\Delta_2\rangle;A\mathrel{-\!\!\star}B\Longrightarrow\Delta}\ \mathrel{-\!\!\star}L_{\mathcal{C}}$$

$$\frac{\Gamma;(A\Longrightarrow\cdot)\langle\cdot\Longrightarrow B\rangle\Longrightarrow\Delta}{\Gamma\Longrightarrow\Delta;A\mathrel{-\!\!\star}B}\ \mathrel{-\!\!\star}R_{\mathcal{C}}$$

Figure 10: Nested sequent calculus $\mathbf{CS_{BBI}}$. We define $(\Gamma\Longrightarrow\Delta)\oplus S$ as $\Gamma;S\Longrightarrow\Delta$ in the rule $EA_{\mathcal{C}}$.

We apply the rule $EA_{\mathcal{C}}$ to generate another sequent

$$(W_1,W_2\Longrightarrow B),W_3;W_1\oplus S_1,(W_2\oplus S_2,W_3\oplus S_3\Longrightarrow\cdot)\Longrightarrow\cdot\ \text{with}\ \begin{cases} S_1 &=& W_2\langle W_3\langle\cdot\Longrightarrow\cdot\rangle\Longrightarrow B\rangle \\ S_2 &=& W_1\langle W_3\langle\cdot\Longrightarrow\cdot\rangle\Longrightarrow B\rangle \\ S_3 &=& (W_1,W_2\Longrightarrow B)\langle\cdot\Longrightarrow\cdot\rangle \end{cases}.$$

Eventually we reach the following sequent which is provable only because of the interaction between S_2 and $A\mathrel{-\!\!\star}B$ via the rule $\mathrel{-\!\!\star}L_{\mathcal{C}}$:

$$C\mathrel{-\!\!\star}\neg(\neg(A\mathrel{-\!\!\star}B)\star C);(W_3\oplus S_3)\langle(W_1\oplus S_1)\langle S_1\Longrightarrow\cdot\rangle\Longrightarrow\neg(A\mathrel{-\!\!\star}B)\star C\rangle;\boxed{S_2;A\mathrel{-\!\!\star}B}\Longrightarrow\cdot$$

Hence, if we omit S_i in the premise of the rule $EA_{\mathcal{C}}$, we lose the completeness of $\mathbf{CS_{BBI}}$. The example sequent arises in proving formula $\neg(((A\star(C\mathrel{-\!\!\star}\neg(\neg(A\mathrel{-\!\!\star}B)\star C)))\wedge\neg B)\star C)$; its proofs in $\mathbf{S_{BBI}}$ and $\mathbf{CS_{BBI}}$ are given in Figures 11 and 12, respectively.

The rule $\emptyset_{\mathsf{m}}U_{\mathcal{C}}$ creates a new child node with a special form of sequent $\emptyset_{\mathsf{m}}\Longrightarrow\cdot$, which can be absorbed back into the parent node by the rule $\emptyset_{\mathsf{m}}D_{\mathcal{C}}$. Intuitively $\emptyset_{\mathsf{m}}\Longrightarrow\cdot$ describes an empty node whose sibling node can be identified with its parent node. Similarly to the rule $EA_{\mathcal{C}}$, the premise of the rule

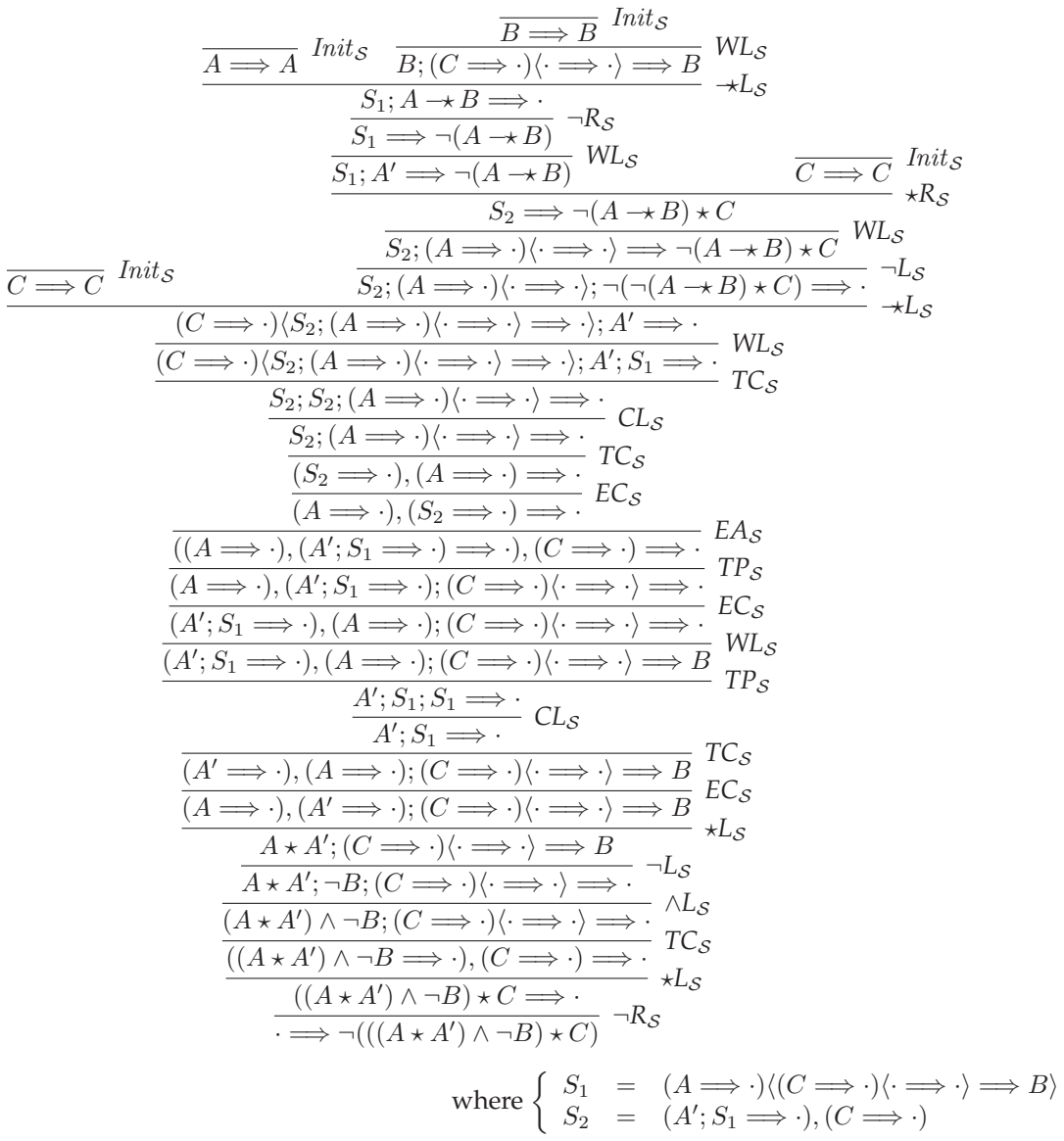


Figure 11: A proof of $\cdot \Longrightarrow \neg(((A \star A') \wedge \neg B) \star C)$ in $\mathbf{S_{BBI}}$ where $A' = C \mathrel{-\!\!\star} \neg(\neg(A \mathrel{-\!\!\star} B) \star C)$

$\emptyset_{\mathsf{m}} D_{\mathcal{C}}$ combines the conclusion with a new sequent $\Gamma_1; S \Longrightarrow \Delta_1$, which represents the same graph structure as the conclusion but has a different reference node. Omitting S in the premise also costs the completeness of $\mathbf{CS_{BBI}}$. For example, if we omit S in the premise of the rule $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, the sequent in Figure 13 is not provable because its proof depends on the interaction between $A \mathrel{-\!\!\star} B$ and S via the rule $\mathrel{-\!\!\star} L_{\mathcal{C}}$; a corresponding proof in $\mathbf{S_{BBI}}$ is given in Figure 14.

Every inference rule in $\mathbf{CS_{BBI}}$ is invertible, *i.e.*, the premise implies the conclusion and vice versa. We can formally prove that both weakening and contraction are admissible in $\mathbf{CS_{BBI}}$. We can also prove the equivalence between $\mathbf{S_{BBI}}$ and $\mathbf{CS_{BBI}}$.

Theorem 7.1 (Weakening and contraction in $\mathbf{CS_{BBI}}$)**.**
If $\Gamma \Longrightarrow \Delta$*, then* $\Gamma; S \Longrightarrow \Delta$ *and* $\Gamma \Longrightarrow \Delta; A$*.*
If $\Gamma; S; S \Longrightarrow \Delta$*, then* $\Gamma; S \Longrightarrow \Delta$*.*
If $\Gamma \Longrightarrow \Delta; A; A$*, then* $\Gamma \Longrightarrow \Delta; A$*.*

Theorem 7.2 (Equivalence between $\mathbf{S_{BBI}}$ and $\mathbf{CS_{BBI}}$)**.** $\Gamma \Longrightarrow \Delta$ *in* $\mathbf{S_{BBI}}$ *if and only if* $\Gamma \Longrightarrow \Delta$ *in* $\mathbf{CS_{BBI}}$*.*

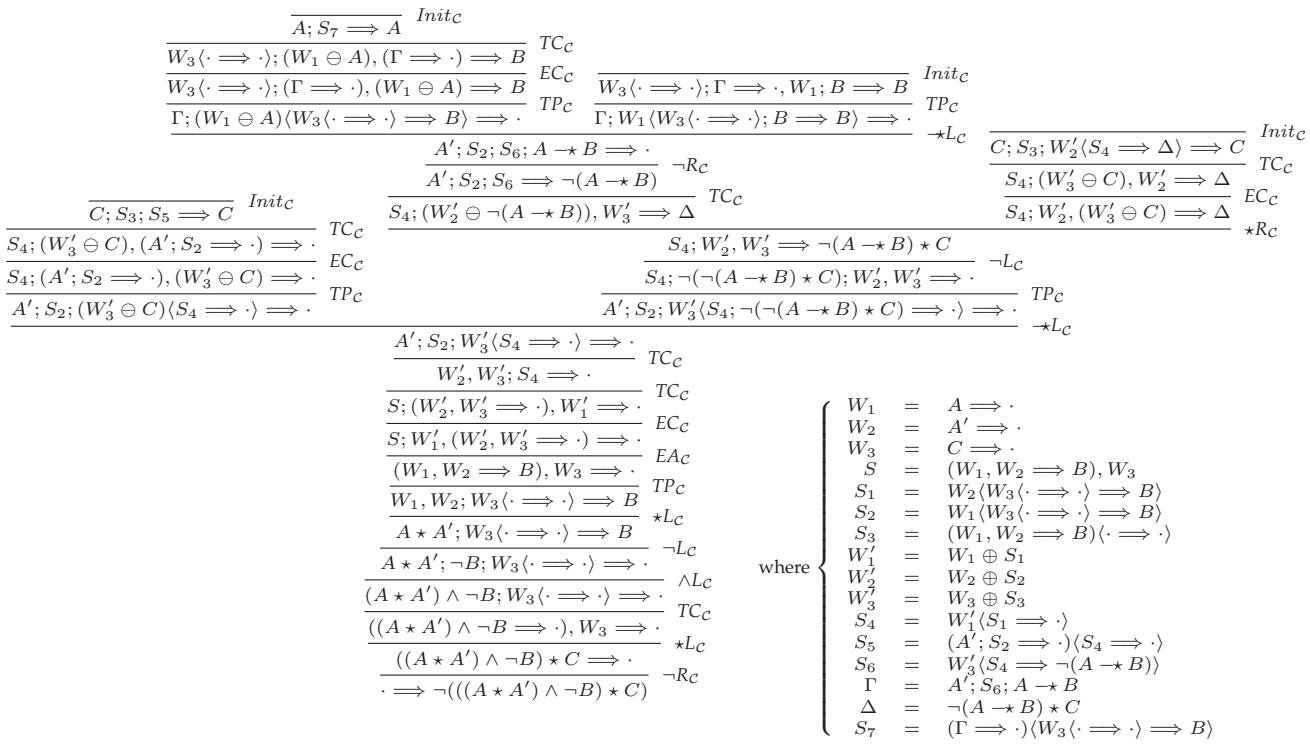


Figure 12: A proof of $\cdot \Longrightarrow \neg(((A \star A') \wedge \neg B) \star C)$ in $\mathbf{CS_{BBI}}$ where $A' = C \mathrel{-\!\!\star} \neg(\neg(A \mathrel{-\!\!\star} B) \star C)$. We define $\Gamma \Longrightarrow \Delta \ominus A$ as $\Gamma \Longrightarrow \Delta; A$.

Figure 15 shows an example of proving in $\mathbf{CS_{BBI}}$ the same sequent as in Figure 5. The proof tree is much smaller: the depth decreases from 16 to 8 and the number of applications of rules decreases from 18 to 12. Besides the amount of non-determinism in proof search is now minimal. In Figure 5, after applying the rule ${\emptyset_{\mathsf{m}}U_{\mathcal{S}}}^2$ (when read from the conclusion to the premise), we have to decide whether or not to duplicate S by applying the contraction rule $CL_{\mathcal{S}}$, and if we skip the rule $CL_{\mathcal{S}}$, proof search fails. In Figure 15, this form of non-determinism does not arise because the contraction rule is embedded into the rule $\emptyset_{\mathsf{m}}U_{\mathcal{C}}$. As a result, except for applying the rule $\emptyset_{\mathsf{m}}U_{\mathcal{C}}$ indefinitely, the only source of non-determinism concerns which of $A \star B$ and $A \star \neg B$ should be considered first by the rule $\star R_{\mathcal{C}}$, which is irrelevant for the purpose of this proof anyway.

Since their role is to change only the reference node without changing the graph structure of nodes, the traverse rule $TC_{\mathcal{C}}$ and the structural rule $EC_{\mathcal{C}}$ in Figure 15 do not increase the complexity of proof search. For example, once we decide to focus on $\neg B$ in $A; (A \Longrightarrow \Delta), (\emptyset_{\mathsf{m}} \Longrightarrow B; \neg B) \Longrightarrow \Delta$, we obtain a unique sequence of rules, namely $EC_{\mathcal{C}}$ followed by $TC_{\mathcal{C}}$, for exposing $\emptyset_{\mathsf{m}} \Longrightarrow B; \neg B$ in the reference node. Thus the cost of proof search is incurred mainly by various decisions on applying the structural and logical rules and not by applications of the traverse rules. Section 8.5 explains how to eliminate the traverse rules altogether in proof search.

8 Backward proof search in $\mathrm{CS_{BBI}}$

This section explains the design of our theorem prover which uses a backward search strategy built on top of $\mathbf{CS_{BBI}}$. Because of the undecidability of Boolean BI [31, 11], our theorem prover implements a semi-decision algorithm. For our purpose, a semi-decision algorithm is still useful because in program verification, we usually attempt to prove formulas that are believed to be true.

Our theorem prover includes a certifier which converts every proof in $\mathbf{CS_{BBI}}$ into an equivalent proof in $\mathbf{S_{BBI}}$ according to our proof of Theorem 7.2. In addition to automated proof search, it also supports an interactive mode, in which the user can issue various tactics to manually change the struc-

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset_{\mathsf{m}}; A; (S' \Longrightarrow B)\langle A \mathrel{-\!\!\star} B \Longrightarrow B\rangle \Longrightarrow A}\,\mathit{Init}_{\mathcal{C}}}{A \mathrel{-\!\!\star} B; (\emptyset_{\mathsf{m}}; A \Longrightarrow A), (S' \Longrightarrow B) \Longrightarrow B}\,\mathit{TC}_{\mathcal{C}}}{A \mathrel{-\!\!\star} B; (S' \Longrightarrow B), (\emptyset_{\mathsf{m}}; A \Longrightarrow A) \Longrightarrow B}\,\mathit{EC}_{\mathcal{C}}}{S'; (\emptyset_{\mathsf{m}}; A \Longrightarrow A)\langle A \mathrel{-\!\!\star} B \Longrightarrow B\rangle \Longrightarrow B}\,\mathit{TP}_{\mathcal{C}} \quad \dfrac{\dfrac{}{A \mathrel{-\!\!\star} B; B; (S' \Longrightarrow B), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow B}\,\mathit{Init}_{\mathcal{C}}}{S'; (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle A \mathrel{-\!\!\star} B; B \Longrightarrow B\rangle \Longrightarrow B}\,\mathit{TP}_{\mathcal{C}}}{\boxed{A \mathrel{-\!\!\star} B}; (\top \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot); \top; \boxed{S} \Longrightarrow B}\,{\mathrel{-\!\!\star}}L_{\mathcal{C}}}{A \mathrel{-\!\!\star} B; (\top \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow B}\,\emptyset_{\mathsf{m}}D_{\mathcal{C}}$$

where $\left\{ \begin{array}{lcl} S & = & (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle A \mathrel{-\!\!\star} B \Longrightarrow B\rangle \\ S' & = & A \mathrel{-\!\!\star} B; (\top \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot); \top \end{array} \right.$

Figure 13: A proof of $A \mathrel{-\!\!\star} B; (\top \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow B$ in $\mathbf{CS_{BBI}}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,\mathit{Init}_{\mathcal{S}}}{\emptyset_{\mathsf{m}}; A \Longrightarrow A}\,\mathit{WL}_{\mathcal{S}} \quad \dfrac{}{B \Longrightarrow B}\,\mathit{Init}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; B \Longrightarrow B}\,\mathit{WL}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; S \Longrightarrow \cdot}\,{\mathrel{-\!\!\star}}L_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; S \Longrightarrow B}\,\mathit{WR}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; \top; S \Longrightarrow B}\,\mathit{WL}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; (\top; S \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow B}\,\emptyset_{\mathsf{m}}D_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; (\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (\top; S \Longrightarrow \cdot) \Longrightarrow B}\,\mathit{EC}_{\mathcal{S}}}{\emptyset_{\mathsf{m}}; (\top; S \Longrightarrow \cdot)\langle A \mathrel{-\!\!\star} B \Longrightarrow B\rangle \Longrightarrow \cdot}\,\mathit{TP}_{\mathcal{S}}}{\emptyset_{\mathsf{m}}; A; (\top; S \Longrightarrow \cdot)\langle A \mathrel{-\!\!\star} B \Longrightarrow B\rangle \Longrightarrow \cdot}\,\mathit{WL}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot), (\top; S \Longrightarrow \cdot) \Longrightarrow B}\,\mathit{TC}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; (\top; S \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow B}\,\mathit{EC}_{\mathcal{S}}}{\top; S; S \Longrightarrow \cdot}\,\mathit{TP}_{\mathcal{S}}}{\top; S \Longrightarrow \cdot}\,\mathit{CL}_{\mathcal{S}}}{A \mathrel{-\!\!\star} B; (\top \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow B}\,\mathit{TC}_{\mathcal{S}}$$

where $S = (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot)\langle A \mathrel{-\!\!\star} B \Longrightarrow B\rangle$.

Figure 14: A proof of $A \mathrel{-\!\!\star} B; (\top \Longrightarrow \cdot), (\emptyset_{\mathsf{m}}; A \Longrightarrow \cdot) \Longrightarrow B$ in $\mathbf{S_{BBI}}$

ture of nodes. Both extensions are a preliminary step toward developing a program verification system based on full separation logic. For space reasons, we do not describe these extensions.

8.1 Basic structure of the proof search algorithm

Figure 16 shows the pseudocode for the proof search algorithm ProveBBI. Given a goal sequent W and a search depth d as input, it attempts to search for a proof tree for W with at most d applications of the structural rules (except the rule $\mathit{EC}_{\mathcal{C}}$) along any search path. First it examines every formula in a given sequent and applies the corresponding logical rule if possible (lines 1–12). After checking if $d = 0$ (line 13), it considers the structural rules $\mathit{EA}_{\mathcal{C}}$, $\emptyset_{\mathsf{m}}D_{\mathcal{C}}$, and $\emptyset_{\mathsf{m}}U_{\mathcal{C}}$ (lines 14–21). It returns either **true** or **fail**, depending on the result of the proof search.

The algorithm ProveBBI assumes that the rule $\mathit{EC}_{\mathcal{C}}$ is embedded into all the other rules. For example, the rule $\mathit{EA}_{\mathcal{C}}$ in Figure 16 indeed refers to one of the four instances obtained by independently applying the rule $\mathit{EC}_{\mathcal{C}}$ to node states W_1, W_2 and $(\Gamma'; W_1, W_2 \Longrightarrow \Delta'), W_3$. Similarly each of the rules $\star R_{\mathcal{C}}$, ${\mathrel{-\!\!\star}}L_{\mathcal{C}}$, and $\emptyset_{\mathsf{m}}D_{\mathcal{C}}$ has two symmetric instances. We also implicitly include the traverse rule derived in Section 3.3 (which promotes the sibling node as the reference node).

