Semantic Cut Elimination for the Logic of Bunched Implications and Structural Extensions, Formalized in Coq

Dan Frumin

Bernoulli Institute, University of Groningen.

Contributing authors: d.frumin@rug.nl;

Abstract

The logic of bunched implications (BI) is a substructural logic that forms the backbone of separation logic, the much studied logic for reasoning about heap-manipulating programs. Although the proof theory and metatheory of BI are mathematically involved, the formalization of important metatheoretical results is still incipient. In this paper we present a self-contained formalized, in the Coq proof assistant, proof of a central metatheoretical property of BI: cut elimination for its sequent calculus, as well the extension of cut elimination to sequent calculus with arbitrary structural rules. The presented proof is *semantic*, in the sense that is obtained by interpreting sequents in a particular "universal" model. This results in a more modular and elegant proof than a standard Gentzen-style cut elimination argument, which can be subtle and error-prone in manual proofs for BI. In particular, our semantic approach avoids unnecessary inversions on proof derivations, or the uses of cut reductions and the multi-cut rule. Our prof is modular and also robust. We demonstrate how our method scales to (i) all extensions of BI with arbitrary structural rules, and (ii) an extension with an S4-like $\Box$ modality.

Keywords: cut elimination, bunched implications, interactive theorem proving, Coq, substructural logics

1 Introduction

The logic of bunched implications (BI) [1] is an extension of intuitionistic logic with substructural connectives. BI (and its classical cousin Boolean BI) is known for, among other things, forming a basis for separation logic [2, 3] – a popular program logic for verification of heap-manipulating programs. The BI itself, and many of its important models, are based on the idea that propositions denote ownership of resources and BI includes a *separating conjunction* connective $*$, which signifies ownership of *disjoint* resources [4]. As an adjoint to $*$, BI also includes a *magic wand* connective $-\!\!*$, which is determined by the property

$$A \vdash B -\!\!* C \qquad \Longleftrightarrow \qquad A * B \vdash C.$$

Additionally, BI includes a unit element Emp for the separating conjunction $*$.

Proof theoretically, BI can be formalized in a Gentzen-style sequent calculus, which operates on the judgments of the form $\Delta \vdash A$, where Δ is not merely a multiset of formulas, but a *bunch*: a tree in which leaves are formulae and nodes are connected with either $;$ or $,$ (signifying connecting the resources using $\wedge$ and $*$, respectively). For example, a bunch might be $((a \wedge b) \,;\, c) \,,\, (d \,;\, e)$. Due to this nested structure of bunches, the left rules in the BI sequent calculus can apply deep inside bunches. For example, an instance of the left rule for $\wedge$, specialized to the bunch above, is

$$\frac{((a \,;\, b) \,;\, c) \,,\, (d \,;\, e) \vdash \varphi}{((a \wedge b) \,;\, c) \,,\, (d \,;\, e) \vdash \varphi}$$

That is, $a \wedge b$ got “destructed” into $a \,;\, b$ in the context $([-] \,;\, c) \,,\, (d \,;\, e)$, where $[-]$ signifies a hole that can be filled.

BI treats separating conjunction $*$ (and, hence, $,$) as a substructural connective, that does not admit contraction and weakening (i.e. neither $a \vdash a * a$ nor $a * b \vdash a$ hold), but it retains the usual structural rules for intuitionistic conjunction $\wedge$ (and, hence, $;$). In the sequent calculus, the corresponding structural rules can as well be applied deeply inside bunches. For example, an instance of a contraction rule might look like this:

$$\frac{\big((a \,,\, b) \,;\, (a \,,\, b)\big) \,,\, c \vdash \varphi}{(a \,,\, b) \,,\, c \vdash \varphi}$$

Here we contract the bunch $(a \,,\, b)$ inside the context $[-] \,,\, c$. In BI we have to permit contraction on arbitrary bunches, whereas in intuitionistic logic contraction on individual formulas is sufficient.

As usual, BI includes a *cut rule*, which formalizes the informal process of applying an intermediate lemma in a proof. Similar to the other rules, the cut rule can be applied on a formula deeply nested inside a bunch:

$$\frac{\Delta' \vdash \psi \qquad \Delta(\psi) \vdash \varphi}{\Delta(\Delta') \vdash \varphi}$$

where $\Delta(-)$ is an arbitrary bunch with a hole.

In this paper we study the *cut elimination* property for BI. That is, every sequent that has a proof in BI involves the cut rule also has a proof that is cut-free (i.e. does not use of the cut rule). From a theoretical point of view, cut elimination can be used to show important meta-theoretical properties (subformula property, consistency, conservativity). From a more practical standpoint, cut elimination is an important ingredient in proof search.

Why formalize cut elimination?