The first phase of the algorithm ProveBBI (lines 1–12) exploits the fact that all the logical rules, including $\star R_{\mathcal{C}}$ and ${\mathrel{-\!\!\star}}L_{\mathcal{C}}$, are invertible, *i.e.*, the premise implies the conclusion and vice versa. Hence it starts by applying the logical rules wherever possible until no more applications are left. For the rules $\star R_{\mathcal{C}}$ and ${\mathrel{-\!\!\star}}L_{\mathcal{C}}$, we make sure that they do not focus more than once on the same pair of a principal

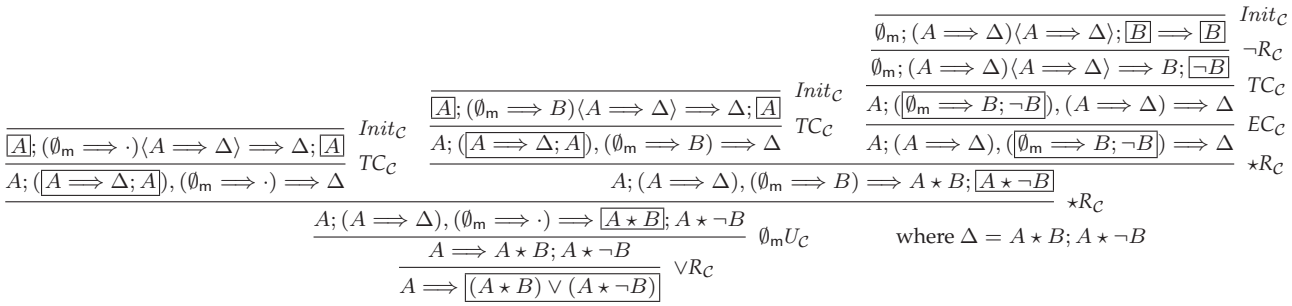


Figure 15: A proof of $A \Longrightarrow (A \star B) \vee (A \star \neg B)$ in $\mathbf{CS_{BBI}}$. Each rectangle marks a principle formula or a sequent to be exposed in the reference node.

formula and a node state (*e.g.*, a pair of $A \mathrel{-\!\!\star} B$ and $(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2)$ in the rule $\star R_{\mathcal{C}}$), since the principal formula survives in the premise. We can also skip every pair of a principal formula and a node state that yields essentially the same sequent as in the conclusion. For example, the following pattern of applying the rule $\star R_{\mathcal{C}}$ is unnecessary because the first sequent in the premise is logically equivalent to the conclusion:

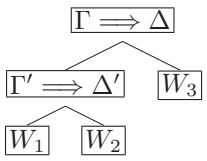
$$\frac{\Gamma; (\Gamma_1 \Longrightarrow \Delta_1; A; A), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B \quad \cdots}{\Gamma; (\Gamma_1 \Longrightarrow \Delta_1; A), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B}\ \star R_{\mathcal{C}}$$

After the first phase, the algorithm ProveBBI recursively invokes itself to apply the structural rules $EA_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, and $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$ in depth-first order (lines 14–21). For example, with $d = 2$, it considers all sequences of length 2 in the following order: $EA_{\mathcal{C}}$–$EA_{\mathcal{C}}$, $EA_{\mathcal{C}}$–$\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, $EA_{\mathcal{C}}$–$\emptyset_{\mathsf{m}} U_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$–$EA_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$–$\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$–$\emptyset_{\mathsf{m}} U_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$–$EA_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$–$\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$–$\emptyset_{\mathsf{m}} U_{\mathcal{C}}$. We choose to consider the rules $EA_{\mathcal{C}}$ and $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$ before the rule $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$, which is the least restrictive rule in the sense that it can be applied to any sequent. Note that applying a structural rule does not create new formulas but only gives rise to new pairs of a formula ($A \star B$ or $A \mathrel{-\!\!\star} B$) and a node state to which the rules $\star R_{\mathcal{C}}$ and $\mathrel{-\!\!\star} L_{\mathcal{C}}$ can be applied. A recursive invocation of ProveBBI immediately focuses on these new pairs during its first phase (lines 10–12).

8.2 Explosion in the search space and in the number of subgoals

While the algorithm ProveBBI eventually finds a proof tree for every provable sequent if given an unlimited search depth, a naive implementation suffers from two problems that are unique to Boolean BI. The first is an explosion in the search space in terms of the amount of conjunctive non-determinism among the logical rules $\star R_{\mathcal{C}}$ and $\mathrel{-\!\!\star} L_{\mathcal{C}}$ and the structural rules. That is, a typical proof search is quickly overwhelmed with too many choices for applying these rules, which are all invertible and thus can be applied aggressively. This problem is due to the structural rules $EA_{\mathcal{C}}$, $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, and $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$, each application of which immediately doubles or quadruples the search space. The second problem is an explosion in the number of subgoals due to the logical rules $\star R_{\mathcal{C}}$ and $\mathrel{-\!\!\star} L_{\mathcal{C}}$, each application of which increments the number of subgoals. Unlike the rule $\vee L_{\mathcal{C}}$ which can be applied to each formula $A \vee B$ at most once, these rules can be applied to each formula $A \star B$ or $A \mathrel{-\!\!\star} B$ as many times as there are corresponding node states. For example, if Γ contains n multiplicative pairs, the rule $\star R_{\mathcal{C}}$ creates 2^n subgoals from $\Gamma \Longrightarrow A \star B$ during the first phase of the algorithm ProveBBI.

The first problem is closely related to the contraction property built into the structural rules. As an example, consider the structural rule $EA_{\mathcal{C}}$ in Figure 10. The graph structure of its conclusion $\Gamma; (\Gamma'; (W_1, W_2) \Longrightarrow \Delta'), W_3 \Longrightarrow \Delta$ can be drawn as follows:



In comparison, the graph structure of the premise has four times more nodes than that of the conclusion because each sequent $W_i \oplus S_i$ ($i = 1, 2, 3$) represents the same graph structure as the conclusion, as

Algorithm ProveBBI(W, d)
Input: goal sequent W, search depth d
Output: **true** or **fail**

for each node w,
  promote w as the reference node by applying the traverse rules
  if $Init_{\mathcal{C}}$, $\bot L_{\mathcal{C}}$, or $\mathsf{I}R_{\mathcal{C}}$ is applicable, return **true**
  if $\neg L_{\mathcal{C}}$, $\neg R_{\mathcal{C}}$, $\vee R_{\mathcal{C}}$, $\mathsf{I}L_{\mathcal{C}}$, $\star L_{\mathcal{C}}$, or $\mathrel{-\!\!\star} R_{\mathcal{C}}$ is applicable,
    $W' \leftarrow$ result of applying the rule
    return ProveBBI(W', d)
  if $\vee L_{\mathcal{C}}$ is applicable,
    W_1 and $W_2 \leftarrow$ result of applying the rule
    return ProveBBI(W_1, d) && ProveBBI(W_2, d)
  if $\star R_{\mathcal{C}}$ or $\mathrel{-\!\!\star} L_{\mathcal{C}}$ is applicable to a fresh node state S,
    W_1 and $W_2 \leftarrow$ result of applying the rule to S
    return ProveBBI(W_1, d) && ProveBBI(W_2, d)
if $d = 0$, return **fail**
for each node state S in node w to which $EA_{\mathcal{C}}$ or $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$ is applicable,
  promote w as the reference node by applying the traverse rules
  $W' \leftarrow$ result of applying the rule to S
  if ProveBBI($W', d-1$) = **true**, return **true**
for each node w,
  promote w as the reference node by applying the traverse rules
  $W' \leftarrow$ result of applying $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$
  if ProveBBI($W', d-1$) = **true**, return **true**
return **fail**

Figure 16: Pseudocode for the proof search algorithm ProveBBI

shown in Figure 17. Consequently the premise provides four times more ways to apply the rules of $\mathbf{CS_{BBI}}$, thus quadrupling the search space. In a similar way, each application of the other structural rules $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$ and $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$ doubles the search space. We remark that the first problem is *not* the price to pay for building contraction into the structural rules, since the same problem of search space explosion remains even if we do not build contraction into the structural rules.

The second problem itself is orthogonal to the first problem, but its effect is heavily exacerbated by the first problem. As an example, consider again the structural rule $EA_{\mathcal{C}}$ in Figure 10 where we set $\Delta = A \star (B \star C)$. After an application of the rule $EA_{\mathcal{C}}$ to obtain the graph structure in Figure 17, two applications of the rule $\star R_{\mathcal{C}}$ ensue to copy A to W_1, B to W_2, and C to W_3. If these formulas happen to involve multiplicative connectives, we can again apply the rules $\star R_{\mathcal{C}}$ and $\mathrel{-\!\!\star} L_{\mathcal{C}}$ to propagate their component formulas, which, in turn, may trigger further applications of the rules $\star R_{\mathcal{C}}$ and $\mathrel{-\!\!\star} L_{\mathcal{C}}$, and so on.

To alleviate the first problem, we need to devise a scheme for prioritizing applications of the structural rules (Section 8.3). We can solve the second problem by borrowing an idea from the inverse method (Section 8.4). In addition, we can eliminate the traverse rules altogether (Section 8.5).

8.3 Prioritizing applications of the structural rules

As a solution to the first problem, we assign a priority, either high or low, to every sibling relation between nodes so as to prioritize all applications of the rules $EA_{\mathcal{C}}$ and $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$, and to every node itself so as to prioritize all applications of the rule $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$. When applying a structural rule, the algorithm ProveBBI first considers sibling relations and nodes with a high priority and then those with a low priority. Below we explain how to assign priorities to sibling relations and how to determine priorities for nodes.

For the rule $EA_{\mathcal{C}}$ as shown in Figure 10, we assign priorities to sibling relations in the premise according to Figure 17 with the following interpretation:

- For a sequent inside a rectangle $\boxed{W}$, every sibling relation in it is assigned the same priority as

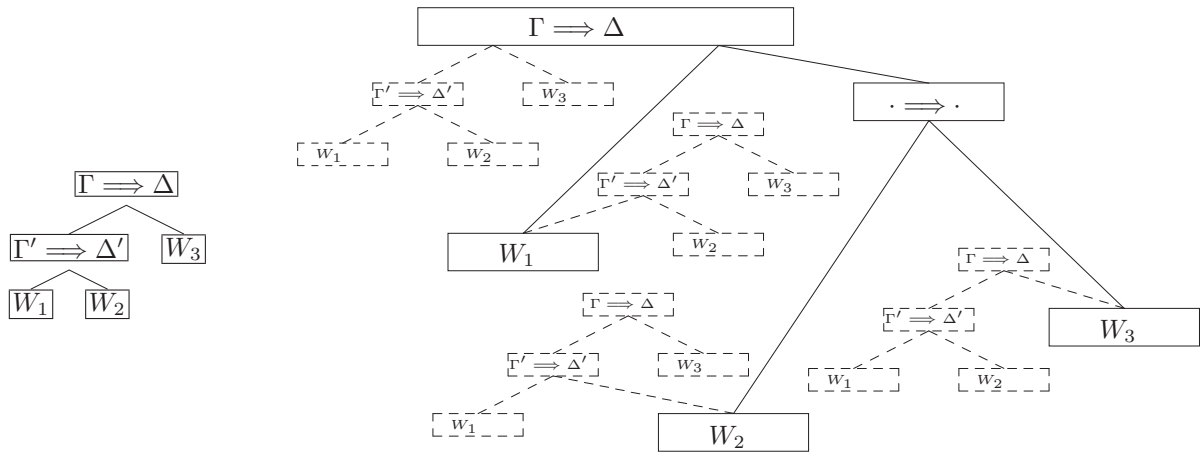


Figure 17: The graph structure of nodes before (conclusion) and after (premise) applying the structural rule $EA_{\mathcal{C}}$ in Figure 10

in the conclusion.

- For a sequent inside a dashed rectangle $\boxed{W}$, every sibling relation in it is assigned a low priority.
- A sibling relation depicted with solid lines $\diagup\!\diagdown$ is assigned a high priority.
- A sibling relation depicted with dashed lines $\cdot\!\cdot\!\cdot$ is assigned a low priority.

The rationale for this assignment is that an application of the rule $EA_{\mathcal{C}}$ is primarily intended to generate sibling relations described by $W_1, (W_2, W_3 \Longrightarrow \cdot)$, rather than S_1, S_2, and S_3 in $W_1 \oplus S_1, (W_2 \oplus S_2, W_3 \oplus S_3 \Longrightarrow \cdot)$. When applying the rule $EA_{\mathcal{C}}$, the algorithm ProveBBI focuses first on those node states both of whose sibling relations have a high priority. In a similar way, we assign priorities to sibling relations in the premise of the rules $\emptyset_{\mathsf{m}}D_{\mathcal{C}}$ and $\emptyset_{\mathsf{m}}U_{\mathcal{C}}$ according to Figure 18.

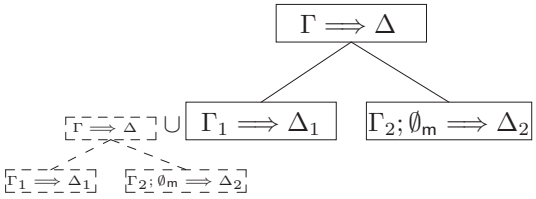
We determine priorities of nodes by analyzing priorities of sibling relations. If a node is involved in a sibling relation with a high priority, it is more likely to be under active consideration by other structural rules than those nodes with no such involvement. Hence we assign a high priority to every node involved in at least one such sibling relation. For the logical rules $\star L_{\mathcal{C}}$ and $\mathrel{-\!\!\star} R_{\mathcal{C}}$, we reuse the priority assigned to the reference node of the conclusion for two new nodes in the premise.

Now we redesign the algorithm ProveBBI as a two-stage algorithm. In the first stage, it applies the structural rules using only sibling relations and nodes with a high priority. Note that it still generates every node with a low priority, which is never used by the structural rules, but may be needed by the logical rules. For example, the two sequents discussed in Section 7 are provable in the first stage precisely because we also generate every node with a low priority. If the proof search fails, it enters the second stage and repeats the proof search without ignoring sibling relations and nodes with a low priority. The second stage is necessary for the completeness of proof search, since some formula requires us to apply the structural rules using those with a low priority as well. In fact, we can find even a formula whose proof tree applies the structural rules using *only* those with a low priority. Section 8.6 presents examples of such formulas.

8.4 Reusing the proof tree from the premise

We solve the problem of an explosion in the number of subgoals with a simple technique of reusing the proof tree from the premise. Suppose that we apply the rule $\star R_{\mathcal{C}}$ or $\mathrel{-\!\!\star} L_{\mathcal{C}}$ to produce two subgoals in the premise, without knowing whether this application is necessary or not. If this application is unnecessary, however, every proof tree for the first premise must be a proof tree for the conclusion as well. Hence, upon finding a proof tree for the first subgoal, we "replay" it against the conclusion, and attempt to prove the second subgoal only in the case of a failure. In this way, we can aggressively apply the rules $\star R_{\mathcal{C}}$ and $\mathrel{-\!\!\star} L_{\mathcal{C}}$ without worrying about an explosion in the number of subgoals. In essence, we

After applying the rule $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$:



After applying the rule $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$:

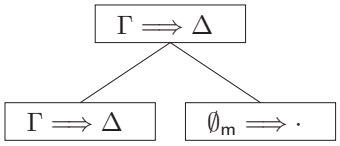


Figure 18: Assigning priorities to sibling relations after applying the rules $\emptyset_{\mathsf{m}} D_{\mathcal{C}}$ and $\emptyset_{\mathsf{m}} U_{\mathcal{C}}$. $(\Gamma \Longrightarrow \Delta) \cup (\Gamma' \Longrightarrow \Delta')$ is defined as $\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$.

partially simulate the inverse method with a moderate overhead of revisiting proof trees (but without entirely reformulating the nested sequent calculus $\mathbf{CS_{BBI}}$).

8.5 Eliminating the traverse rules

The algorithm ProveBBI invokes the traverse rules to change the reference node (lines 2, 15, 19 in Figure 16), but we can eliminate the traverse rules altogether with a slight change in the representation of sequents. The basic observation is that the traverse rules change only the reference node without altering the graph structure of nodes. Hence, by rewriting every rule in such a way that it directly focuses on any formula or node without requiring a reference node, we can discard the traverse rules.

To this end, we introduce a labelled sequent which assigns a unique label w to every node and annotates all formulas and $\emptyset_{\mathsf{m}}$'s in it with w:

$$\begin{array}{rrcl} \text{labelled sequent} & L & = & \Xi \vdash \Sigma \Longrightarrow \Pi \\ \text{graph structure} & \Xi & ::= & \cdot \mid \Xi, w \sim w_1 \cdot w_2 \\ \text{labelled truth context} & \Sigma & ::= & \cdot \mid \Sigma, A@w \mid \Sigma, \emptyset_{\mathsf{m}}@w \\ \text{labelled falsehood context} & \Pi & ::= & \cdot \mid \Pi, A@w \end{array}$$

$w \sim w_1 \cdot w_2$ in the graph structure specifies that w is a parent node of w_1 and w_2. Then we can convert every sequent to a *unique* labelled sequent modulo renaming labels because a sequent determines a unique graph structure of nodes where each node contains a unique set of true formulas, $\emptyset_{\mathsf{m}}$'s, and false formulas. Let us write $[\![W]\!]$ for the unique labelled sequent converted from sequent W. For a rule deducing W from W' (and W'') in $\mathbf{CS_{BBI}}$, we derive a new rule that deduces $[\![W]\!]$ from $[\![W']\!]$ (and $[\![W'']\!]$) in a single step; for an axiom deducing W in $\mathbf{CS_{BBI}}$, we derive a new axiom deducing $[\![W]\!]$. We refer to the resultant system as the labelled $\mathbf{CS_{BBI}}$:

$$\frac{W'}{W} R \ \text{ in } \mathbf{CS_{BBI}} \quad \longrightarrow \quad \frac{[\![W']\!]}{[\![W]\!]} R_{\mathcal{L}} \ \text{ in the labelled } \mathbf{CS_{BBI}}$$

The labelled $\mathbf{CS_{BBI}}$ has no traverse rules because the premise and conclusion of a traverse rule in $\mathbf{CS_{BBI}}$ represent the same graph structure of nodes. Still it is equivalent to $\mathbf{CS_{BBI}}$ because the definition of labelled sequents embeds the traverse rules into all the inference rules:

Proposition 8.1. *W is provable in $\mathbf{CS_{BBI}}$ if and only if $[\![W]\!]$ is provable in the labelled $\mathbf{CS_{BBI}}$.*

Proof. The proof proceeds by induction on the structure of the proof of W and $[\![W]\!]$. The base case, in which we prove W or $[\![W]\!]$ by an axiom, is obvious because each axiom in $\mathbf{CS_{BBI}}$ has a corresponding axiom in the labelled $\mathbf{CS_{BBI}}$.

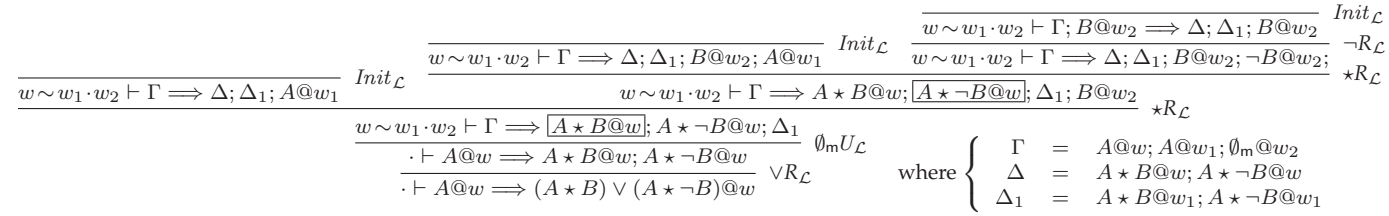


Figure 19: A proof of $\cdot \vdash A@w \Longrightarrow (A \star B) \vee (A \star \neg B)@w$ in the labelled $\mathbf{CS_{BBI}}$. The rule $\star R_{\mathcal{L}}$ focuses on the formula inside the rectangle.

Consider a proof $\mathcal{D} = \dfrac{\vdots}{\dfrac{W'}{W}\ R}$ in $\mathbf{CS_{BBI}}$. By induction hypothesis, we have a proof of $[\![W']\!]$ in the labelled $\mathbf{CS_{BBI}}$. If R is a traverse rule, we have a proof of $[\![W]\!]$ because $[\![W']\!] = [\![W]\!]$. Otherwise, because of the way that the labelled $\mathbf{CS_{BBI}}$ is designed, there is a rule $R_{\mathcal{L}}$ that deduces $[\![W]\!]$ from $[\![W']\!]$ and we obtain a proof of $[\![W]\!]$ in the labelled $\mathbf{CS_{BBI}}$.

Consider a proof of $\mathcal{E} = \dfrac{\vdots}{\dfrac{L'}{[\![W]\!]}\ R_{\mathcal{L}}}$ in the labelled $\mathbf{CS_{BBI}}$. Because of the way that the labelled $\mathbf{CS_{BBI}}$ is designed, there is a sequent W' such that $L' = [\![W']\!]$. Moreover $\mathbf{CS_{BBI}}$ has an inference rule $\dfrac{W'_0}{W_0}\ R$ such that $[\![W_0]\!] = [\![W]\!]$ and $[\![W'_0]\!] = [\![W']\!]$, which implies that we can deduce W from W_0 and W'_0 from W' by applying the traverse rules and the rule $EC_{\mathcal{C}}$. By induction hypothesis, we have a proof of W' in $\mathbf{CS_{BBI}}$. Then we can obtain a proof of W as follows:

$$\dfrac{\dfrac{\dfrac{W'}{\vdots}}{\dfrac{W'_0}{W_0}\ R}}{\dfrac{\vdots}{W}}$$

□

Figure 19 shows an example of proving in the labelled $\mathbf{CS_{BBI}}$ the same formula as in Figure 15. The depth decreases from 8 to 6 and the number of applications of rules decreases from 12 to 8. The rule $Init_{\mathcal{L}}$ immediately completes the proof when it detects the same labelled formula in both contexts of a given labelled sequent. The rule $\neg R_{\mathcal{L}}$ also directly focuses on $\neg B@w_2$, regardless of the presence of $w \sim w_1 \cdot w_2$ in the graph structure. In this way, the labelled $\mathbf{CS_{BBI}}$ dispenses with the traverse rules, yielding a smaller proof tree than $\mathbf{CS_{BBI}}$.

8.6 Experimental results

We compare a naive implementation of the algorithm ProveBBI with an optimized implementation that incorporates those ideas described in Sections 8.3 and 8.4. Both implementations internally use the labelled $\mathbf{CS_{BBI}}$ to eliminate the traverse rules, as explained in Section 8.5. Our implementations are written in Objective CAML and run on Ubuntu Linux 11.10 with Intel Core i7-960 3.2GHz and 6 gigabytes of main memory.

Figure 20 shows results of running both implementations (*naive* and *optimized*) on 14 representative formulas. For a given formula A, we use sequent $\cdot \Longrightarrow A$ and search depth d as input to the algorithm ProveBBI. Except for experiment (c), we set d to the minimum search depth for finding a proof tree. The result is either the return value of ProveBBI (**true** and **fail**) or **error** if the proof search does not terminate

	formula	d	*naive*		*optimized*		
			result	cost	result	cost	time
(a)	$(A \mathrel{-\!\!\star} B) \wedge (\top \star (\mathsf{I} \wedge A)) \to B$	1	**true**	9	**true**	9	0.001
	$(\mathsf{I} \mathrel{-\!\!\star} \neg(\neg A \star \mathsf{I})) \to A$	1	**true**	9	**true**	9	0.001
	$\neg((A \mathrel{-\!\!\star} \neg(A \star B)) \wedge ((\neg A \mathrel{-\!\!\star} \neg B) \wedge B))$	1	**true**	26	**true**	19	0.001
	$\mathsf{I} \to (A \mathrel{-\!\!\star} B \mathrel{-\!\!\star} C) \mathrel{-\!\!\star} A \star B \mathrel{-\!\!\star} C$	2	**true**	700	**true**	76	0.021
	$\mathsf{I} \to A \star (B \star C) \mathrel{-\!\!\star} A \star B \star C$	2	**true**	2606	**true**	263	0.051
	$\mathsf{I} \to A \star ((B_1 \mathrel{-\!\!\star} B_2) \star C) \mathrel{-\!\!\star} A \star (B_1 \mathrel{-\!\!\star} B_2) \star C$	2	**true**	12317	**true**	336	0.102
♯	$\neg((A \mathrel{-\!\!\star} \neg(\neg(D \mathrel{-\!\!\star} \neg(A \star (C \star B))) \star A)) \wedge C \star (D \wedge (A \star B)))$	2	**true**	121000	**true**	380	0.053
	$\neg((C_1 \star (C_2 \star C_3)) \wedge ((A \mathrel{-\!\!\star} \neg(\neg(B \mathrel{-\!\!\star} \neg(C_2 \star (C_3 \star C_1))) \star A)) \star (B \wedge (A \star \top))))$	2	**true**	1614053	**true**	43	0.023
♯	$\neg((A \mathrel{-\!\!\star} \neg(\neg(D \mathrel{-\!\!\star} \neg(C \star B_2 \star (B_1 \star A))) \star A)) \wedge C \star (D \wedge (A \star (B_1 \star B_2))))$	3	**error**	-	**true**	74856	128.4
(b)	$A \star (B \star (C \star D)) \to D \star (C \star (B \star A))$	2	**true**	2618	**true**	56	0.012
	$A \star (B \star (C \star D)) \to D \star (B \star (C \star A))$	3	**true**	15686	**true**	34	0.041
	$A \star (B \star (C \star (D \star E))) \to E \star (D \star (A \star (B \star C)))$	3	**error**	-	**true**	2320	2.2
	$A \star (B \star (C \star (D \star E))) \to E \star (B \star (A \star (C \star D)))$	4	**error**	-	**true**	2921	28.2
(c)	$\mathsf{I} \to A \star ((B_1 \mathrel{-\!\!\star} B_2) \star (C \star D)) \mathrel{-\!\!\star} A \star D \star (C \star (B_1 \mathrel{-\!\!\star} B_2))$	2	**fail**	1795	**fail**	1837	0.438
		3	**error**	-	**true**	5942	8.2
		4	**error**	-	**true**	181	7.0
		5	**error**	-	**true**	182	127.0
		6	**error**	-	**error**	-	-

Figure 20: Results of running two implementations (*naive* and *optimized*) of the algorithm ProveBBI. The elapsed time is in seconds.

within 10 minutes. In measuring the cost in terms of the number of applications of the rules, we exclude the rule $EC_{\mathcal{C}}$ which is already embedded into all the other rules. The elapsed time is in seconds.

In experiment (a), we test nine formulas (all involving multiplicative connectives) of increasing complexity. We observe that the optimized implementation is much less susceptible to the exponential growth of the search space than the naive implementation, thereby demonstrating that the two optimizations in Sections 8.3 and 8.4 are indeed highly effective. The two formulas marked ♯ require only those applications of the rule $EA_{\mathcal{C}}$ in which sibling relations with a low priority are visited; hence the proof search finishes in the second stage of the algorithm ProveBBI.

A similar pattern is observed in experiment (b) which is designed to measure the effectiveness of the two optimizations specifically against the rule $EA_{\mathcal{C}}$. Ideally each formula requires just as many applications of the rule $EA_{\mathcal{C}}$ as search depth d (because of the way that we specify d) as well as a few more applications of the logical rules, but the actual cost is much higher because of the side effect of applying the rule $EA_{\mathcal{C}}$.

Experiment (c) tests the effect of increasing d for a common formula which can be proven with two applications of the rule $EA_{\mathcal{C}}$ followed by an application of the rule $\emptyset_{\mathsf{m}}D_{\mathcal{C}}$. A search depth of 3 eventually produces such an ideal proof tree, but only after a number of wrong applications of the rule $EA_{\mathcal{C}}$. An increase of d to 4, however, immediately incurs a wrong application of the rule $EA_{\mathcal{C}}$ which happens to lead to the correct two applications of the rule $EA_{\mathcal{C}}$, which is why it produces a proof tree at a much lower cost (from 5942 to 181). A further increase of d to 5 does not significantly increase the cost, but the elapsed time becomes much longer because of the extra overhead of manipulating much larger sequents.

Overall we find that the optimized implementation is reasonably fast in proving typical formulas of Boolean BI.

9 Related work

Because of its support for local reasoning and potential in large-scale program verification, separation logic has attracted many researchers to develop automated verification tools based on it and to embed it in existing proof assistants. We first review previous research along these lines, and then discuss related proof theory.

9.1 Automated verification tools based on separation logic

Separation logic has been the basis for a number of automated verification tools targeting programs using mutable data structures. The first such tool is Smallfoot by Berdine *et al*. [6] which aims to test

the feasibility of automated verification using separation logic. To achieve full automation, it permits no pointer arithmetic and verifies only shape properties of linked lists and trees. Space Invader by Distefano *et al*. [16] permits pointer arithmetic by integrating the abstract interpretation method into the symbolic execution method in [7]. THOR by Magill *et al*. [32] is an extension of Space Invader which is capable of tracking the length of linked lists. SLAyer by Berdine *et al*. [4] is another extension of Space Invader which uses higher-order predicates to express common properties of nodes in linked lists. The use of higher-order predicates enables SLAyer to verify shape properties of composite linked lists such as linked lists of circular linked lists.

There are also several tools supporting arbitrary data structures. HIP by Nguyen and Chin [38] allows users to specify invariants on arbitrary data structures in terms of inductive predicates. Since checking these invariants usually relies on basic properties of inductive predicates that are easy to prove but difficult to discover automatically, HIP requires users to explicitly state such properties in the form of lemmas, which are automatically proven and then applied as necessary. Similarly to HIP, VeriFast by Jacobs *et al*. [28] relies on user-supplied inductive predicates and lemmas. Unlike HIP, however, VeriFast requires users to provide proofs of these lemmas and specify when to apply them. jStar by Distefano and Parkinson [17] is an extension of Space Invader which exploits user-supplied abstraction rules in order to support arbitrary data structures. Its distinguishing feature is the ability to infer loop invariants automatically. Xisa by Chang and Rival [14] takes a different approach by indirectly specifying invariants on data structures with validation code. Xisa analyzes validation code to extract inductive predicates for describing invariants as well as lemmas for describing their basic properties. Since validation code can be written in common programming languages, users of Xisa do not need the expertise to specify invariants of interest in terms of inductive predicates.

All these tools use as their logical foundation not full separation logic but only its decidable fragment by Berdine *et al*. [5], which does not include separating implication $-\!\!\star$. As shown by Ishtiaq and O'Hearn [27], lack of separating implication implies no support for backward reasoning by weakest precondition generation for those programs performing heap mutation or allocation. As a result, these tools allow only forward reasoning based on symbolic execution as in [7] and do not demonstrate the full potential of separation logic in program verification.

9.2 Embedding separation logic in proof assistants

Instead of developing fully automated verification tools, several researchers embed separation logic in existing proof assistants. Weber [43] embeds separation logic in Isabelle by modeling heaps as partial functions from addresses to values. Tuch *et al*. [42] present another embedding of separation logic in Isabelle which models heaps as arrays of bytes in order to support C-like byte operations. Marti *et al*. [33] embed separation logic in Coq by using abstract data types to model heaps, and use this embedding to verify the correctness of the heap manager of an operating system. Based on full separation logic, these approaches offer a higher level of expressivity than automated verification tools, but at the expense of automation.

There are several attempts to develop a set of automation tactics in order to reduce the burden of writing manual proofs when using embeddings of separation logic. Appel [1] develops a set of automation tactics for the embedding in [33] and verifies the correctness of an in-place list reversal procedure. McCreight [34] proposes another set of automation tactics and verifies the correctness of Cheney's copying garbage collector. Ynot by Nanevski *et al*. [35] also embeds separation logic in Coq, but takes a different approach by integrating it into the type system of Coq. The basic idea of Ynot is to extend Coq with monadic types for effectful computations and write imperative programs whose type carries their own specifications. Chlipala *et al*. [15] develop a variant of Ynot with automation tactics which allows users to omit low-level proof steps, such as choosing which branch of a disjunction to prove, and to focus on high-level proof steps such as applying induction. Nanevski *et al*. [36] reformulate the Hoare Type Theory in [35] with a new heap model satisfying the algebraic properties of partial commutative monoids, and show that the proper use of these properties reduces the number of proof obligations and also keeps their proofs short.

9.3 Proof search in the logic of BI and separation logic

Previous work on proof search in the logic of BI mainly focuses on intuitionistic BI, which is another member in the family that inherits multiplicative connectives from intuitionistic linear logic (like Boolean BI), but additive connectives from intuitionistic propositional logic. Galmiche and Méry [20, 21] present a labelled tableau calculus for a propositional fragment without $\bot$ and develop a theorem prover, called BILL, on top of it. Their later paper [23] extends the calculus for full intuitionistic BI. Donnelly *et al.* [18] investigate the inverse method for a propositional fragment without units ($\top$, $\bot$, and I) and develop a forward theorem prover.

For Boolean BI, no theorem prover has been developed yet because of the lack of a proof theory suitable for proof search. Larchey-Wendling and Gamliche [30] formulate a labelled tableau calculus by extending the labelled tableau calculus for intuitionistic BI in [23], but only in order to investigate the relation between intuitionistic BI and Boolean BI. Brotherston [10] shows that a modular combination of display calculi for classical logic and intuitionistic linear logic gives rise to a display calculus $\mathbf{DL_{BBI}}$ for Boolean BI, the first cut-free syntactic formulation of Boolean BI, and proves the cut elimination property by observing that its rules obey all the syntactic constraints given in [2]. Developing a practical proof search strategy on top of it, however, is far from easy because of the complexity due to its display rules and the difficulty in restricting applications of the contraction rules [9].

Galmiche and Méry [22] present a labelled tableau calculus for separation logic. It lies somewhere between syntactic (tableau) and semantic (labelled) formulations because labels correspond to heaps in separation logic. Their calculus, albeit sound and complete, does not directly translate to a proof search strategy in its current form. It is easy to build a tableau for a given formula according to the calculus, but one needs to check if all branches in the tableau are logically or structurally inconsistent. This requires two semantic functions (a measure and an interpretation) for each branch, and the calculus does not specify how to obtain such semantic functions. For a similar reason, the labelled tableau calculus for Boolean BI in [30] does not directly translate to a proof search strategy. For theorem provers for the decidable fragment of separation logic by Berdine *et al*. [5] (without separating implication), see, for example, [6, 17, 37].

9.4 Nested sequent calculi

A nested sequent calculus is one whose sequent may contain smaller sequents. It has been used as a proof-theoretic formulation of some modal and tense logics [29, 12] for which no sequent calculus of the standard form exists. $\mathbf{S_{BBI}}$ is also a nested sequent calculus because a sequent may contain smaller sequents.

A nested sequent calculus is often obtained as an optimization of an equivalent display calculus that is not simplified to a sequent calculus of the standard form. Gore *et al.* [24, 25, 26] propose such nested sequent calculi for bi-intuitionistic logic and classical tense logic. In particular, their nested sequent calculus $\mathbf{SKt}$ for classical tense logic is similar to $\mathbf{S_{BBI}}$ in that it has two residual rules corresponding to the traverse rules of $\mathbf{S_{BBI}}$. The main difference is that $\mathbf{SKt}$ uses only a tree structure of sequents and has no rule for associativity (which changes parent-child and sibling relations between sequents). Hence it is much easier to embed contraction rules in $\mathbf{SKt}$ than in $\mathbf{S_{BBI}}$ and the problem of search space explosion due to structural rules as in $\mathbf{CS_{BBI}}$ does not exist in $\mathbf{SKt}$.

9.5 Comparison between $\mathbf{S_{BBI}}$ and $\mathbf{DL_{BBI}}$

We have seen in Section 6.3 that for Boolean BI, sequents in $\mathbf{S_{BBI}}$ essentially represent a normal form of consecutions in $\mathbf{DL_{BBI}}$. For intuitionistic BI, Brotherston [10] establishes a stronger result that sequents in its sequent calculus are literally a normal form of consecutions in its display calculus and thus belong to the same syntactic category. He also conjectures that a cut-free sequent calculus for Boolean BI is unlikely to exist if it has no negative structural connective such as $\sharp$ in $\mathbf{DL_{BBI}}$ (see Section 5 in [10]). Our discovery of $\mathbf{S_{BBI}}$ does *not* contradict his conjecture because we can think of sequents in $\mathbf{S_{BBI}}$ as implicitly applying a negative structural connective to falsehood contexts.

What is equally important, however, is that introducing only a negative structural connective is not enough to achieve a cut-free sequent calculus for Boolean BI, which must use a graph structure of sequents in which a sequent may have multiple parent sequents. In the case of $\mathbf{DL_{BBI}}$, the linear

structural connective $-\!\circ$ allows such a graph structure of sequents, but it is not clear whether it is such a graph structure that $-\!\circ$ is originally intended to express. For example, even though it does not require such a graph structure, the display calculus for intuitionistic BI in [10] also has the same linear structural connective $-\!\circ$, which is in fact from the display calculus for linear logic [3]. In contrast, $\mathbf{S_{BBI}}$ introduces an adjoint pair $W\langle W'\rangle$ for the sole purpose of allowing such a graph structure of sequents.

10 Conclusion

Despite its close connection with separation logic, Boolean BI has not received much attention from the research community. Such a lack of research, which is quite unusual considering the status of separation logic in the field of program verification, is perhaps due to the difficulty of finding a proof theory suitable for theorem proving. Our nested sequent calculus $\mathbf{S_{BBI}}$ as well as a theorem prover based on it may serve as a test bed for developing proof search strategies for Boolean BI. In particular, its use of nested sequents allowing multiple parent sequents may shed new light on how to deal with separating implication in a theorem prover for separation logic.

References

[1] Andrew W. Appel. Tactics for separation logic. http://www.cs.princeton.edu/~appel/papers/septacs.pdf, 2006.

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

[3] Nuel Belnap. Linear logic displayed. *Notre Dame Journal of Formal Logic*, 31:14–25, 1990.

[4] Josh Berdine, Cristiano Calcagno, Byron Cook, Dino Distefano, Peter W. O'Hearn, Thomas Wies, and Hongseok Yang. Shape analysis for composite data structures. In *Proc. CAV*, pages 178–192, 2007.

[5] Josh Berdine, Cristiano Calcagno, and Peter W. O'Hearn. A decidable fragment of separation logic. In *Proc. FSTTCS*, pages 97–109, 2004.

[6] Josh Berdine, Cristiano Calcagno, and Peter W. O'Hearn. Smallfoot: Modular automatic assertion checking with separation logic. In *Proc. FMCO*, pages 115–137, 2005.

[7] Josh Berdine, Cristiano Calcagno, and Peter W. O'Hearn. Symbolic execution with separation logic. In *Proc. APLAS*, pages 52–68, 2005.

[8] Lars Birkedal, Noah Torp-Smith, and John C. Reynolds. Local reasoning about a copying garbage collector. In *Proc. POPL*, pages 220–231, 2004.

[9] James Brotherston. A cut-free proof theory for boolean BI (via display logic). Technical Report DTR09-13, Imperial College London, 2009.

[10] James Brotherston. A unified display proof theory for bunched logic. In *Proc. MFPS*, pages 197–211, 2010.

[11] James Brotherston and Max Kanovich. Undecidability of propositional separation logic and its neighbours. In *Proc. LICS*, pages 130–139, 2010.

[12] Kai Brünnler. Deep sequent systems for modal logic. In *Proc. Advances in Modal Logic*, pages 107–119, 2006.

[13] Cristiano Calcagno, Dino Distefano, Peter W. O'Hearn, and Hongseok Yang. Beyond reachability: Shape abstraction in the presence of pointer arithmetic. In *Proceeedings of the 13th International Static Analysis Symposium*, pages 182–203, 2006.

[14] Bor-Yuh Evan Chang and Xavier Rival. Relational inductive shape analysis. In *Proc. POPL*, pages 247–260, 2008.

[15] Adam Chlipala, Gregory Malecha, Greg Morrisett, Avraham Shinnar, and Ryan Wisnesky. Effective interactive proofs for higher-order imperative programs. In *Proc. ICFP*, pages 79–90, 2009.

[16] Dino Distefano, Peter W. O'Hearn, and Hongseok Yang. A local shape analysis based on separation logic. In *Proc. TACAS*, pages 287–302, 2006.

[17] Dino Distefano and Matthew J. Parkinson. jStar: towards practical verification for Java. In *Proc. OOPSLA*, pages 213–226, 2008.

[18] Kevin Donnelly, Tyler Gibson, Neel Krishnaswami, Stephen Magill, and Sungwoo Park. The inverse method for the logic of bunched implications. In *Proc. LPAR*, pages 466–480, 2004.

[19] Didier Galmiche and Dominique Larchey-Wendling. Expressivity properties of boolean BI through relational models. In *Proc. FSTTCS*, pages 357–368, 2006.

[20] Didier Galmiche and Daniel Méry. Proof-search and countermodel generation in propositional BI logic. In *Proc. TACS*, pages 263–282, 2001.

[21] Didier Galmiche and Daniel Méry. Semantic labelled tableaux for propositional BI (without bottom). *Journal of Logic and Computation*, 13:70–753, 2003.

[22] Didier Galmiche and Daniel Méry. Tableaux and resource graphs for separation logic. *Journal of Logic and Computation*, 20:189–231, 2010.

[23] Didier Galmiche, Daniel Méry, and David J. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, 15:1033–1088, 2005.

[24] Rajeev Goré, Linda Postniece, and Alwen Tiu. Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents. In *Proc. Advances in Modal Logic*, pages 43–66, 2008.

[25] Rajeev Goré, Linda Postniece, and Alwen Tiu. Taming displayed tense logics using nested sequents with deep inference. In *Proc. TABLEAUX*, pages 189–204, 2009.

[26] Rajeev Goré, Linda Postniece, and Alwen Tiu. On the correspondence between display postulates and deep inference in nested sequent calculi for tense logics. *Logical Methods in Computer Science*, 7:1–38, 2011.

[27] Samin S. Ishtiaq and Peter W. O'Hearn. BI as an assertion language for mutable data structures. In *Proc. POPL*, pages 14–26, 2001.

[28] Bart Jacobs, Jan Smans, and Frank Piessens. VeriFast: Imperative programs as proofs. In *Proc. VSTTE*, pages 59–68, 2010.

[29] Ryo Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53(1):119–136, 1994.

[30] Dominique Larchey-Wendling and Didier Galmiche. Exploring the relation between intuitionistic BI and boolean BI: an unexpected embedding. *Mathematical Structures in Computer Science*, 19:435–500, 2009.

[31] Dominique Larchey-Wendling and Didier Galmiche. The undecidability of boolean BI through phase semantics. In *Proc. LICS*, pages 140–149, 2010.