Cut elimination is a staple in metatheory of logics. Because of that, the question of cut elimination is often one of the first to be raised, whenever a new logic or a new sequent calculus is proposed. It is then common to prove cut elimination directly, by providing a recursive procedure on derivation trees, potentially using additional measure(s) to prove that this procedure terminates.

Proofs organized along those lines are repetitive, consist of many sub-cases, and include many implicit details (e.g. about the structure of the contexts). As a result, it is not uncommon to see proofs that are "analogous" to known correct proofs of cut elimination for related systems, or proofs that only discuss a couple of cases that are considered illustrative, with the bulk of the proof being left as a (rarely completed) exercise for the reader.

Unfortunately, due to the interplay and complexity of all the details, such informal proofs can be quite risky. In the case of BI, the deep nested structure of bunches and explicit structural rules contribute to the complexity and the level of details. For example, a proof of cut elimination for BI given in [5, Chapter 6] had a gap, that was later fixed in [6]. The issue seems to arise from the treatment of the contraction rule. In presence of explicit contraction a naive approach of pushing each instance of the cut rule up along the derivation tree does not necessarily work. In order to resolve this, the cut rule should be generalized to the *multicut* rule, combining contraction and cut together. Then cut elimination is generalized to multicut elimination, offering a stronger induction hypothesis that can be applied to subproofs. Unfortunately, this generalization was originally done in a way that only works for some of the cases. See [6] for more details.[1]

This is not the only instance of erroneous proofs of cut elimination slipping in. Several sequent calculus formulations for bi-intuitionistic logic were wrongly believed to enjoy cut elimination. These mistakes were later fixed in [9]. Other instances include an incorrect proof of cut elimination for full intuitionistic linear logic, fixed in [10, 11]; an incorrect proof of cut elimination for nested

[1]It is possible to avoid the multicut generalization by using more fine-grained measure functions, see [7] for the case of intuitionistic logic. As another alternative, Brotherston [8] gave a proof of cut elimination for BI by going through a displayed calculus.

sequent systems for modal logic [12], fixed in [13]. While not incorrect in itself, cut elimination for a formulation of the provability logic GL by Sambin and Valentini [14] with explicit structural rules was subject of some controversy until it was resolved in [15].

Semantic cut elimination.

To counterbalance informal pen-and-paper proofs of cut elimination for BI, we provide a fully formalized proof in the Coq proof assistant. However, instead of trying to formalize an intricate Gentzen-style process, as in [6], we approach cut elimination using the ideas of algebraic proof theory: a research area aimed at making tight connections between structural proof theory and algebraic semantics of logics. In our proof we adapt the methods of algebraic semantic cut elimination for linear logic [16, 17], in which cut elimination is obtained by constructing a special model for linear logic that is universal w.r.t. cut-free provability. We believe that this approach to cut elimination is more amendable to formalization and extension, than a direct Gentzen-style proof.

Semantic cut elimination for BI was first developed by Galatos and Jipsen [18], building on their work on residuated frames [19]. Their approach is quite general, and the proof makes heavy use of intermediate structures (the aforementioned residuated frames), which lie in between sequent calculus and algebraic semantics. By contrast, the proof presented here only involves the "syntax" (sequent calculus), and the "semantics" (algebraic models) parts. This leaves us with fewer structures to consider in the formalization.

To demonstrate the modularity of our proof, we extend it to cover two different types of extensions of BI. Firstly, we consider BI extended with a particular class of structural rules (*simple structural rules*), which cover weakening and contraction (both for $\mathbf{,}$ and $\mathbf{;}$), as well as many other kinds of structural rules. Secondly, we consider BI extended with an S4-like $\Box$ modality. In both cases we show that we do not have to make a lot of modifications to the proof, and the modifications that we do have to make are, in a way, systematic.

1.1 Contributions and Outline

The main contributions of this paper are as follows. We present an algebraic proof of cut elimination for BI. Our proof can be seen as a simplification of the Galatos and Jipsen's method [18], without the framework of residuated frames. We demonstrate the modularity of our approach by extending it to cut elimination of BI with an S4-like modality (modalities were not previously considered in the framework of residuated frames). We formalize the results in the Coq proof assistant, which is to our knowledge the first published formalization of cut elimination for BI.

The remained of the paper is structured as follows. In Section 2 we present the main idea behind semantic proofs of cut elimination. In Section 3 and Section 4 we recall the sequence calculus for BI and its (standard) algebraic semantics via BI algebras. In Section 5 we consider when a closure operator on a BI algebra induces a BI subalgebra. We then apply this construction in

Section 6 to obtain a "universal" model for cut-free provability, and use it to prove cut elimination. In the next sections we demonstrate the extensibility of this approach. First, in Section 7 we extend cut elimination to BI with arbitrary *analytic structural rules* – a restricted form of structural rules. Then, in Section 8 we show that this restriction can be lifted, and BI admits cut elimination for all structural rules. Finally, in Section 9 we extend cut elimination to BI with an S4-like modality. We discuss our formalization efforts in Section 10. We discuss related work in Section 11 and conclude in Section 12.