[32] Stephen Magill, Josh Berdine, Edmund M. Clarke, and Byron Cook. Arithmetic strengthening for shape analysis. In *Proc. SAS*, pages 419–436, 2007.

[33] Nicolas Marti, Reynald Affeldt, and Akinori Yonezawa. Formal verification of the heap manager of an operating system using separation logic. In *Proc. ICFEM*, pages 400–419, 2006.

[34] Andrew McCreight. Practical tactics for separation logic. In *Proc. TPHOLs*, pages 343–358, 2009.

[35] Aleksandar Nanevski, Greg Morrisett, Avraham Shinnar, Paul Govereau, and Lars Birkedal. Ynot: Dependent types for imperative programs. In *Proc. ICFP*, pages 229–240, 2008.

[36] Aleksandar Nanevski, Viktor Vafeiadis, and Josh Berdine. Structuring the verification of heap-manipulating programs. In *Proc. POPL*, pages 261–274, 2010.

[37] Juan Antonio Navarro Pérez and Andrey Rybalchenko. Separation logic + superposition calculus = heap theorem prover. In *Proc. PLDI*, pages 556–566. ACM, 2011.

[38] Huu Hai Nguyen and Wei-Ngan Chin. Enhancing program verification with lemmas. In *Proc. CAV*, pages 355–369, 2008.

[39] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5:215–244, 1999.

[40] David J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*. Kluwer Academic Pub, 2002.

[41] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proc. LICS*, pages 55–74, 2002.

[42] Harvey Tuch, Gerwin Klein, and Michael Norrish. Types, bytes, and separation logic. In *Proc. POPL*, pages 97–108, 2007.

[43] Tjark Weber. Towards mechanized program verification with separation logic. In *Proc. CSL*, pages 250–264, 2004.

[44] Hongseok Yang. An example of local reasoning in BI pointer logic: the Schorr-Waite graph marking algorithm. In *Proceedings of the 1st Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management*, pages 41–68, 2001.

A Preliminaries on the sequent calculus $\mathrm{S_{BBI}}$

Proposition A.1. *Either $w, \rho \models A$ or $w, \rho \not\models A$ holds for any element w and valuation ρ.*

Proof. By induction on the structure of A. □

Proposition A.2. $w, \rho \models_{\mathcal{S}} (\Gamma \Longrightarrow \Delta), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot)$ *if and only if* $w, \rho \models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$.

Proof.
$w, \rho \models_{\mathcal{S}} (\Gamma \Longrightarrow \Delta), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot)$
iff. $\exists w', w'' \in U$ such that $w \in w' \circ w''$ and $w', \rho \models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$ and $w'', \rho \models_{\mathcal{S}} \emptyset_{\mathsf{m}}$
iff. $\exists w' \in U$ such that $w \in w' \circ e$ and $w', \rho \models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$ from $w'' = e$ (since $w'', \rho \models_{\mathcal{S}} \emptyset_{\mathsf{m}}$ iff. $w'' = e$)
iff. $w, \rho \models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$ from $w = w'$ (since $w \in w' \circ e = \{w'\}$)
□

Proposition A.3. $w, \rho \models_{\mathcal{S}} W_1, W_2$ *if and only if* $w, \rho \models_{\mathcal{S}} W_2, W_1$.

Proof.
$w, \rho \models_{\mathcal{S}} W_1, W_2$
iff. $\exists w_1, w_2 \in U$ such that $w \in w_1 \circ w_2$ and $w_1, \rho \models_{\mathcal{W}} W_1$ and $w_2, \rho \models_{\mathcal{W}} W_2$
iff. $\exists w_2, w_1 \in U$ such that $w \in w_2 \circ w_1$ and $w_2, \rho \models_{\mathcal{W}} W_2$ and $w_1, \rho \models_{\mathcal{W}} W_1$
from $w_1 \circ w_2 = w_2 \circ w_1$ (commutativity)
iff. $w, \rho \models_{\mathcal{S}} W_2, W_1$ □

Proposition A.4. $w, \rho \models_{\mathcal{S}} (W_1, W_2 \Longrightarrow \cdot), W_3$ *if and only if* $w, \rho \models_{\mathcal{S}} W_1, (W_2, W_3 \Longrightarrow \cdot)$.

Proof.
$w, \rho \models_{\mathcal{S}} (W_1, W_2 \Longrightarrow \cdot), W_3$
iff. $\exists w_1, w_2, w_3 \in U$ such that $w \in (w_1 \circ w_2) \circ w_3$ and $w_1, \rho \models_{\mathcal{W}} W_1$ and $w_2, \rho \models_{\mathcal{W}} W_2$ and $w_3, \rho \models_{\mathcal{W}} W_3$
iff. $\exists w_1, w_2, w_3 \in U$ such that $w \in w_1 \circ (w_2 \circ w_3)$ and $w_1, \rho \models_{\mathcal{W}} W_1$ and $w_2, \rho \models_{\mathcal{W}} W_2$ and $w_3, \rho \models_{\mathcal{W}} W_3$
from $w_1 \circ (w_2 \circ w_3) = (w_1 \circ w_2) \circ w_3$ (associativity)
iff. $w, \rho \models_{\mathcal{S}} W_1, (W_2, W_3 \Longrightarrow \cdot)$ □

Proposition A.5. *$A \Longrightarrow A$ is derivable for any formula A.*

Proof. By induction on the structure of the formula A. Here we consider the case of $A = A_1 \star A_2$:

$$\dfrac{\dfrac{\begin{array}{cc}\text{IH on } A_1 & \text{IH on } A_2\\ A_1 \Longrightarrow A_1 & A_2 \Longrightarrow A_2\end{array}}{(A_1 \Longrightarrow \cdot), (A_2 \Longrightarrow \cdot) \Longrightarrow A_1 \star A_2}\ \star R_{\mathcal{S}}}{A_1 \star A_2 \Longrightarrow A_1 \star A_2}\ \star L_{\mathcal{S}}$$

All the remaining cases are similar. □

Proposition A.6.
If $\Gamma \Longrightarrow \Delta$, then $\Gamma; \Gamma' \Longrightarrow \Delta$.
If $\Gamma \Longrightarrow \Delta$, then $\Gamma \Longrightarrow \Delta; \Delta'$.

Proof. By induction on the structure of Γ' and Δ'. □

Proposition A.7.
If $\Gamma; \Gamma'; \Gamma' \Longrightarrow \Delta$, then $\Gamma; \Gamma' \Longrightarrow \Delta$.
If $\Gamma \Longrightarrow \Delta; \Delta'; \Delta'$, then $\Gamma \Longrightarrow \Delta; \Delta'$.

Proof. By induction on the structure of Γ' and Δ'. □

Proposition A.8.
If $\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2)\langle \Gamma_3 \Longrightarrow \Delta_3\rangle \Longrightarrow \Delta_1$, then $\Gamma_2; (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_3 \Longrightarrow \Delta_3\rangle \Longrightarrow \Delta_2$.

Proof.

$$\dfrac{\dfrac{\dfrac{\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2)\langle \Gamma_3 \Longrightarrow \Delta_3\rangle \Longrightarrow \Delta_1}{\Gamma_3; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta_3}\ TC_{\mathcal{S}}}{\Gamma_3; (\Gamma_2 \Longrightarrow \Delta_2), (\Gamma_1 \Longrightarrow \Delta_1) \Longrightarrow \Delta_3}\ E_{\mathcal{S}}}{\Gamma_2; (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_3 \Longrightarrow \Delta_3\rangle \Longrightarrow \Delta_2}\ TP_{\mathcal{S}}$$

□

Proposition A.9.
If $\Gamma_2; (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta_2$, then $\Gamma; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta$.

Proof.

$$\dfrac{\dfrac{\Gamma_2; (\Gamma_1 \Longrightarrow \Delta_1)\langle \overset{\mathcal{D}}{\Gamma \Longrightarrow \Delta}\rangle \Longrightarrow \Delta_2}{\Gamma; (\Gamma_2 \Longrightarrow \Delta_2), (\Gamma_1 \Longrightarrow \Delta_1) \Longrightarrow \Delta}\ TC_{\mathcal{S}}}{\Gamma; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta}\ EC_{\mathcal{S}}$$

□

Proposition A.10 (Inversion)**.**

$$\textit{If } \mathcal{D} :: \begin{cases} \omega[\Gamma; \top \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta; \bot] \\ \omega[\Gamma; \neg A \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta; \neg A] \\ \omega[\Gamma; A \wedge B \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta; A \vee B] \\ \omega[\Gamma; \mathsf{I} \Longrightarrow \Delta] \\ \omega[\Gamma; A \star B \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta; A \mathrel{-\!\!\star} B] \end{cases} \textit{, then } \mathcal{E} :: \begin{cases} \omega[\Gamma \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta; A] \\ \omega[\Gamma; A \Longrightarrow \Delta] \\ \omega[\Gamma; A; B \Longrightarrow \Delta] \\ \omega[\Gamma \Longrightarrow \Delta; A; B] \\ \omega[\Gamma; \emptyset_{\mathsf{m}} \Longrightarrow \Delta] \\ \omega[\Gamma; (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow \Delta] \\ \omega[\Gamma; (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow B\rangle \Longrightarrow \Delta] \end{cases} \textit{ and } \|\mathcal{E}\| \leq \|\mathcal{D}\|$$

Proof. By induction on the size of proof $\mathcal{D}$. □

Corollary A.11.
$\Gamma; \top \Longrightarrow \Delta$ *if and only if* $\Gamma \Longrightarrow \Delta$.
$\Gamma \Longrightarrow \Delta; \bot$ *if and only if* $\Gamma \Longrightarrow \Delta$.
$\Gamma; \neg A \Longrightarrow \Delta$ *if and only if* $\Gamma \Longrightarrow \Delta; A$.
$\Gamma \Longrightarrow \Delta; \neg A$ *if and only if* $\Gamma; A \Longrightarrow \Delta$.
$\Gamma; A \wedge B \Longrightarrow \Delta$ *if and only if* $\Gamma; A; B \Longrightarrow \Delta$.
$\Gamma \Longrightarrow \Delta; A \vee B$ *if and only if* $\Gamma \Longrightarrow \Delta; A; B$.
$\Gamma; \mathsf{I} \Longrightarrow \Delta$ *if and only if* $\Gamma; \emptyset_{\mathsf{m}} \Longrightarrow \Delta$.
$\Gamma; A \star B \Longrightarrow \Delta$ *if and only if* $\Gamma; (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow \Delta$.
$\Gamma \Longrightarrow \Delta; A \mathrel{-\!\!\star} B$ *if and only if* $\Gamma; (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow B\rangle \Longrightarrow \Delta$.

Proof. The proof of *if* part is immediate by corresponding rules, and the proof of *only if* part is immediate by Proposition A.10. □

B Cut elimination in $\mathbf{S_{BBI}}$

Proof of Lemma 4.2. By case analysis on the cut formula C.

Case: $C = P$ where $\mathcal{D} = \dfrac{}{P \Longrightarrow \boxed{P}}\ \mathit{Init}_{\mathcal{S}}$ and $\mathcal{E} = \dfrac{}{\boxed{P} \Longrightarrow P}\ \mathit{Init}_{\mathcal{S}}$

$P \Longrightarrow P$ by $\mathit{Init}_{\mathcal{S}}$

Case: $C = \top$ where $\mathcal{D} = \dfrac{}{\cdot \Longrightarrow \boxed{\top}}\ \top R_{\mathcal{S}}$ and $\mathcal{E} = \dfrac{\mathcal{E}' :: \Gamma' \Longrightarrow \Delta'}{\Gamma'; \boxed{\top} \Longrightarrow \Delta'}\ \top L_{\mathcal{S}}$

$\Gamma' \Longrightarrow \Delta'$ from $\mathcal{E}'$

Case: $C = \bot$ where $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \boxed{\bot}}\ \bot R_{\mathcal{S}}$ and $\mathcal{E} = \dfrac{}{\boxed{\bot} \Longrightarrow \cdot}\ \bot L_{\mathcal{S}}$

$\Gamma \Longrightarrow \Delta$ from $\mathcal{D}'$

Case: $C = \neg C'$ where $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma; C' \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \boxed{\neg C'}}\ \neg R_{\mathcal{S}}$ and $\mathcal{E} = \dfrac{\mathcal{E}' :: \Gamma' \Longrightarrow \Delta'; C'}{\Gamma'; \boxed{\neg C'} \Longrightarrow \Delta'}\ \neg L_{\mathcal{S}}$

$\Gamma'; \Gamma \Longrightarrow \Delta'; \Delta$ by applying $/C/$ to C', $\mathcal{D}'$, and $\mathcal{E}'$

Case: $C = C_1 \wedge C_2$ where $\mathcal{D} = \dfrac{\mathcal{D}_1 :: \Gamma_1 \Longrightarrow \Delta_1; C_1 \quad \mathcal{D}_2 :: \Gamma_2 \Longrightarrow \Delta_2; C_2}{\Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2; \boxed{C_1 \wedge C_2}}\ \wedge R_{\mathcal{S}}$
and $\mathcal{E} = \dfrac{\mathcal{E}' :: \Gamma'; C_1; C_2 \Longrightarrow \Delta'}{\Gamma'; \boxed{C_1 \wedge C_2} \Longrightarrow \Delta'}\ \wedge L_{\mathcal{S}}$

$\mathcal{G} :: \Gamma_1; \Gamma'; C_2 \Longrightarrow \Delta_1; \Delta'$ by applying $/C/$ to C_1, $\mathcal{D}_1$, and $\mathcal{E}'$
$\Gamma_1; \Gamma_2; \Gamma' \Longrightarrow \Delta_1; \Delta_2; \Delta'$ by applying $/C/$ to C_2, $\mathcal{D}_2$, and $\mathcal{G}$

Case: $C = C_1 \vee C_2$ where $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma \Longrightarrow \Delta; C_1; C_2}{\Gamma \Longrightarrow \Delta; \boxed{C_1 \vee C_2}}\ \vee R_{\mathcal{S}}$
and $\mathcal{E} = \dfrac{\mathcal{E}_1 :: \Gamma'_1; C_1 \Longrightarrow \Delta'_1 \quad \mathcal{E}_2 :: \Gamma'_2; C_2 \Longrightarrow \Delta'_2}{\Gamma'_1; \Gamma'_2; \boxed{C_1 \vee C_2} \Longrightarrow \Delta'_1; \Delta'_2}\ \vee L_{\mathcal{S}}$

$\mathcal{G} :: \Gamma; \Gamma'_1 \Longrightarrow \Delta; \Delta'_1; C_2$ by applying $/C/$ to C_1, $\mathcal{D}'$, and $\mathcal{E}_1$
$\Gamma; \Gamma'_1; \Gamma'_2 \Longrightarrow \Delta; \Delta'_1; \Delta'_2$ by applying $/C/$ to C_2, $\mathcal{G}$, and $\mathcal{E}_2$

Case: $C = C_1 \to C_2$ where $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma; C_1 \Longrightarrow \Delta; C_2}{\Gamma \Longrightarrow \Delta; \boxed{C_1 \to C_2}}\ {\to}R_{\mathcal{S}}$

and $\mathcal{E} = \dfrac{\mathcal{E}_1 :: \Gamma'_1 \Longrightarrow \Delta'_1; C_1 \quad \mathcal{E}_2 :: \Gamma'_2; C_2 \Longrightarrow \Delta'_2}{\Gamma'_1; \Gamma'_2; \boxed{C_1 \to C_2} \Longrightarrow \Delta'_1; \Delta'_2}\ {\to}L_{\mathcal{S}}$

$\mathcal{G} :: \Gamma; \Gamma'_1 \Longrightarrow \Delta; \Delta'_1; C_2$ by applying $/C/$ to C_1, $\mathcal{E}_1$, and $\mathcal{D}'$

$\Gamma; \Gamma'_1; \Gamma'_2 \Longrightarrow \Delta; \Delta'_1; \Delta'_2$ by applying $/C/$ to C_2, $\mathcal{G}$, and $\mathcal{E}_2$

Case: $C = \mathsf{I}$ where $\mathcal{D} = \dfrac{}{\emptyset_{\mathsf{m}} \Longrightarrow \mathsf{I}}\ \mathsf{I}R_{\mathcal{S}}$ and $\mathcal{E} = \dfrac{\mathcal{E}' :: \Gamma'; \emptyset_{\mathsf{m}} \Longrightarrow \Delta'}{\Gamma'; \mathsf{I} \Longrightarrow \Delta'}\ \mathsf{I}L_{\mathcal{S}}$

$\Gamma'; \emptyset_{\mathsf{m}} \Longrightarrow \Delta'$ from $\mathcal{E}'$

Case: $C = C_1 \star C_2$ where $\mathcal{D} = \dfrac{\mathcal{D}_1 :: \Gamma_1 \Longrightarrow \Delta_1; C_1 \quad \mathcal{D}_2 :: \Gamma_2 \Longrightarrow \Delta_2; C_2}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \boxed{C_1 \star C_2}}\ {\star}R_{\mathcal{S}}$

and $\mathcal{E} = \dfrac{\mathcal{E}' :: \Gamma'; (C_1 \Longrightarrow \cdot), (C_2 \Longrightarrow \cdot) \Longrightarrow \Delta'}{\Gamma'; \boxed{C_1 \star C_2} \Longrightarrow \Delta'}\ {\star}L_{\mathcal{S}}$

$\mathcal{G}_1 :: (C_2 \Longrightarrow \cdot)\langle \Gamma' \Longrightarrow \Delta' \rangle; C_1 \Longrightarrow \cdot$ by $TP_{\mathcal{S}}$

$(C_2 \Longrightarrow \cdot)\langle \Gamma' \Longrightarrow \Delta' \rangle; \Gamma_1 \Longrightarrow \Delta_1$ by applying $/C/$ to C_1, $\mathcal{D}_1$, and $\mathcal{G}_1$

$\mathcal{G}_2 :: (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma' \Longrightarrow \Delta' \rangle; C_2 \Longrightarrow \cdot$ by Proposition A.8

$(\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma' \Longrightarrow \Delta' \rangle; \Gamma_2 \Longrightarrow \Delta_2$ by applying $/C/$ to C_2, $\mathcal{D}_2$, and $\mathcal{G}_2$

$(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2); \Gamma' \Longrightarrow \Delta'$ by Proposition A.9

Case: $C = C_1 \mathrel{-\!\!\star} C_2$ where $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma; (C_1 \Longrightarrow \cdot)\langle \cdot \Longrightarrow C_2 \rangle \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \boxed{C_1 \mathrel{-\!\!\star} C_2}}\ {\mathrel{-\!\!\star}}R_{\mathcal{S}}$

and $\mathcal{E} = \dfrac{\mathcal{E}_1 :: \Gamma'_1 \Longrightarrow \Delta'_1; C_1 \quad \mathcal{E}_2 :: \Gamma'_2; C_2 \Longrightarrow \Delta'_2}{(\Gamma'_1 \Longrightarrow \Delta'_1)\langle \Gamma'_2 \Longrightarrow \Delta'_2 \rangle; \boxed{C_1 \mathrel{-\!\!\star} C_2} \Longrightarrow \cdot}\ {\mathrel{-\!\!\star}}L_{\mathcal{S}}$

$\mathcal{G}_1 :: (\Gamma \Longrightarrow \Delta)\langle \cdot \Longrightarrow C_2 \rangle; C_1 \Longrightarrow \cdot$ by Proposition A.8

$(\Gamma \Longrightarrow \Delta)\langle \cdot \Longrightarrow C_2 \rangle; \Gamma'_1 \Longrightarrow \Delta'_1$ by applying $/C/$ to C_1, $\mathcal{E}_1$, and $\mathcal{D}'$

$\mathcal{G}_2 :: (\Gamma \Longrightarrow \Delta), (\Gamma'_1 \Longrightarrow \Delta'_1) \Longrightarrow C_2$ by Proposition A.9

$(\Gamma \Longrightarrow \Delta), (\Gamma'_1 \Longrightarrow \Delta'_1); \Gamma'_2 \Longrightarrow \Delta'_2$ by applying $/C/$ to C_2, $\mathcal{G}_1$ and $\mathcal{E}_2$

$\Gamma; (\Gamma'_1 \Longrightarrow \Delta'_1)\langle \Gamma'_2 \Longrightarrow \Delta'_2 \rangle \Longrightarrow \Delta$ by $TP_{\mathcal{S}}$

□

Proof of Lemma 4.3. By induction on the structure of proof $\mathcal{D}$ of $\omega[\Gamma_1 \Longrightarrow \Delta_1; C^{n_1}] \cdots [\Gamma_k \Longrightarrow \Delta_k; C^{n_k}]$.

Suppose that none of C's are the principal formula of $R_{\mathcal{D}}$. In this case, we complete the proof by applying $R_{\mathcal{D}}$ again to the induction hypothesis. For example, let us consider the following case that $R_{\mathcal{D}}$ is $CL_{\mathcal{S}}$:

$\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma_1; S_1; S_2; S_2 \Longrightarrow \Delta_1; C^{n_1}}{\Gamma_1; S_1; S_2 \Longrightarrow \Delta_1; C^{n_1}}\ CL_{\mathcal{S}}$

where $\begin{array}{l} S_1 = \sigma_1[\Gamma_2 \Longrightarrow \Delta_2; C^{n_2}] \cdots [\Gamma_i \Longrightarrow \Delta_i; C^i] \\ S_2 = \sigma_2[\Gamma_{i+1} \Longrightarrow \Delta_{i+1}; C^{n_{i+1}}] \cdots [\Gamma_k \Longrightarrow \Delta_k; C^{n_k}] \end{array}$ and $\omega = [; \sigma_1; \sigma_2 \Longrightarrow]$

$\Gamma_1; \Gamma'; S'_1; S'_2; S'_2 \Longrightarrow \Delta_1; \Delta'$ where $\begin{array}{l} S'_1 = \sigma_1[\Gamma_2; \Gamma' \Longrightarrow \Delta_2; \Delta'] \cdots [\Gamma_i; \Gamma' \Longrightarrow \Delta_i; \Delta'] \\ S'_2 = \sigma_2[\Gamma_{i+1}; \Gamma' \Longrightarrow \Delta_{i+1}; \Delta'] \cdots [\Gamma_k; \Gamma' \Longrightarrow \Delta_k; \Delta'] \end{array}$ by IH on $\mathcal{D}'$

$\Gamma_1; \Gamma'; S'_1; S'_2 \Longrightarrow \Delta_1; \Delta'$ by $CL_{\mathcal{S}}$

All the remaining cases are similar.

Suppose that one of C's is the principal formula of $R_{\mathcal{D}}$.

Case: $\mathcal{D} = \dfrac{}{P \Longrightarrow \boxed{P}}\ Init_{\mathcal{S}}$ where $C = P$ and $\omega = []$

$\Gamma'; P \Longrightarrow \Delta'$ assumption

Case: $\mathcal{D} = \dfrac{}{\cdot \Longrightarrow \boxed{\top}}\ \top R_{\mathcal{S}}$ where $C = \top$ and $\omega = []$

$\Gamma' \Longrightarrow \Delta'$ by Lemma 4.2

Case: $\mathcal{D} = \dfrac{}{\emptyset_{\mathsf{m}} \Longrightarrow \boxed{\mathsf{I}}}\ \mathsf{I} R_{\mathcal{S}}$ where $C = \mathsf{I}$ and $\omega = []$

$\Gamma'; \emptyset_{\mathsf{m}} \Longrightarrow \Delta'$ by Lemma 4.2

Case: $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma_1; \gamma[\Gamma_2 \Longrightarrow \Delta_2; \bot^{n_2}] \cdots [\Gamma_k \Longrightarrow \Delta_k; \bot^{n_k}] \Longrightarrow \Delta_1; \bot^{n_1}}{\Gamma_1; \gamma[\Gamma_2 \Longrightarrow \Delta_2; \bot^{n_2}] \cdots [\Gamma_k \Longrightarrow \Delta_k; \bot^{n_k}] \Longrightarrow \Delta_1; \bot^{n_1}; \boxed{\bot}}\ \bot R_{\mathcal{S}}$
where $C = \top$ and $\omega = [; \gamma \Longrightarrow]$

$\Gamma_1; \Gamma'; \gamma[\Gamma_1; \Gamma' \Longrightarrow \Delta_1; \Delta'] \cdots [\Gamma_k; \Gamma' \Longrightarrow \Delta_k; \Delta'] \Longrightarrow \Delta_1; \Delta'$ by IH on $\mathcal{D}'$

Case: $\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma_1; \gamma[\Gamma_2 \Longrightarrow \Delta_2; \neg C'^{n_2}] \cdots [\Gamma_k \Longrightarrow \Delta_k; \neg C'^{n_k}]; C' \Longrightarrow \Delta_1; \neg C'^{n_1}}{\Gamma_1; \gamma[\Gamma_2 \Longrightarrow \Delta_2; \neg C'^{n_2}] \cdots [\Gamma_k \Longrightarrow \Delta_k; \neg C'^{n_k}] \Longrightarrow \Delta_1; \neg C'^{n_1}; \boxed{\neg C'}}\ \neg R_{\mathcal{S}}$
where $C = \neg C'$ and $\omega = [; \gamma \Longrightarrow]$

$\Gamma_1; \Gamma'; \gamma[\Gamma_2; \Gamma' \Longrightarrow \Delta_2; \Delta'] \cdots [\Gamma_k; \Gamma' \Longrightarrow \Delta_k; \Delta']; C' \Longrightarrow \Delta_1; \Delta'$ by IH on $\mathcal{D}'$
$\Gamma_1; \Gamma'; \gamma[\Gamma_2; \Gamma' \Longrightarrow \Delta_2; \Delta'] \cdots [\Gamma_k; \Gamma' \Longrightarrow \Delta_k; \Delta'] \Longrightarrow \Delta_1; \Delta'; \boxed{\neg C'}$ by $\neg L_{\mathcal{S}}$
$\Gamma_1; \Gamma'; \Gamma'; \gamma[\Gamma_2; \Gamma' \Longrightarrow \Delta_2; \Delta'] \cdots [\Gamma_k; \Gamma' \Longrightarrow \Delta_k; \Delta'] \Longrightarrow \Delta_1; \Delta'; \Delta'$ by Lemma 4.2
$\Gamma_1; \Gamma'; \gamma[\Gamma_2; \Gamma' \Longrightarrow \Delta_2; \Delta'] \cdots [\Gamma_k; \Gamma' \Longrightarrow \Delta_k; \Delta'] \Longrightarrow \Delta_1; \Delta'$ by Proposition A.7

All the remaining cases are similar. □

Proof of Lemma 4.4.
By induction on the structure of proof $\mathcal{E}$ of $\omega'[\Gamma'_1; C^{n_1} \Longrightarrow \Delta'_1] \cdots [\Gamma'_k; C^{n_k} \Longrightarrow \Delta'_k]$.

The proof proceeds similarly to the proof of Lemma 4.3. When all C's are not the principal formula of $R_{\mathcal{E}}$, the proof proceeds by applying $R_{\mathcal{E}}$ again to the induction hypothesis completes the proof. When one of C's is the principal formula of $R_{\mathcal{E}}$, the proof is also the same as the proof of Lemma 4.3 except that it depends on Lemma 4.3 instead of Lemma 4.2. □

Proof of Theorem 4.1. By induction on the structure of the cut formula C.
$\Gamma \Longrightarrow \Delta; C$ assumption
$\Gamma'; C \Longrightarrow \Delta'$ assumption
$/C/$ by IH on C
$\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$ by Lemma 4.4
□

C Soundness and completeness of $\mathbf{S}_{\mathbf{BBI}}$

Proof of Theorem 5.1. By induction on the structure of proof $\mathcal{D}$ of $\Gamma \Longrightarrow \Delta$.

Most cases are immediate, and thus we present non-trivial cases only.

Case $\mathcal{D} = \dfrac{\Gamma_{\mathrm{p}}; (\Gamma \Longrightarrow \Delta), (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}) \Longrightarrow \Delta_{\mathrm{p}}}{\Gamma; (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}) \langle \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \rangle \Longrightarrow \Delta}\ TP_{\mathcal{S}}$ where $W = \Gamma; (\Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}) \langle \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \rangle \Longrightarrow \Delta$

Suppose that there is an element w that satisfies $w, \rho \models W$ for some valuation ρ. Since the element w satisfies $w, \rho \models_{\mathcal{S}} W_{\mathrm{s}}\langle W_{\mathrm{p}}\rangle$ ($W_{\mathrm{s}} = \Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}}$ and $W_{\mathrm{p}} = \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}}$), elements w_{s} and w_{p} such that $w_{\mathrm{p}} \in w \circ w_{\mathrm{s}}$ should exist. These elements w, w_{s}, and w_{p} satisfy $w, \rho \models_{\mathcal{W}} W'$ ($W' = \Gamma \Longrightarrow \Delta$), $w_{\mathrm{s}}, \rho \models_{\mathcal{W}} W_{\mathrm{s}}$, and $w_{\mathrm{p}}, \rho \models_{\mathcal{W}} W_{\mathrm{p}}$, respectively. The existence of the element w_{p} leads to a contradiction because such an element cannot exist according to the induction hypothesis. Therefore we conclude that there is no element w and valuation ρ that satisfies $w, \rho \models_{\mathcal{W}} W$.

Case $\mathcal{D} = \dfrac{\Gamma_{\mathrm{c}1}; (\Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2})\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{\mathrm{c}1}}{\Gamma; (\Gamma_{\mathrm{c}1} \Longrightarrow \Delta_{\mathrm{c}1}), (\Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2}) \Longrightarrow \Delta}\ TC_{\mathcal{S}}$
where $W = \Gamma; (\Gamma_{\mathrm{c}1} \Longrightarrow \Delta_{\mathrm{c}1}), (\Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2}) \Longrightarrow \Delta$

Suppose that there is an element w that satisfies $w, \rho \models W$ for some valuation ρ. Since the element w satisfies $w, \rho \models_{\mathcal{S}} W_{\mathrm{c}1}, W_{\mathrm{c}2}$ ($W_{\mathrm{c}1} = \Gamma_{\mathrm{c}1} \Longrightarrow \Delta_{\mathrm{c}1}$ and $W_{\mathrm{c}2} = \Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2}$), elements $w_{\mathrm{c}1}$ and $w_{\mathrm{c}2}$ such that $w \in w_{\mathrm{c}1} \circ w_{\mathrm{c}2}$ should exist. These elements w, $w_{\mathrm{c}1}$, and $w_{\mathrm{c}2}$ satisfy $w, \rho \models_{\mathcal{W}} W'$ ($W' = \Gamma \Longrightarrow \Delta$), $w_{\mathrm{c}1}, \rho \models_{\mathcal{W}} W_{\mathrm{c}1}$, and $w_{\mathrm{c}2}, \rho \models_{\mathcal{W}} W_{\mathrm{c}2}$, respectively. The existence of the element $w_{\mathrm{c}1}$ leads to a contradiction because such an element cannot exist according to the induction hypothesis. Therefore we conclude that there is no element w and valuation ρ that satisfies $w, \rho \models_{\mathcal{W}} W$.

Case $\mathcal{D} = \dfrac{\Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2}{\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta_1}\ \emptyset_{\mathsf{m}} D_{\mathcal{S}}$ where $W = \Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta_1$

Suppose that there is an element w that satisfies $w, \rho \models_{\mathcal{W}} W$ for some valuation ρ. Note that $w, \rho \models_{\mathcal{W}} W$ holds if and only if $w, \rho \models_{\mathcal{W}} \Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2$ holds:

$w, \rho \models_{\mathcal{W}} W$
iff. $w, \rho \models_{\mathcal{W}} \Gamma_1 \Longrightarrow \Delta_1$ and $w, \rho \models_{\mathcal{S}} (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot)$
iff. $w, \rho \models_{\mathcal{W}} \Gamma_1 \Longrightarrow \Delta_1$ and $w, \rho \models_{\mathcal{W}} \Gamma_2 \Longrightarrow \Delta_2$ by Proposition A.2
iff. $w, \rho \models_{\mathcal{W}} \Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2$

Thus the assumption leads to a contradiction because $\not\models_{\mathcal{W}} \Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2$ according to the induction hypothesis. Therefore we conclude that there is no element w and valuation ρ that satisfies $w, \rho \models_{\mathcal{W}} W$.

The case that $R_{\mathcal{D}}$ is $\emptyset_{\mathsf{m}} U_{\mathcal{S}}$ is similar.

Case $\mathcal{D} = \dfrac{\Gamma; W_2, W_1 \Longrightarrow \Delta}{\Gamma; W_1, W_2 \Longrightarrow \Delta}\ EC_{\mathcal{S}}$ where $W = \Gamma; W_1, W_2 \Longrightarrow \Delta$

Suppose that there is an element w that satisfies $w, \rho \models_{\mathcal{W}} W$ for some valuation ρ. Note that $w, \rho \models \Gamma; W_2, W_1 \Longrightarrow \Delta$ holds if and only if $w, \rho \models \Gamma; W_1, W_2 \Longrightarrow \Delta$ holds:

$w, \rho \models \Gamma; W_2, W_1 \Longrightarrow \Delta$
iff. $w, \rho \models \Gamma \Longrightarrow \Delta$ and $w, \rho \models_{\mathcal{S}} W_1, W_2$
iff. $w, \rho \models \Gamma \Longrightarrow \Delta$ and $w, \rho \models_{\mathcal{S}} W_2, W_1$ by Proposition A.3
iff. $w, \rho \models \Gamma; W_2, W_1 \Longrightarrow \Delta$

Thus the assumption leads to a contradiction because $\not\models_{\mathcal{W}} \Gamma; W_2, W_1 \Longrightarrow \Delta$ according to the induction hypothesis. Therefore we conclude that there is no element w and valuation ρ that satisfies $w, \rho \models_{\mathcal{W}} W$.

The case that $R_{\mathcal{D}}$ is $EA_{\mathcal{S}}$ is similar except that it depends on Proposition A.4 instead of Proposition A.3.

Case $\mathcal{D} = \dfrac{\overset{\mathcal{D}_1}{\Gamma_1 \Longrightarrow \Delta_1; A} \quad \overset{\mathcal{D}_2}{\Gamma_2 \Longrightarrow \Delta_2; B}}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\ \star R_{\mathcal{S}}$ where $W = (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B$

Suppose that there is an element w that satisfies $w, \rho \models_{\mathcal{W}} W$ for some valuation ρ. Since the element w satisfies $w, \rho \models_{\mathcal{S}} W_1, W_2$ (where $W_1 = \Gamma_1 \Longrightarrow \Delta_1$ and $W_2 = \Gamma_2 \Longrightarrow \Delta_2$), elements w_1 and w_2 such that $w \in w_1 \circ w_2$ should exist. These elements w_1 and w_2 satisfy $w_1, \rho \models_{\mathcal{W}} W_1$ and $w_2, \rho \models_{\mathcal{W}} W_2$, respectively. Note that either $w_1, \rho \not\models A$ or $w_2, \rho \not\models B$ should hold because the element w also satisfies $w, \rho \not\models A \star B$:

- Suppose that $w_1, \rho \not\models A$ holds:

  This assumption leads to a contradiction because it implies the existence of element w_1 that satisfies $w_1, \rho \models_{\mathcal{W}} \Gamma_1 \Longrightarrow \Delta_1; A$, but such an element cannot exist according to the induction hypothesis on $\mathcal{D}_1$.

- Suppose that $w_2, \rho \not\models B$ holds:

  This assumption leads to a contradiction because it implies the existence of element w_2 that satisfies $w_2, \rho \models_{\mathcal{W}} \Gamma_2 \Longrightarrow \Delta_2; B$, but such an element cannot exist according to the induction hypothesis on $\mathcal{D}_2$.

Both cases lead to a contradiction. Therefore we conclude that there is no element w and valuation ρ that satisfies $w, \rho \models_{\mathcal{W}} W$.

$$\text{Case } \mathcal{D} = \dfrac{\Gamma_1 \Longrightarrow \Delta_1; A \quad \Gamma_2; B \Longrightarrow \Delta_2}{(\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2\rangle; A \mathrel{-\!\!\star} B \Longrightarrow \cdot}\ {\mathrel{-\!\!\star}}L_{\mathcal{S}}$$

where $W = (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2\rangle; A \mathrel{-\!\!\star} B \Longrightarrow \cdot$

Suppose that there is an element w that satisfies $w, \rho \models_{\mathcal{W}} W$. Since the element w satisfies $w, \rho \models_{\mathcal{S}} W_1\langle W_2\rangle$ (where $W_1 = \Gamma_1 \Longrightarrow \Delta_1$ and $W_2 = \Gamma_2 \Longrightarrow \Delta_2$), elements w_1 and w_2 such that $w_2 \in w \circ w_1$ should exist. These elements w_1 and w_2 satisfy $w_1, \rho \models_{\mathcal{W}} W_1$ and $w_2, \rho \models_{\mathcal{W}} W_2$, respectively. Note that either $w_1, \rho \models A$ or $w_1, \rho \not\models A$ should hold:

- Suppose that $w_1, \rho \models A$ holds.

  Since the element w satisfies $w, \rho \models A \mathrel{-\!\!\star} B$, the element w_2 should satisfy $w_2, \rho \models B$, which implies the existence of element w_2 that satisfies $w_2, \rho \models_{\mathcal{W}} \Gamma_2; B \Longrightarrow \Delta_2$. Thus, this assumption leads to a contradiction because such an element cannot exist according to the induction hypothesis on $\mathcal{D}_2$.

- Suppose that $w_1, \rho \not\models A$ holds.

  The assumption implies the existence of element w_1 that satisfies $w_1, \rho \models_{\mathcal{W}} \Gamma_1 \Longrightarrow \Delta_1; A$. Thus, this assumption leads to a contradiction because such an element cannot exist according to the induction hypothesis on $\mathcal{D}_1$.

Both cases lead to a contradiction. Therefore, we may conclude that there is no element w and valuation ρ that satisfies $w, \rho \models_{\mathcal{W}} W$. □

Proof of Lemma 5.3. By simultaneous induction on the structure of W and S.

Suppose that $w, \rho \models_{\mathcal{S}} S$iff. $w, \rho \models [\![S]\!]_{\mathrm{s}}$ holds for every $S \in \Gamma$ where $W = \Gamma \Longrightarrow \Delta$ (by IH on W). This assumption implies that the following relationship holds:

$$w, \rho \models_{\mathcal{W}} \Gamma \Longrightarrow \Delta \quad \text{iff.} \quad \begin{array}{l} \forall S \in \Gamma.\ w, \rho \models_{\mathcal{S}} S \\ \forall A \in \Delta.\ w, \rho \not\models A \end{array} \quad \text{iff.} \quad \begin{array}{l} \forall S \in \Gamma.\ w, \rho \models [\![S]\!]_{\mathrm{s}} \\ \forall A \in \Delta.\ w, \rho \not\models A \end{array}$$

By definition, $\forall S \in \Gamma.\ w, \rho \models [\![S]\!]_{\mathrm{s}}$ holds if and only if $w, \rho \models [\![S_1]\!]_{\mathrm{s}} \wedge \cdots \wedge [\![S_n]\!]_{\mathrm{s}}$ holds. Because $\neg A_1 \wedge \neg A_2$iff. $\neg(A_1 \vee A_2)$ holds, $\forall A \in \Delta.\ w, \rho \not\models A$ holds if and only if $w, \rho \models \neg(A_1 \vee \cdots \vee A_m)$ holds. Note that $[\![\Gamma]\!]_{\mathrm{g}} = [\![S_1]\!]_{\mathrm{s}} \wedge \cdots \wedge [\![S_n]\!]_{\mathrm{s}}$ and $[\![\Delta]\!]_{\mathrm{d}} = \neg(A_1 \vee \cdots \vee A_m)$. Therefore, from these observations, we conclude that $w, \rho \models_{\mathcal{W}} W$iff. $w, \rho \models [\![W]\!]_{\mathrm{w}}$.

Suppose that $w, \rho \models_{\mathcal{W}} W$iff. $w, \rho \models [\![W]\!]_{\mathrm{w}}$ holds for every $W \in S$ (by IH on S). With this assumption, we conclude that $w, \rho \models_{\mathcal{S}} S$iff. $w, \rho \models [\![S]\!]_{\mathrm{s}}$ holds because the satisfaction relation for multiplicative formulas can be rewritten in terms of the satisfaction relation for node states as follows:

$$\begin{array}{rll} w, \rho \models \mathsf{I} & \text{iff.} & w, \rho \models_{\mathcal{S}} \emptyset_{\mathsf{m}} \\ w, \rho \models A \star B & \text{iff.} & w, \rho \models_{\mathcal{S}} (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \\ w, \rho \models \neg(A \mathrel{-\!\!\star} B) & \text{iff.} & w, \rho \models_{\mathcal{S}} (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow B\rangle \end{array}$$

□

Proof of Proposition 5.4.

$\not\models_{\mathcal{W}} W$
iff. $\forall w \in U$ and ρ. $w, \rho \not\models_{\mathcal{W}} W$
iff. $\forall w \in U$ and ρ. $w, \rho \not\models [\![W]\!]_{\mathrm{w}}$ by Lemma 5.3
iff. $\forall w \in U$ and ρ. $w, \rho \models \neg[\![W]\!]_{\mathrm{w}}$
iff. $\models \neg[\![W]\!]_{\mathrm{w}}$ □

Proof of Lemma 5.5. By simultaneous induction on the structure of $W = \Gamma' \Longrightarrow \Delta'$ and S.

We first prove $\Gamma; [\![\Gamma' \Longrightarrow \Delta']\!]_{\mathrm{w}} \Longrightarrow \Delta$ if and only if $\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$. Suppose that $\Gamma' = S_1; \cdots; S_n$ and $\Delta' = A_1; \cdots; A_m$.

$$\Gamma; \Gamma' \Longrightarrow \Delta; \Delta' \quad \text{iff.} \quad \Gamma; [\![S_1]\!]_{\mathrm{s}}; \cdots; [\![S_n]\!]_{\mathrm{s}} \Longrightarrow \Delta; A_1; \cdots; A_m \quad \text{iff.} \quad \Gamma; [\![\Gamma' \Longrightarrow \Delta']\!]_{\mathrm{w}} \Longrightarrow \Delta$$

The left *if and only if* relation holds because we may assume that $\Gamma; [\![S]\!]_{\mathrm{s}} \Longrightarrow \Delta$iff. $\Gamma; S \Longrightarrow \Delta$ for any Γ and Δ (by IH on W). The right *if and only if* relation holds because $\Gamma; [\![\Gamma' \Longrightarrow \Delta']\!]_{\mathrm{w}} \Longrightarrow \Delta$ is derivable from $\Gamma; [\![S_1]\!]_{\mathrm{s}}; \cdots; [\![S_n]\!]_{\mathrm{s}} \Longrightarrow \Delta; A_1; \cdots; A_m$ by applying only $\wedge L_{\mathcal{S}}$, $\vee R_{\mathcal{S}}$, and $\neg L_{\mathcal{S}}$, and all the rules involved are invertible (Corollary A.11).

We then prove $\Gamma; [\![S]\!]_{\mathrm{s}} \Longrightarrow \Delta$ if and only if $\Gamma; S \Longrightarrow \Delta$. Similarly we complete the proof by using the inversion property of $\mathbf{S_{BBI}}$ (Corollary A.11). □

Proof of Proposition 5.6.
$\cdot \Longrightarrow \neg[\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}}$ assumption.
$[\![\Gamma]\!]_{\mathrm{g}} \Longrightarrow [\![\Delta]\!]_{\mathrm{d}}$ by Corollary A.11
$\Gamma \Longrightarrow \Delta$ by Lemma 5.5
□

Proof of Lemma 5.7. By induction on the structure of proof $\mathcal{D}$ of $\vdash A$.

Since $\mathbf{S_{BBI}}$ is a conservative extension of the sequent calculus for classical propositional logic, all the axioms and rules for additive connectives are derivable in $\mathbf{S_{BBI}}$. Therefore, the only axioms and rules that we need to consider are those axioms and rules for multiplicative connectives.

We first shows that all the axioms are derivable in $\mathbf{S_{BBI}}$.

Case $\mathcal{D} = \vdash C \to \mathsf{I} \star C$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset_{\mathsf{m}} \Longrightarrow \mathsf{I}}\,\mathsf{I}R_{\mathcal{S}} \quad \dfrac{}{C \Longrightarrow C}\,Init_{\mathcal{S}}}{(\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (C \Longrightarrow \cdot) \Longrightarrow \mathsf{I} \star C}\,\star R_{\mathcal{S}}}{(C \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \mathsf{I} \star C}\,EC_{\mathcal{S}}}{C \Longrightarrow \mathsf{I} \star C}\,\emptyset_{\mathsf{m}}U_{\mathcal{S}}}{\cdot \Longrightarrow C \to \mathsf{I} \star C}\,{\to}R_{\mathcal{S}}$$

Case $\mathcal{D} = \vdash \mathsf{I} \star C \to C$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{C \Longrightarrow C}\,Init_{\mathcal{S}}}{(C \Longrightarrow \cdot), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow C}\,\emptyset_{\mathsf{m}}D_{\mathcal{S}}}{(\emptyset_{\mathsf{m}} \Longrightarrow \cdot), (C \Longrightarrow \cdot) \Longrightarrow C}\,EC_{\mathcal{S}}}{\emptyset_{\mathsf{m}}; (C \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}\,TP_{\mathcal{S}}}{\mathsf{I}; (C \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}\,\mathsf{I}L_{\mathcal{S}}}{(\mathsf{I} \Longrightarrow \cdot), (C \Longrightarrow \cdot) \Longrightarrow C}\,TC_{\mathcal{S}}}{\mathsf{I} \star C \Longrightarrow C}\,\star L_{\mathcal{S}}}{\cdot \Longrightarrow \mathsf{I} \star C \to C}\,{\to}R_{\mathcal{S}}$$

Case $\mathcal{D} = \vdash A_1 \star A_2 \to A_2 \star A_1$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A_2 \Longrightarrow A_2}\,Init_{\mathcal{S}} \quad \dfrac{}{A_1 \Longrightarrow A_1}\,Init_{\mathcal{S}}}{(A_2 \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow A_2 \star A_1}\,\star R_{\mathcal{S}}}{(A_2 \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow A_2 \star A_1}\,EC_{\mathcal{S}}}{(A_1 \Longrightarrow \cdot), (A_2 \Longrightarrow \cdot) \Longrightarrow A_2 \star A_1}\,\star L_{\mathcal{S}}}{\cdot \Longrightarrow A_1 \star A_2 \to A_2 \star A_1}\,\to R_{\mathcal{S}}$$

Case $\mathcal{D} = \vdash A_1 \star (A_2 \star A_3) \to (A_1 \star A_2) \star A_3$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A_1 \Longrightarrow A_1}\,Init_{\mathcal{S}} \quad \dfrac{}{A_2 \Longrightarrow A_2}\,Init_{\mathcal{S}}}{(A_1 \Longrightarrow \cdot), (A_2 \Longrightarrow \cdot) \Longrightarrow A_1 \star A_2}\,\star R_{\mathcal{S}}}{(A_2 \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow A_1 \star A_2}\,EC_{\mathcal{S}} \quad \dfrac{}{A_3 \Longrightarrow A_3}\,Init_{\mathcal{S}}}{((A_2 \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow \cdot), (A_3 \Longrightarrow \cdot) \Longrightarrow (A_1 \star A_2) \star A_3}\,\star R_{\mathcal{S}}}{(A_3 \Longrightarrow \cdot), ((A_2 \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow \cdot) \Longrightarrow (A_1 \star A_2) \star A_3}\,EC_{\mathcal{S}}}{((A_3 \Longrightarrow \cdot), (A_2 \Longrightarrow \cdot) \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow (A_1 \star A_2) \star A_3}\,EA_{\mathcal{S}}}{(A_3 \Longrightarrow \cdot), (A_2 \Longrightarrow \cdot); (A_1 \Longrightarrow \cdot)\langle \cdot \Longrightarrow (A_1 \star A_2) \star A_3\rangle \Longrightarrow \cdot}\,TP_{\mathcal{S}}}{(A_2 \Longrightarrow \cdot), (A_3 \Longrightarrow \cdot); (A_1 \Longrightarrow \cdot)\langle \cdot \Longrightarrow (A_1 \star A_2) \star A_3\rangle \Longrightarrow \cdot}\,EC_{\mathcal{S}}}{A_2 \star A_3; (A_1 \Longrightarrow \cdot)\langle \cdot \Longrightarrow (A_1 \star A_2) \star A_3\rangle \Longrightarrow \cdot}\,\star L_{\mathcal{S}}}{(A_2 \star A_3 \Longrightarrow \cdot), (A_1 \Longrightarrow \cdot) \Longrightarrow (A_1 \star A_2) \star A_3}\,TC_{\mathcal{S}}}{(A_1 \Longrightarrow \cdot), (A_2 \star A_3 \Longrightarrow \cdot) \Longrightarrow (A_1 \star A_2) \star A_3}\,EC_{\mathcal{S}}}{A_1 \star (A_2 \star A_3) \Longrightarrow (A_1 \star A_2) \star A_3}\,\star L_{\mathcal{S}}}{\cdot \Longrightarrow A_1 \star (A_2 \star A_3) \to (A_1 \star A_2) \star A_3}\,\to R_{\mathcal{S}}$$

We then show that all the inference rules are derivable in $\mathbf{S_{BBI}}$.

Case $\mathcal{D} = \dfrac{\overset{\mathcal{D}_1}{\vdash A_1 \to A_2} \quad \overset{\mathcal{D}_2}{\vdash B_1 \to B_2}}{\vdash (A_1 \star B_1) \to (A_2 \star B_2)}\,\star H$

$\mathcal{G}_A :: \cdot \Longrightarrow A_1 \to A_2$ by IH on $\mathcal{D}_1$

$\mathcal{G}_B :: \cdot \Longrightarrow B_1 \to B_2$ by IH on $\mathcal{D}_2$

$\mathcal{G}'_A :: A_1; A_1 \to A_2 \Longrightarrow A_2$ by $\dfrac{\dfrac{}{A_1 \Longrightarrow A_1}\,Init_{\mathcal{S}} \quad \dfrac{}{A_2 \Longrightarrow A_2}\,Init_{\mathcal{S}}}{A_1; A_1 \to A_2 \Longrightarrow A_2}\,\to L_{\mathcal{S}}$

$\mathcal{G}'_B :: B_1; B_1 \to B_2 \Longrightarrow B_2$ by $\dfrac{\dfrac{}{B_1 \Longrightarrow B_1}\,Init_{\mathcal{S}} \quad \dfrac{}{B_2 \Longrightarrow B_2}\,Init_{\mathcal{S}}}{B_1; B_1 \to B_2 \Longrightarrow B_2}\,\to L_{\mathcal{S}}$

$\mathcal{G}_1 :: A_1 \Longrightarrow A_2$ by applying Theorem 4.1 to $\mathcal{G}_A$ and $\mathcal{G}'_A$

$\mathcal{G}_2 :: B_1 \Longrightarrow B_2$ by applying Theorem 4.1 to $\mathcal{G}_B$ and $\mathcal{G}'_B$

$\cdot \Longrightarrow (A_1 \star B_1) \to (A_2 \star B_2)$ by $\dfrac{\dfrac{\dfrac{}{A_1 \Longrightarrow A_2}\,Init_{\mathcal{S}} \quad \dfrac{}{B_1 \Longrightarrow B_2}\,Init_{\mathcal{S}}}{(A_1 \Longrightarrow \cdot), (B_1 \Longrightarrow \cdot) \Longrightarrow A_2 \star B_2}\,\star R_{\mathcal{S}}}{A_1 \star B_1 \Longrightarrow A_2 \star B_2}\,\star \mathsf{L}}{\cdot \Longrightarrow (A_1 \star B_1) \to (A_2 \star B_2)}\,\to \mathsf{R}$

Case $\mathcal{D} = \dfrac{\overset{\mathcal{D}'}{\vdash (A \star B) \to C}}{\vdash A \to (B \mathrel{-\!\!\star} C)}\,\to\!\star H_1$

$\mathcal{G}_1 :: (A \Longrightarrow \cdot), (B \Longrightarrow \cdot); A \star B \to C \Longrightarrow C$

by $\dfrac{\dfrac{\overline{A \Longrightarrow A}\ \mathit{Init}_{\mathcal{S}} \quad \overline{B \Longrightarrow B}\ \mathit{Init}_{\mathcal{S}}}{(A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow A \star B}\ \star R_{\mathcal{S}} \quad \overline{C \Longrightarrow C}\ \mathit{Init}_{\mathcal{S}}}{(A \Longrightarrow \cdot), (B \Longrightarrow \cdot); A \star B \to C \Longrightarrow C}\ \to L_{\mathcal{S}}$

$\mathcal{G}_2 :: \cdot \Longrightarrow A \star B \to C$ by IH on $\mathcal{D}'$

$\mathcal{G} :: (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow C$ by applying Theorem 4.1 to $\mathcal{G}_1$ and $\mathcal{G}_2$

$\cdot \Longrightarrow A \to (B \mathrel{-\!\!\star} C)$ by $\dfrac{\dfrac{\dfrac{\mathcal{G}}{(A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow C}\ \mathit{TP}_{\mathcal{S}}}{A; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}\ \mathrel{-\!\!\star} R_{\mathcal{S}}}{A \Longrightarrow B \mathrel{-\!\!\star} C}\ \to R_{\mathcal{S}}}{\cdot \Longrightarrow A \to (B \mathrel{-\!\!\star} C)}\ \to R_{\mathcal{S}}$

Case $\mathcal{D} = \dfrac{\overset{\mathcal{D}'}{\vdash A \to (B \mathrel{-\!\!\star} C)}}{\vdash (A \star B) \to C}\ \mathrel{-\!\!\star} H_2$

$\mathcal{G}_1 :: \cdot \Longrightarrow A \to (B \mathrel{-\!\!\star} C)$ by IH on $\mathcal{D}'$

$\mathcal{G}_2 :: A; A \to (B \mathrel{-\!\!\star} C); (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot$

by $\dfrac{\dfrac{\overline{A \Longrightarrow A}\ \mathit{Init}_{\mathcal{S}} \quad \dfrac{\overline{B \Longrightarrow B}\ \mathit{Init}_{\mathcal{S}} \quad \overline{C \Longrightarrow C}\ \mathit{Init}_{\mathcal{S}}}{B \mathrel{-\!\!\star} C; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}\ \mathrel{-\!\!\star} L_{\mathcal{S}}}{A; A \to (B \mathrel{-\!\!\star} C); (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}\ \to L_{\mathcal{S}}$

$\mathcal{G} :: A; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot$ by apply Theorem 4.1 to $\mathcal{G}_1$ and $\mathcal{G}_2$

$\cdot \Longrightarrow (A \star B) \to C$ by $\dfrac{\dfrac{\dfrac{\mathcal{G}}{A; (B \Longrightarrow \cdot)\langle \cdot \Longrightarrow C\rangle \Longrightarrow \cdot}\ \mathit{TC}_{\mathcal{S}}}{(A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow C}\ \star L_{\mathcal{S}}}{A \star B \Longrightarrow C}\ \to R_{\mathcal{S}}}{\cdot \Longrightarrow (A \star B) \to C}\ \to R_{\mathcal{S}}$

□

Proof of Theorem 5.2.

$\not\models_{\mathcal{W}} \Gamma \Longrightarrow \Delta$ assumption

$\models \neg[\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}}$ by Proposition 5.4

$\vdash \neg[\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}}$ by Theorem 2.1

$\cdot \Longrightarrow \neg[\![\Gamma \Longrightarrow \Delta]\!]_{\mathrm{w}}$ by Lemma 5.7

$\Gamma \Longrightarrow \Delta$ by Proposition 5.6

□

D Display calculus for Boolean BI

Proof of Lemma 6.3. By induction on the structure of proof $\mathcal{D}$ of $X \vdash_{\mathcal{D}} Y$.

We write $R_{\mathcal{D}}$ to denote the last inference rule of proof $\mathcal{D}$.

Interestingly, for some cases, the induction hypothesis itself already implies the conclusion to be shown. For example, let us consider the case that $R_{\mathcal{D}}$ is $\mathit{AD1a}_{\mathcal{D}}$:

Case $\mathcal{D} = \dfrac{\overset{\mathcal{D}'}{X'; X \vdash_{\mathcal{D}} Y'}}{X \vdash_{\mathcal{D}} \sharp X'; Y'}\ \mathit{AD1a}_{\mathcal{D}}$ where $Y = \sharp X'; Y'$

By the definition of $[\![]\!]_{\mathcal{C}}$, the followings hold:

$$\begin{array}{rcl} [\![X \vdash_{\mathcal{D}} \sharp X'; Y]\!]_{\mathcal{C}} & = & [\![X]\!]_{\mathcal{X}} \uplus [\![X']\!]_{\mathcal{X}} \uplus [\![Y']\!]_{\mathcal{Y}} \\ [\![X'; X \vdash_{\mathcal{D}} Y']\!]_{\mathcal{C}} & = & [\![X']\!]_{\mathcal{X}} \uplus [\![X]\!]_{\mathcal{X}} \uplus [\![Y']\!]_{\mathcal{Y}} \end{array}$$

Note that truth contexts and falsehood contexts are unordered, which means that $[\![X \vdash_{\mathcal{D}} \sharp X'; Y]\!]_{\mathcal{C}}$ and $[\![X'; X \vdash_{\mathcal{D}} Y']\!]_{\mathcal{C}}$ are same. Therefore, we may conclude that $[\![X \vdash_{\mathcal{D}} \sharp X'; Y]\!]_{\mathcal{C}}$ holds.

The proof of most of cases proceeds by applying the corresponding rule in $\mathbf{S_{BBI}}$ to the induction hypothesis. For example, let us consider the case that $R_{\mathcal{D}}$ is $\star R_{\mathcal{D}}$:

$$\text{Case } \mathcal{D} = \dfrac{\overset{\mathcal{D}_1}{X_1 \vdash_{\mathcal{D}} A} \quad \overset{\mathcal{D}_2}{X_2 \vdash_{\mathcal{D}} B}}{X_1, X_2 \vdash_{\mathcal{D}} A \star B}\ \star R_{\mathcal{D}} \quad \text{where } X = X_1, X_2 \text{ and } Y = A \star B$$

Let $[\![X_1]\!]_{\mathcal{X}}$ be $\Gamma_1 \Longrightarrow \Delta_1$ and $[\![X_2]\!]_{\mathcal{X}}$ be $\Gamma_1 \Longrightarrow \Delta_2$, then the proof proceeds as follows:

$\Gamma_1 \Longrightarrow \Delta_1; A$ by IH on $\mathcal{D}_1$
$\Gamma_2 \Longrightarrow \Delta_2; B$ by IH on $\mathcal{D}_2$
$(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B$ by $\star R_{\mathcal{S}}$

For some cases, the proof may require additional steps that apply $EC_{\mathcal{S}}$. For example, let us consider the case the $R_{\mathcal{D}}$ is $\emptyset_{\mathsf{m}}\, U_{\mathcal{D}}$:

$$\text{Case } \mathcal{D} = \dfrac{\overset{\mathcal{D}'}{X_2, X \vdash_{\mathcal{D}} Y}}{X_1 \vdash_{\mathcal{D}} X_2 \multimap Y}\ MD1a_{\mathcal{D}}$$

Let $[\![X_1]\!]_{\mathcal{X}}$ be $\Gamma_1 \Longrightarrow \Delta_1$, $[\![X_2]\!]_{\mathcal{X}}$ be $\Gamma_2 \Longrightarrow \Delta_2$, and $[\![Y]\!]_{\mathcal{Y}}$ be $\Gamma' \Longrightarrow \Delta'$, then the proof proceeds as follows:

$\Gamma'; (\Gamma_2 \Longrightarrow \Delta_2), (\Gamma_1 \Longrightarrow \Delta_1) \Longrightarrow \Delta'$ by IH on $\mathcal{D}'$
$\Gamma'; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta'$ by $EC_{\mathcal{S}}$
$\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2) \langle \Gamma' \Longrightarrow \Delta' \rangle \Longrightarrow \Delta_1$ by $TP_{\mathcal{S}}$

All the remaining cases are similar to one of these cases. □

Lemma D.1. *If* $\Gamma \Longrightarrow \Delta$, *then* $[\![\Gamma]\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta]\!]_{\mathcal{D}}$.

Proof. By induction on the structure of proof of $\Gamma \Longrightarrow \Delta$.

We write $R_{\mathcal{D}}$ to denote the last inference rule of proof $\mathcal{D}$.

The proof proceeds by applying the corresponding rule in $\mathbf{DS_{BBI}}$ to the induction hypothesis with applying several display rules. For example, let us consider that $R_{\mathcal{D}}$ is $\to L_{\mathcal{S}}$:

$$\text{Case } \mathcal{D} = \dfrac{\overset{\mathcal{D}_1}{\Gamma_1 \Longrightarrow \Delta_1; A} \quad \overset{\mathcal{D}_2}{\Gamma_2; B \Longrightarrow \Delta_2}}{\Gamma_1; \Gamma_2; A \to B \Longrightarrow \Delta_1; \Delta_2}\ \to L_{\mathcal{S}}$$

$\mathcal{G}_1 :: [\![\Gamma_1]\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta_1]\!]_{\mathcal{D}}; A$ by IH on $\mathcal{D}_1$
$\mathcal{G}_2 :: [\![\Gamma_2]\!]_{\mathcal{G}}; B \vdash_{\mathcal{D}} [\![\Delta_2]\!]_{\mathcal{D}}$ by IH on $\mathcal{D}_2$
$B \vdash_{\mathcal{D}} \sharp[\![\Gamma_2]\!]_{\mathcal{G}}; [\![\Delta_2]\!]_{\mathcal{D}}$ by the display rules
$A \to B \vdash_{\mathcal{D}} \sharp([\![\Gamma_1]\!]_{\mathcal{G}}; \sharp[\![\Delta_1]\!]_{\mathcal{D}}); (\sharp[\![\Gamma_2]\!]_{\mathcal{G}}; [\![\Delta_2]\!]_{\mathcal{D}})$ by $\to L_{\mathcal{D}}$
$([\![\Gamma_1]\!]_{\mathcal{G}}; [\![\Gamma_2]\!]_{\mathcal{G}}); A \to B \vdash_{\mathcal{D}} [\![\Delta_1]\!]_{\mathcal{D}}; [\![\Delta_2]\!]_{\mathcal{D}}$ by the display rules (with the rule $AAL_{\mathcal{D}}$ and $AAR_{\mathcal{D}}$)

□

Proof of Lemma 6.4.
$\Gamma \Longrightarrow \Delta$ assumption
$[\![\Gamma]\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta]\!]_{\mathcal{D}}$ by Lemma D.1

$[\![\Gamma \Longrightarrow \Delta]\!]_{\mathcal{W}} \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}$ by $\dfrac{\dfrac{[\![\Gamma]\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta]\!]_{\mathcal{D}}}{[\![\Gamma]\!]_{\mathcal{G}} \vdash_{\mathcal{D}} [\![\Delta]\!]_{\mathcal{D}}; \emptyset_{\mathsf{a}}}\ \emptyset_{\mathsf{a}} R_{\mathcal{D}}}{[\![\Gamma \Longrightarrow \Delta]\!]_{\mathcal{W}} \vdash_{\mathcal{D}} \emptyset_{\mathsf{a}}}\ AD2b_{\mathcal{D}}$

□

Proof of Lemma 6.5. By induction on the size of Γ and Δ.

The proof of $[\![[\![\Delta]\!]_{\mathcal{D}}]\!]_{\mathcal{Y}} = \cdot \Longrightarrow \Delta$ is immediate, and the proof of $[\![[\![\Gamma]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} = \Gamma \Longrightarrow \cdot$ is also immediate except for two non-trivial cases. Thus, here, we present the proof of these two non-trivial cases:

Case: $\Gamma = \Gamma'; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2)$

$$\begin{array}{rcl}
[\![[\![\Gamma'; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2)]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} & = & [\![[\![\Gamma']\!]_{\mathcal{G}}; ([\![\Gamma_1]\!]_{\mathcal{G}}; \sharp[\![\Delta_1]\!]_{\mathcal{D}}), ([\![\Gamma_2]\!]_{\mathcal{G}}; \sharp[\![\Delta_2]\!]_{\mathcal{D}})]\!]_{\mathcal{X}} \\
& = & [\![[\![\Gamma']\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus ([\![[\![\Gamma_1]\!]_{\mathcal{G}}; \sharp[\![\Delta_1]\!]_{\mathcal{D}}]\!]_{\mathcal{X}}, [\![[\![\Gamma_2]\!]_{\mathcal{G}}; \sharp[\![\Delta_2]\!]_{\mathcal{D}}]\!]_{\mathcal{X}} \Longrightarrow \cdot) \\
& = & [\![[\![\Gamma']\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus (([\![[\![\Gamma_1]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus [\![[\![\Delta_1]\!]_{\mathcal{D}}]\!]_{\mathcal{Y}}), ([\![[\![\Gamma_2]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus [\![[\![\Delta_2]\!]_{\mathcal{D}}]\!]_{\mathcal{Y}}) \Longrightarrow \cdot) \\
& = & (\Gamma' \Longrightarrow \cdot) \uplus \\
& & \quad (((\Gamma_1 \Longrightarrow \cdot) \uplus (\cdot \Longrightarrow \Delta_1)), ((\Gamma_2 \Longrightarrow \cdot) \uplus (\cdot \Longrightarrow \Delta_2)) \Longrightarrow \cdot) \\
& = & \Gamma'; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \cdot
\end{array}$$

Case: $\Gamma = \Gamma'; (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2 \rangle$

$$\begin{array}{rcl}
[\![[\![\Gamma'; (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2 \rangle]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} & = & [\![[\![\Gamma']\!]_{\mathcal{G}}; \sharp(([\![\Gamma_1]\!]_{\mathcal{G}}; \sharp[\![\Delta_1]\!]_{\mathcal{D}} \multimap (\sharp([\![\Gamma_2]\!]_{\mathcal{G}}; \sharp[\![\Delta_2]\!]_{\mathcal{D}})))]\!]_{\mathcal{X}} \\
& = & [\![[\![\Gamma']\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus ([\![[\![\Gamma_1]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus [\![[\![\Delta_1]\!]_{\mathcal{D}}]\!]_{\mathcal{Y}} \langle [\![[\![\Gamma_2]\!]_{\mathcal{G}}]\!]_{\mathcal{X}} \uplus [\![[\![\Delta_2]\!]_{\mathcal{D}}]\!]_{\mathcal{Y}} \rangle \Longrightarrow \cdot) \\
& = & (\Gamma' \Longrightarrow \cdot) \uplus \\
& & \quad (((\Gamma_1 \Longrightarrow \cdot) \uplus (\cdot \Longrightarrow \Delta_1))\langle (\Gamma_2 \Longrightarrow \cdot) \uplus (\cdot \Longrightarrow \Delta_2) \rangle \Longrightarrow \cdot) \\
& = & \Gamma'; (\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2 \rangle \Longrightarrow \cdot
\end{array}$$

□

E Admissibility of weakening and contraction in $\mathbf{CS_{BBI}}$

This section proves the admissibility of weakening and contraction in $\mathbf{CS_{BBI}}$ (Theorems E.1 and E.2) from which Theorem 7.1 follows.

Theorem E.1 (Admissibility of weakening)**.**
If $\Gamma \Longrightarrow \Delta$ is provable in $\mathbf{CS_{BBI}}$, then both $\Gamma; S \Longrightarrow \Delta$ and $\Gamma \Longrightarrow \Delta; A$ are also provable in $\mathbf{CS_{BBI}}$.

Theorem E.2 (Admissibility of contraction)**.**
If $\Gamma; S; S \Longrightarrow \Delta$ is provable in $\mathbf{CS_{BBI}}$, $\Gamma; S \Longrightarrow \Delta$ is also provable in $\mathbf{CS_{BBI}}$.
If $\Gamma \Longrightarrow \Delta; A; A$ is provable in $\mathbf{CS_{BBI}}$, $\Gamma \Longrightarrow \Delta; A$ is also provable in $\mathbf{CS_{BBI}}$.

We take an indirect approach to proving the admissibility of weakening and contraction. We first formulate another nested sequent calculus $\mathbf{vCS_{BBI}}$. We then prove the admissibility of weakening and contraction in $\mathbf{vCS_{BBI}}$. We finally show the admissibility of weakening and contraction in $\mathbf{CS_{BBI}}$ by showing the equivalence between $\mathbf{CS_{BBI}}$ and $\mathbf{vCS_{BBI}}$.

Figure 21 shows the nested sequent calculus $\mathbf{vCS_{BBI}}$ which is obtained by embedding the rule $EC_{\mathcal{C}}$ into all the other rules in $\mathbf{CS_{BBI}}$. Embedding the rule $EC_{\mathcal{C}}$ spawns four new rules from the rule $EA_{\mathcal{C}}$, and two new rules from each of the rules $\emptyset_{\mathsf{m}}D_{\mathcal{C}}$, $TC_{\mathcal{C}}$, $\star R_{\mathcal{C}}$.

Lemma E.3 (Inversion in $\mathbf{vCS_{BBI}}$)**.**

1. *If $\mathcal{D} :: \omega[\Gamma; \neg A \Longrightarrow \Delta]$ is provable in $\mathbf{vCS_{BBI}}$, then $\mathcal{E} :: \omega[\Gamma \Longrightarrow \Delta; A]$ is also provable in $\mathbf{vCS_{BBI}}$ where $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.*
2. *If $\mathcal{D} :: \omega[\Gamma \Longrightarrow \Delta; \neg A]$ is provable in $\mathbf{vCS_{BBI}}$, then $\mathcal{E} :: \omega[\Gamma; A \Longrightarrow \Delta]$ is also provable in $\mathbf{vCS_{BBI}}$ where $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.*
3. *If $\mathcal{D} :: \omega[\Gamma; A \vee B \Longrightarrow \Delta]$ is provable in $\mathbf{vCS_{BBI}}$, then both $\begin{cases} \mathcal{E}_1 :: \omega[\Gamma; A \Longrightarrow \Delta] \\ \mathcal{E}_2 :: \omega[\Gamma; B \Longrightarrow \Delta] \end{cases}$ are also provable in $\mathbf{vCS_{BBI}}$ where $\|\mathcal{E}_i\| \leq \|\mathcal{D}\|$ ($i = 1, 2$).*
4. *If $\mathcal{D} :: \omega[\Gamma \Longrightarrow \Delta; A \vee B]$ is provable in $\mathbf{vCS_{BBI}}$, then $\mathcal{E} :: \omega[\Gamma \Longrightarrow \Delta; A; B]$ is also provable in $\mathbf{vCS_{BBI}}$ where $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.*
5. *If $\mathcal{D} :: \omega[\Gamma; \mathsf{I} \Longrightarrow \Delta]$ is provable in $\mathbf{vCS_{BBI}}$, then $\mathcal{E} :: \omega[\Gamma; \emptyset_{\mathsf{m}} \Longrightarrow \Delta]$ is also provable in $\mathbf{vCS_{BBI}}$ where $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.*
6. *If $\mathcal{D} :: \omega[\Gamma; A \star B \Longrightarrow \Delta]$ is provable in $\mathbf{vCS_{BBI}}$, then $\mathcal{E} :: \omega[\Gamma; (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow \Delta]$ is also provable in $\mathbf{vCS_{BBI}}$ where $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.*

Structural rules:

$$\left.\begin{array}{c}\dfrac{\Gamma;(\Gamma';W_1,W_2\Longrightarrow\Delta'),W_3;W_1',(W_2',W_3'\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;(\Gamma';W_1,W_2\Longrightarrow\Delta'),W_3\Longrightarrow\Delta}\ EA^1_{\mathcal{C}_v}\\[2ex] \dfrac{\Gamma;W_3,(\Gamma';W_1,W_2\Longrightarrow\Delta');W_1',(W_2',W_3'\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;W_3,(\Gamma';W_1,W_2\Longrightarrow\Delta')\Longrightarrow\Delta}\ EA^2_{\mathcal{C}_v}\end{array}\right\}\ \text{where}\ \left\{\begin{array}{l}W_1'=W_1\oplus W_2\langle\Gamma';W_3\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta'\rangle\\ W_2'=W_2\oplus W_1\langle\Gamma';W_3\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta'\rangle\\ W_3'=W_3\oplus(\Gamma';W_1,W_2\Longrightarrow\Delta')\langle\Gamma\Longrightarrow\Delta\rangle\end{array}\right.$$

$$\left.\begin{array}{c}\dfrac{\Gamma;(\Gamma';W_2,W_1\Longrightarrow\Delta'),W_3;W_1',(W_2',W_3'\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;(\Gamma';W_2,W_1\Longrightarrow\Delta'),W_3\Longrightarrow\Delta}\ EA^3_{\mathcal{C}_v}\\[2ex] \dfrac{\Gamma;W_3,(\Gamma';W_2,W_1\Longrightarrow\Delta');W_1',(W_2',W_3'\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;W_3,(\Gamma';W_2,W_1\Longrightarrow\Delta')\Longrightarrow\Delta}\ EA^4_{\mathcal{C}_v}\end{array}\right\}\ \text{where}\ \left\{\begin{array}{l}W_1'=W_1\oplus W_2\langle\Gamma';W_3\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta'\rangle\\ W_2'=W_2\oplus W_1\langle\Gamma';W_3\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta'\rangle\\ W_3'=W_3\oplus(\Gamma';W_2,W_1\Longrightarrow\Delta')\langle\Gamma\Longrightarrow\Delta\rangle\end{array}\right.$$

$$\dfrac{\Gamma;(\Gamma\Longrightarrow\Delta),(\emptyset_{\mathsf{m}}\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma\Longrightarrow\Delta}\ \emptyset_{\mathsf{m}}U_{\mathcal{C}_v}$$

$$\left.\begin{array}{c}\dfrac{\Gamma;\Gamma_{\mathrm{c}1};(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1}),(\Gamma_{\mathrm{c}2};\emptyset_{\mathsf{m}}\Longrightarrow\Delta_{\mathrm{c}2});S\Longrightarrow\Delta;\Delta_{\mathrm{c}1}}{\Gamma;(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1}),(\Gamma_{\mathrm{c}2};\emptyset_{\mathsf{m}}\Longrightarrow\Delta_{\mathrm{c}2})\Longrightarrow\Delta}\ \emptyset_{\mathsf{m}}D^1_{\mathcal{C}_v}\\[2ex] \dfrac{\Gamma;\Gamma_{\mathrm{c}1};(\Gamma_{\mathrm{c}2};\emptyset_{\mathsf{m}}\Longrightarrow\Delta_{\mathrm{c}2}),(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1});S\Longrightarrow\Delta;\Delta_{\mathrm{c}1}}{\Gamma;(\Gamma_{\mathrm{c}2};\emptyset_{\mathsf{m}}\Longrightarrow\Delta_{\mathrm{c}2}),(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1})\Longrightarrow\Delta}\ \emptyset_{\mathsf{m}}D^2_{\mathcal{C}_v}\end{array}\right\}\ \text{where}\ S=(\Gamma_{\mathrm{c}2};\emptyset_{\mathsf{m}}\Longrightarrow\Delta_{\mathrm{c}2})\langle\Gamma\Longrightarrow\Delta\rangle$$

Traverse rules:

$$\dfrac{\Gamma_{\mathrm{c}1};(\Gamma_{\mathrm{c}2}\Longrightarrow\Delta_{\mathrm{c}2})\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta_{\mathrm{c}1}}{\Gamma;(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1}),(\Gamma_{\mathrm{c}2}\Longrightarrow\Delta_{\mathrm{c}2})\Longrightarrow\Delta}\ TC^1_{\mathcal{C}_v}\qquad \dfrac{\Gamma_{\mathrm{c}2};(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1})\langle\Gamma\Longrightarrow\Delta\rangle\Longrightarrow\Delta_{\mathrm{c}2}}{\Gamma;(\Gamma_{\mathrm{c}1}\Longrightarrow\Delta_{\mathrm{c}1}),(\Gamma_{\mathrm{c}2}\Longrightarrow\Delta_{\mathrm{c}2})\Longrightarrow\Delta}\ TC^2_{\mathcal{C}_v}$$

$$\dfrac{\Gamma_{\mathrm{p}};(\Gamma\Longrightarrow\Delta),(\Gamma_{\mathrm{s}}\Longrightarrow\Delta_{\mathrm{s}})\Longrightarrow\Delta_{\mathrm{p}}}{\Gamma;(\Gamma_{\mathrm{s}}\Longrightarrow\Delta_{\mathrm{s}})\langle\Gamma_{\mathrm{p}}\Longrightarrow\Delta_{\mathrm{p}}\rangle\Longrightarrow\Delta}\ TP_{\mathcal{C}_v}$$

Logical rules:

$$\dfrac{}{\Gamma;A\Longrightarrow\Delta;A}\ Init_{\mathcal{C}_v}\qquad \dfrac{}{\Gamma;\bot\Longrightarrow\Delta}\ \bot L_{\mathcal{C}_v}\qquad \dfrac{\Gamma\Longrightarrow\Delta;A}{\Gamma;\neg A\Longrightarrow\Delta}\ \neg L_{\mathcal{C}_v}\qquad \dfrac{\Gamma;A\Longrightarrow\Delta}{\Gamma\Longrightarrow\Delta;\neg A}\ \neg R_{\mathcal{C}_v}$$

$$\dfrac{\Gamma;A\Longrightarrow\Delta\quad\Gamma;B\Longrightarrow\Delta}{\Gamma;A\vee B\Longrightarrow\Delta}\ \vee L_{\mathcal{C}_v}\qquad \dfrac{\Gamma\Longrightarrow\Delta;A;B}{\Gamma\Longrightarrow\Delta;A\vee B}\ \vee R_{\mathcal{C}_v}\qquad \dfrac{\Gamma;\emptyset_{\mathsf{m}}\Longrightarrow\Delta}{\Gamma;\mathsf{I}\Longrightarrow\Delta}\ \mathsf{I}L_{\mathcal{C}_v}\qquad \dfrac{}{\Gamma;\emptyset_{\mathsf{m}}\Longrightarrow\Delta;\mathsf{I}}\ \mathsf{I}R_{\mathcal{C}_v}$$

$$\dfrac{\Gamma;(\Gamma_1\Longrightarrow\Delta_1;A),(\Gamma_2\Longrightarrow\Delta_2)\Longrightarrow\Delta;A\star B\quad \Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2\Longrightarrow\Delta_2;B)\Longrightarrow\Delta;A\star B}{\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2\Longrightarrow\Delta_2)\Longrightarrow\Delta;A\star B}\ \star R^1_{\mathcal{C}_v}\qquad \dfrac{\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2\Longrightarrow\Delta_2;A)\Longrightarrow\Delta;A\star B\quad \Gamma;(\Gamma_1\Longrightarrow\Delta_1;B),(\Gamma_2\Longrightarrow\Delta_2)\Longrightarrow\Delta;A\star B}{\Gamma;(\Gamma_1\Longrightarrow\Delta_1),(\Gamma_2\Longrightarrow\Delta_2)\Longrightarrow\Delta;A\star B}\ \star R^2_{\mathcal{C}_v}$$

$$\dfrac{\Gamma;(A\Longrightarrow\cdot),(B\Longrightarrow\cdot)\Longrightarrow\Delta}{\Gamma;A\star B\Longrightarrow\Delta}\ \star L_{\mathcal{C}_v}$$

$$\dfrac{\Gamma;(A\Longrightarrow\cdot)\langle\cdot\Longrightarrow B\rangle\Longrightarrow\Delta}{\Gamma\Longrightarrow\Delta;A\mathrel{-\!\!\star}B}\ \mathrel{-\!\!\star}R_{\mathcal{C}_v}\qquad \dfrac{\Gamma;(\Gamma_1\Longrightarrow\Delta_1;A)\langle\Gamma_2\Longrightarrow\Delta_2\rangle;A\mathrel{-\!\!\star}B\Longrightarrow\Delta\quad \Gamma;(\Gamma_1\Longrightarrow\Delta_1)\langle\Gamma_2;B\Longrightarrow\Delta_2\rangle;A\mathrel{-\!\!\star}B\Longrightarrow\Delta}{\Gamma;(\Gamma_1\Longrightarrow\Delta_1)\langle\Gamma_2\Longrightarrow\Delta_2\rangle;A\mathrel{-\!\!\star}B\Longrightarrow\Delta}\ \mathrel{-\!\!\star}L_{\mathcal{C}_v}$$

Figure 21: Nested sequent Calculus $\mathbf{vCS_{BBI}}$

7. *If* $\mathcal{D}::\omega[\Gamma\Longrightarrow\Delta;A\mathrel{-\!\!\star}B]$ *is provable in* $\mathbf{vCS_{BBI}}$*, then* $\mathcal{E}::\omega[\Gamma;(A\Longrightarrow\cdot)\langle\cdot\Longrightarrow B\rangle\Longrightarrow\Delta]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *where* $\|\mathcal{E}\|\leq\|\mathcal{D}\|$.

Proposition E.5 states the admissibility of the rule $EC_{\mathcal{C}}$ in $\mathbf{vCS_{BBI}}$. Proposition E.5 follows from Lemma E.4, which is a generalization of Proposition E.5.

Lemma E.4.
If $\mathcal{D}::\omega[\Gamma;W_1,W_2\Longrightarrow\Delta]$ *is provable in* $\mathbf{vCS_{BBI}}$,
then $\mathcal{E}::\omega[\Gamma;W_2,W_1\Longrightarrow\Delta]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *and* $\|\mathcal{E}\|\leq\|\mathcal{D}\|$.

Proposition E.5 (Admissibility of the rule $EC_{\mathcal{C}}$ in $\mathbf{vCS_{BBI}}$)**.**
If $\Gamma;W_1,W_2\Longrightarrow\Delta$ *is provable in* $\mathbf{vCS_{BBI}}$*, then* $\Gamma;W_2,W_1\Longrightarrow\Delta$ *is also provable in* $\mathbf{vCS_{BBI}}$.

The proof of Lemma E.4 proceeds by induction on the size of the proof of $\omega[\Gamma;W_1,W_2\Longrightarrow\Delta]$. The proof is straightforward, but we resort to induction on the size of proofs because of those rules whose premise duplicate the contexts from their conclusion, such as the rule $\emptyset_{\mathsf{m}}U_{\mathcal{C}_v}$. For example, consider the

following case in which the rule $\emptyset_{\mathsf{m}}U_{\mathcal{C}_v}$ is the last inference rule in the proof of $\omega[\Gamma; W_1, W_2 \Longrightarrow \Delta]$:

$$\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma; W_1, W_2; (\Gamma; W_1, W_2 \Longrightarrow \Delta), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta}{\Gamma; W_1, W_2 \Longrightarrow \Delta}\ \emptyset_{\mathsf{m}}U_{\mathcal{C}_v} \quad \text{where } \omega = []$$

Note that the premise contains two copies of W_1, W_2. Thus we need to apply the induction hypothesis twice, which necessitates the induction on the size of proofs.

We then prove the admissibility of weakening and contraction in $\mathbf{vCS_{BBI}}$. We prove first the admissibility of weakening and then the admissibility of contraction. Proposition E.7 states the admissibility of weakening in $\mathbf{vCS_{BBI}}$ and follows from Lemma E.6.

Lemma E.6.
If $\mathcal{D} :: \omega[\Gamma \Longrightarrow \Delta]$ *is provable in* $\mathbf{vCS_{BBI}}$,
then both $\begin{cases} \mathcal{E}_1 :: \omega[\Gamma; S \Longrightarrow \Delta] \\ \mathcal{E}_2 :: \omega[\Gamma \Longrightarrow \Delta; A] \end{cases}$ *are also provable in* $\mathbf{vCS_{BBI}}$ *and* $\|\mathcal{E}_i\| \leq \|\mathcal{D}\|$ *(*$i = 1, 2$*).*

Proposition E.7 (Admissibility of weakening in $\mathbf{vCS_{BBI}}$)**.**
If $\Gamma \Longrightarrow \Delta$ *is provable in* $\mathbf{vCS_{BBI}}$, *then both* $\Gamma; S \Longrightarrow \Delta$ *and* $\Gamma \Longrightarrow \Delta; A$ *are also provable in* $\mathbf{vCS_{BBI}}$.

The proof of Lemma E.6 proceeds by induction on the size of the proof of $\omega[\Gamma \Longrightarrow \Delta]$. The proof is also straightforward, similarly to the proof of Lemma E.4.

Proposition E.9 states the admissibility of contraction in $\mathbf{vCS_{BBI}}$ and follows from Lemma E.8.

Lemma E.8.

1. *If* $\mathcal{D} :: \omega[\Gamma; S; S \Longrightarrow \Delta]$ *is provable in* $\mathbf{vCS_{BBI}}$, *then* $\mathcal{E} :: \omega[\Gamma; S \Longrightarrow \Delta]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *and* $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.
2. *If* $\mathcal{D} :: \omega[\Gamma_{\mathrm{c}1}; W_{\mathrm{c}2}\langle \Gamma; W_{\mathrm{c}1}, W_{\mathrm{c}2} \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{\mathrm{c}2}]$ *is provable in* $\mathbf{vCS_{BBI}}$, *then* $\mathcal{E} :: \omega[\Gamma_{\mathrm{c}1}; W_{\mathrm{c}2}\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{\mathrm{c}2}]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *and* $\|\mathcal{E}\| \leq \|\mathcal{D}\|$ *where* $\begin{cases} W_{\mathrm{c}1} = \Gamma_{\mathrm{c}1} \Longrightarrow \Delta_{\mathrm{c}1} \\ W_{\mathrm{c}2} = \Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2} \end{cases}$.
3. *If* $\mathcal{D} :: \omega[\Gamma_{\mathrm{p}}; (\Gamma; W_{\mathrm{s}}\langle W_{\mathrm{p}}\rangle \Longrightarrow \Delta), W_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{p}}]$ *is provable in* $\mathbf{vCS_{BBI}}$, *then* $\mathcal{E} :: \omega[\Gamma_{\mathrm{p}}; (\Gamma \Longrightarrow \Delta), W_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{p}}]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *and* $\|\mathcal{E}\| \leq \|\mathcal{D}\|$ *where* $\begin{cases} W_{\mathrm{s}} = \Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}} \\ W_{\mathrm{p}} = \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \end{cases}$.
4. *If* $\mathcal{D} :: \omega[\Gamma_{\mathrm{s}}; (\Gamma; W_{\mathrm{s}}\langle W_{\mathrm{p}}\rangle \Longrightarrow \Delta)\langle W_{\mathrm{p}}\rangle \Longrightarrow \Delta_{\mathrm{s}}]$ *is provable in* $\mathbf{vCS_{BBI}}$, *then* $\mathcal{E} :: \omega[\Gamma_{\mathrm{s}}; (\Gamma \Longrightarrow \Delta)\langle W_{\mathrm{p}}\rangle \Longrightarrow \Delta_{\mathrm{s}}]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *and* $\|\mathcal{E}\| \leq \|\mathcal{D}\|$ *where* $\begin{cases} W_{\mathrm{s}} = \Gamma_{\mathrm{s}} \Longrightarrow \Delta_{\mathrm{s}} \\ W_{\mathrm{p}} = \Gamma_{\mathrm{p}} \Longrightarrow \Delta_{\mathrm{p}} \end{cases}$.
5. *If* $\mathcal{D} :: \omega[\Gamma \Longrightarrow \Delta; A; A]$ *is provable in* $\mathbf{vCS_{BBI}}$, *then* $\mathcal{E} :: \omega[\Gamma \Longrightarrow \Delta; A]$ *is also provable in* $\mathbf{vCS_{BBI}}$ *where* $\|\mathcal{E}\| \leq \|\mathcal{D}\|$.

Proposition E.9 (Admissibility of contraction $\mathbf{vCS_{BBI}}$)**.**
If $\Gamma; S; S \Longrightarrow \Delta$ *is provable in* $\mathbf{vCS_{BBI}}$, $\Gamma; S \Longrightarrow \Delta$ *is also provable in* $\mathbf{vCS_{BBI}}$.
If $\Gamma \Longrightarrow \Delta; A; A$ *is provable in* $\mathbf{vCS_{BBI}}$, $\Gamma \Longrightarrow \Delta; A$ *is also provable in* $\mathbf{vCS_{BBI}}$.

The proof of Lemma E.8 proceeds by simultaneous induction on the size of proofs $\mathcal{D}$ and $\mathcal{E}$. We prove Lemma E.8 by simultaneously proving these five independent clauses in it. For example, let us consider the following case in which the rule $TC^1_{\mathcal{C}_v}$ is the last inference rule in the proof of $\omega[\Gamma; S; S \Longrightarrow \Delta]$ when proving the first clause:

$$\mathcal{D} = \dfrac{\Gamma_{\mathrm{c}1}; (\Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2})\langle \Gamma; S \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{\mathrm{c}1}}{\Gamma; S; S \Longrightarrow \Delta}\ TC^1_{\mathcal{C}_v} \quad \text{where } \begin{cases} \omega = [] \\ S = (\Gamma_{\mathrm{c}1} \Longrightarrow \Delta_{\mathrm{c}1}), (\Gamma_{\mathrm{c}2} \Longrightarrow \Delta_{\mathrm{c}2}) \end{cases}$$

It is easy to see that we need the second clause to complete the proof. The proof of all the other clauses is also similar.

The proof of Lemma E.8 uses Lemmas E.4 and E.6. As an example of using Lemma E.4, let us consider the following case in which the rule $TC^2_{\mathcal{C}_v}$ is the last inference rule in the proof of $\omega[\Gamma; S; S \Longrightarrow \Delta]$ when proving the first clause:

$$\mathcal{D} = \dfrac{\mathcal{D}' :: \Gamma_{c2}; (\Gamma_{c1} \Longrightarrow \Delta_{c1})\langle \Gamma; S \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{c2}}{\Gamma; S; S \Longrightarrow \Delta}\, TC^2_{\mathcal{C}_v} \quad \text{where} \left\{ \begin{array}{l} \omega = [] \\ S = (\Gamma_{c1} \Longrightarrow \Delta_{c1}), (\Gamma_{c2} \Longrightarrow \Delta_{c2}) \end{array} \right.$$

The proof proceeds as follows:

$\mathcal{D}'' :: \Gamma_{c2}; (\Gamma_{c1} \Longrightarrow \Delta_{c1})\langle \Gamma; S' \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{c2}$
by Lemma E.4 where $S' = (\Gamma_{c2} \Longrightarrow \Delta_{c2}), (\Gamma_{c1} \Longrightarrow \Delta_{c1})$

$\Gamma_{c2}; (\Gamma_{c1} \Longrightarrow \Delta_{c1})\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta_{c2}$ by IH (from the second lemma) on $\mathcal{D}''$

$\Gamma; W_{c1}, W_{c2} \Longrightarrow \Delta$ by the rule $TC^R_{\mathcal{C}}$

As an example of using Lemma E.6, let us consider the following case in which the rule $\star R^1_{\mathcal{C}_v}$ is the last inference rule in $\omega[\Gamma; S; S \Longrightarrow \Delta]$ when proving the first clause:

$$\mathcal{D} = \dfrac{\mathcal{D}_1 :: \Gamma; S; S_1 \Longrightarrow \Delta'; A \star B \quad \mathcal{D}_2 :: \Gamma; S; S_2 \Longrightarrow \Delta'; A \star B}{\Gamma; S; S \Longrightarrow \Delta'; A \star B}\, \star R^1_{\mathcal{C}_v}$$

$$\text{where} \left\{ \begin{array}{l} \omega = [] \\ \Delta = \Delta'; A \star B \\ S = (\Gamma_{c1} \Longrightarrow \Delta_{c1}), (\Gamma_{c2} \Longrightarrow \Delta_{c2}) \\ S_1 = (\Gamma_{c1} \Longrightarrow \Delta_{c1}; A), (\Gamma_{c2} \Longrightarrow \Delta_{c2}) \\ S_2 = (\Gamma_{c1} \Longrightarrow \Delta_{c1}), (\Gamma_{c2} \Longrightarrow \Delta_{c2}; B) \end{array} \right.$$

The proof proceeds as follows:

$\mathcal{D}'_1 :: \Gamma; S_1; S_1 \Longrightarrow \Delta'; A \star B$ by applying Lemma E.6 on $\mathcal{D}_1$

$\mathcal{D}'_2 :: \Gamma; S_2; S_2 \Longrightarrow \Delta'; A \star B$ by applying Lemma E.6 on $\mathcal{D}_2$

$\mathcal{G}_1 :: \Gamma; S_1 \Longrightarrow \Delta'; A \star B$ by IH (from the first lemma) on $\mathcal{D}'_1$

$\mathcal{G}_2 :: \Gamma; S_2 \Longrightarrow \Delta'; A \star B$ by IH (from the first lemma) on $\mathcal{D}'_2$

$\Gamma; S \Longrightarrow \Delta'; A \star B$ by $\dfrac{\overset{\mathcal{G}_1}{\Gamma; S_1 \Longrightarrow \Delta'; A \star B} \quad \overset{\mathcal{G}_2}{\Gamma; S_2 \Longrightarrow \Delta'; A \star B}}{\Gamma; S \Longrightarrow \Delta'; A \star B}\, \star R_{\mathcal{C}}$

Note that we cannot proceed further without applying Lemma E.6.

Propositions E.10 and E.11 state the soundness and completeness of $\mathbf{vCS_{BBI}}$ with respect to $\mathbf{CS_{BBI}}$. The proof of soundness proceeds by induction on the structure of the proof $\Gamma \Longrightarrow \Delta$ in $\mathbf{vCS_{BBI}}$, and the proof of completeness proceeds by induction on the structure of the proof $\Gamma \Longrightarrow \Delta$ in $\mathbf{CS_{BBI}}$.

Proposition E.10 (Soundness of $\mathbf{vCS_{BBI}}$)**.**
If $\Gamma \Longrightarrow \Delta$ *is provable in* $\mathbf{vCS_{BBI}}$, $\Gamma \Longrightarrow \Delta$ *is also provable in* $\mathbf{CS_{BBI}}$.

Proof. By induction on the structure of the proof $\Gamma \Longrightarrow \Delta$ in $\mathbf{vCS_{BBI}}$. □

Proposition E.11 (Completeness of $\mathbf{vCS_{BBI}}$)**.**
If $\Gamma \Longrightarrow \Delta$ *is provable in* $\mathbf{CS_{BBI}}$, $\Gamma \Longrightarrow \Delta$ *is also provable in* $\mathbf{vCS_{BBI}}$.

Proof. By induction on the structure of the proof $\Gamma \Longrightarrow \Delta$ in $\mathbf{CS_{BBI}}$. □

The proof of Theorems E.1 and E.2 is now is straightforward because of the admissibility of the weakening and contraction rules in $\mathbf{vCS_{BBI}}$ (Proposition E.7 and E.9) and the soundness and completeness of $\mathbf{vCS_{BBI}}$ with respect to $\mathbf{CS_{BBI}}$ (Proposition E.10 and E.11).

F Soundness and completeness of $\mathbf{CS_{BBI}}$

This section proves the soundness and completeness of $\mathbf{CS_{BBI}}$ with respect to $\mathbf{S_{BBI}}$ (Theorem 7.2).

Theorem F.1 (Soundness). *If $\Gamma \Longrightarrow \Delta$ is provable in $\mathbf{CS_{BBI}}$, then $\Gamma \Longrightarrow \Delta$ is also provable in $\mathbf{S_{BBI}}$.*

Theorem F.2 (Completeness). *If $\Gamma \Longrightarrow \Delta$ is provable in $\mathbf{S_{BBI}}$, then $\Gamma \Longrightarrow \Delta$ is also provable in $\mathbf{CS_{BBI}}$.*

We first prove the soundness of $\mathbf{CS_{BBI}}$ by showing that each rule in $\mathbf{CS_{BBI}}$ is derivable in $\mathbf{S_{BBI}}$. It is straightforward to derive every $\mathbf{CS_{BBI}}$ rule except the rule $EA_{\mathcal{C}}$ in $\mathbf{S_{BBI}}$. For example, Figure 22 shows the derivation of the rule $\star R_{\mathcal{C}}$ in $\mathbf{S_{BBI}}$. $WL^*_{\mathcal{S}}$ and $WR^*_{\mathcal{S}}$ in Figure 22 denote the multiple applications of the rules $WL_{\mathcal{S}}$ and $WR_{\mathcal{S}}$, respectively. The derivation of the rule $EA_{\mathcal{C}}$ in $\mathbf{S_{BBI}}$ is also straightforward, but tedious. Here we give a sketch of how to derive it in $\mathbf{S_{BBI}}$:

$$\frac{\Gamma; (\Gamma'; W_1, W_2 \Longrightarrow), W_3; W_1', (W_2', W_3' \Longrightarrow \cdot) \Longrightarrow \Delta}{\Gamma; (\Gamma'; W_1, W_2 \Longrightarrow), W_3 \Longrightarrow \Delta}\ EA_{\mathcal{C}}$$

$$\text{where } W_i' = W_i \oplus S_i (i = 1, 2, 3) \text{ and } \begin{cases} S_1 = W_2\langle \Gamma'; W_3\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta'\rangle \\ S_2 = W_1\langle \Gamma'; W_3\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta'\rangle \\ S_3 = (\Gamma'; W_1, W_2 \Longrightarrow \Delta')\langle \Gamma \Longrightarrow \Delta\rangle \end{cases}$$

The goal is to derive $\Gamma; (\Gamma'; W_1, W_2 \Longrightarrow), W_3 \Longrightarrow \Delta$ in $\mathbf{S_{BBI}}$ from

$$\Gamma; (\Gamma'; W_1, W_2 \Longrightarrow \Delta'), W_3; W_1', (W_2', W_3' \Longrightarrow \cdot) \Longrightarrow \Delta.$$

We first apply the rule $EA_{\mathcal{S}}$ to derive

$$\Gamma; (\Gamma'; W_1, W_2 \Longrightarrow \Delta'), W_3; (W_1', W_2' \Longrightarrow \cdot), W_3' \Longrightarrow \Delta.$$

We then use the weakening rules (in conjunction with the traverse rules) to derive the following sequent which contains two copies of the same node state:

$$\Gamma; (\Gamma'; W_1', W_2' \Longrightarrow \Delta'), W_3'; (\Gamma'; W_1', W_2' \Longrightarrow \Delta'), W_3' \Longrightarrow \Delta$$

We then apply the rule $CL_{\mathcal{S}}$ to produce a sequent equivalent to

$$\Gamma_3; S_3; (\Gamma'; W_1', W_2' \Longrightarrow \Delta')\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta_3.$$

We weaken S_3 to $(\Gamma'; W_1', W_2' \Longrightarrow \Delta')\langle \Gamma \Longrightarrow \Delta\rangle$ and then apply the rule $CL_{\mathcal{S}}$, which results in a sequent equivalent to:

$$\Gamma_1; S_1; W_2'\langle \Gamma'; W_3\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta'\rangle \Longrightarrow \Delta_1$$

Similarly we weaken S_1 to $W_2'\langle \Gamma'; W_3\langle \Gamma \Longrightarrow \Delta\rangle \Longrightarrow \Delta'\rangle$, and then apply the rule $CL_{\mathcal{S}}$ again, which results in a sequent equivalent to:

$$\Gamma_2; S_2; S_2 \Longrightarrow \Delta_2$$

Then applying the rule $CL_{\mathcal{S}}$ results in a world sequent equivalent to the goal.

We prove the completeness of $\mathbf{CS_{BBI}}$ by induction on the structure of the proof of $\Gamma \Longrightarrow \Delta$ in $\mathbf{S_{BBI}}$. The proof uses the admissibility of weakening and contraction (Theorems E.1 and E.2). For example, consider the following case in which the rule $\star R_{\mathcal{S}}$ is the last inference rule in the proof of $\Gamma \Longrightarrow \Delta$ in $\mathbf{S_{BBI}}$:

$$\frac{\begin{array}{cc} \mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma_1 \Longrightarrow \Delta_1; A & \Gamma_2 \Longrightarrow \Delta_2; B \end{array}}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\ \star R_{\mathcal{S}}$$

The goal is to prove $(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B$ in $\mathbf{CS_{BBI}}$. The induction hypothesis allows us to assume that both $W_1 = \Gamma_1 \Longrightarrow \Delta_1; A$ and $W_2 = \Gamma_2 \Longrightarrow \Delta_2; B$ are provable in $\mathbf{CS_{BBI}}$. Let $\mathcal{E}_1$ be the proof of W_1 in $\mathbf{CS_{BBI}}$ and $\mathcal{E}_2$ the proof of W_2 in $\mathbf{CS_{BBI}}$. We then obtain two proofs $\mathcal{G}_1$ and $\mathcal{G}_2$ from $\mathcal{E}_1$ and $\mathcal{E}_2$ by applying Theorem E.1, respectively:

$\mathcal{G}_1 :: \Gamma_1; S_1 \Longrightarrow \Delta_1; A$ where $S_1 = (\Gamma_2 \Longrightarrow \Delta_2)\langle \cdot \Longrightarrow A \star B\rangle$ by Theorem E.1 on $\mathcal{E}_1$.
$\mathcal{G}_2 :: \Gamma_2; S_2 \Longrightarrow \Delta_2; B$ where $S_2 = (\Gamma_1 \Longrightarrow \Delta_1)\langle \cdot \Longrightarrow A \star B\rangle$ by Theorem E.1 on $\mathcal{E}_2$.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{
              \dfrac{
                \dfrac{
                  \dfrac{
                    \dfrac{
                      \dfrac{
                        \dfrac{
                          \dfrac{\Gamma; (\Gamma_1 \Longrightarrow \Delta_1; A), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B}{\Gamma_1; S_1 \Longrightarrow \Delta_1; A}\ TP_{\mathcal{S}}
                          \quad
                          \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2; B) \Longrightarrow \Delta; A \star B}{\Gamma_1; S_3 \Longrightarrow \Delta_1}\ TP_{\mathcal{S}}}{\Gamma_1; S_1; S_3 \Longrightarrow \Delta_1}\ WL_{\mathcal{S}}}{\Gamma; (\Gamma_1; S_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2; B) \Longrightarrow \Delta; A \star B}\ TC_{\mathcal{S}}}{\Gamma; (\Gamma_2 \Longrightarrow \Delta_2; B), (\Gamma_1; S_1 \Longrightarrow \Delta_1) \Longrightarrow \Delta; A \star B}\ EC_{\mathcal{S}}}{\Gamma_2; S_2 \Longrightarrow \Delta_2; B}\ TP_{\mathcal{S}}
                        }{(\Gamma_1; S_1 \Longrightarrow \Delta_1), (\Gamma_2; S_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\ \star R_{\mathcal{S}}
                      }{(\Gamma_1; S_1 \Longrightarrow \Delta_1), (\Gamma_2; S_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B}\ WR^*_{\mathcal{S}}
                    }{\Gamma; (\Gamma_1; S_1 \Longrightarrow \Delta_1), (\Gamma_2; S_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B}\ WL^*_{\mathcal{S}}
                  }{\Gamma; (\Gamma_2; S_2 \Longrightarrow \Delta_2), (\Gamma_1; S_1 \Longrightarrow \Delta_1) \Longrightarrow \Delta; A \star B}\ EC_{\mathcal{S}}
                }{\Gamma_2; S_2; S_2 \Longrightarrow \Delta_2}\ TP_{\mathcal{S}}
              }{\Gamma_2; S_2 \Longrightarrow \Delta_2}\ CL_{\mathcal{S}}
            }{\Gamma; (\Gamma_2 \Longrightarrow \Delta_2), (\Gamma_1; S_1 \Longrightarrow \Delta_1) \Longrightarrow \Delta; A \star B}\ TC_{\mathcal{S}}
          }{\Gamma; (\Gamma_1; S_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B}\ EC_{\mathcal{S}}
        }{\Gamma_1; S_1; S_1 \Longrightarrow \Delta_1}\ TP_{\mathcal{S}}
      }{\Gamma_1; S_1 \Longrightarrow \Delta_1}\ CL_{\mathcal{S}}
    }{\Gamma; (\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow \Delta; A \star B}\ TC_{\mathcal{S}}
  }{}
}{}
$$

$$
\text{where} \begin{cases} S_1 = (\Gamma_2 \Longrightarrow \Delta_2)\langle \Gamma \Longrightarrow \Delta; A \star B \rangle \\ S_2 = (\Gamma_1; S_1 \Longrightarrow \Delta_1)\langle \Gamma \Longrightarrow \Delta; A \star B \rangle \\ S_3 = (\Gamma_2 \Longrightarrow \Delta_2; B)\langle \Gamma \Longrightarrow \Delta; A \star B \rangle \end{cases}
$$

Figure 22: The derivation of the rule $\star R_{\mathcal{C}}$ in $\mathbf{S_{BBI}}$

Finally we complete the proof as follows:

$$
\dfrac{
  \dfrac{\overset{\mathcal{G}_1}{\Gamma_1; S_1 \Longrightarrow \Delta_1; A}}{(\Gamma_1 \Longrightarrow \Delta_1; A), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\ TC_{\mathcal{C}}
  \quad
  \dfrac{\dfrac{\overset{\mathcal{G}_2}{\Gamma_2; S_2 \Longrightarrow \Delta_2; B}}{(\Gamma_2 \Longrightarrow \Delta_2; B), (\Gamma_1 \Longrightarrow \Delta_1) \Longrightarrow A \star B}\ TC_{\mathcal{C}}}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2; B) \Longrightarrow A \star B}\ EC_{\mathcal{C}}
}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\ \star R_{\mathcal{C}}
$$

All the other cases are similar except when the last inference rule is either the rule $CL_{\mathcal{S}}$ or the rule $CR_{\mathcal{S}}$; we use Theorem E.2 in such cases.