1.2 Formalization

The formalization is available online at:
https://github.com/co-dan/BI-cutelim.
In this paper we specifically refer to the version with git hash d26ff19. Throughout the paper, identifiers in monospaced font (`like this`) accompany statements and proposition. They indicate the names of the statements in the Coq formalization and link to the corresponding place in the online documentation. For example, the link `proves` points to the inductive definition of the BI sequent calculus.

1.3 Publication History

This article is an extended version of a paper presented under the same title at CPP 2022. Compared to the conference version, we have added the following content. Section 7 on extensions of BI with analytic structural rules have been expanded. Section 10 have been greatly expanded and highlights multiple parts of the formalization. Finally, Section 8 on analytic completion of structural rules is completely new, including the formalization.

2 Semantic Cut Elimination

In this section we explain some of the ideas and intuitions behind a semantic proof of cut elimination in a semi-formal way, before diving straight into the complexities of BI. The starting point is that there is a class of algebras in which we can interpret logic. The main idea is to find a particular algebra $\mathcal{C}$, in which we can interpret the sequent calculus, and which has a property that if $[\![\psi]\!] \leq [\![\varphi]\!]$ in $\mathcal{C}$, then $\psi \vdash \varphi$ is derivable without applications of the CUT rule. In this case, we say that $\mathcal{C}$ is a "universal" algebra for cut-free provability. Then, cut elimination can be obtained by the (sound) interpretation of sequent calculus into $\mathcal{C}$.

Finding such a "universal" algebra is reminiscent of proving *completeness* of a logic w.r.t. a class of algebras. In the case of completeness, we construct a "universal" algebra $\mathcal{L}$ such that $[\![\psi]\!] \leq [\![\varphi]\!]$ implies derivability of $\psi \vdash \varphi$. This Lindenbaum-Tarski algebra $\mathcal{L}$ is usually defined to be the collection of equivalence classes of formulas modulo interprovability:

$$[\varphi] \triangleq \{\psi \mid (\psi \vdash \varphi) \ \wedge \ (\varphi \vdash \psi)\}$$

And the ordering $\leq$ on $\mathcal{L}$ is induced by provability:

$$[\varphi] \leq [\psi] \iff \varphi \vdash \psi.$$

Provability does not depend on the representative of the equivalence class, and so we get a poset $\mathcal{L}$. The logical operators are interpreted in $\mathcal{L}$ in such a way that $[\![\varphi]\!] = [\varphi]$. The argument for completeness then goes as follows: suppose that $[\![\varphi]\!] \leq [\![\psi]\!]$ in all the possible algebras; then, in particular that inequality holds in $\mathcal{L}$, which amounts to $\varphi \vdash \psi$. Thus, any valid sequent is derivable.

It is precisely the connection between provability and the order on the algebra that makes this model useful. We can imagine a reformulation of the above model in terms of cut-free provability in sequent calculus:

$$[\varphi] \leq [\psi] \iff \varphi \vdash_{\mathsf{cf}} \psi.$$

This adaptation, however, does not work. In order to prove that the ordering $\leq$ is transitive, we need to show

$$\frac{\varphi \vdash_{\mathsf{cf}} \psi \qquad \psi \vdash_{\mathsf{cf}} \chi}{\varphi \vdash_{\mathsf{cf}} \chi}$$

which amounts to showing that CUT is admissible in the cut-free fragment. We seem to be back at square one.

To fix this, instead of interpreting formulas as sets of equivalent formulas (which is what equivalence classes can be seen as), we would like to interpret formulas as sets of *contexts* which prove the formula:

$$\langle\varphi\rangle \triangleq \{\Delta \mid \Delta \vdash_{\mathsf{cf}} \varphi\}.$$

Then, inclusion of sets is a good candidate for the ordering, because $\psi \in \langle\psi\rangle$ and, hence, $\langle\psi\rangle \subseteq \langle\varphi\rangle$ implies $\psi \vdash_{\mathsf{cf}} \varphi$.

But how do we interpret logical connectives? We can interpret $\top$ as the set of all contexts; then, clearly $\top = \langle\mathsf{True}\rangle$. However, we cannot pick the empty set as an interpretation of $\bot$: the set $\langle\mathsf{False}\rangle$ is non-empty, as it contains at least False itself. What we need is to find an interpretation $[\![-]\!]$ such that $[\![\varphi]\!] \subseteq [\![\psi]\!]$ implies $\varphi \vdash_{\mathsf{cf}} \psi$ (or, equivalently $\varphi \in \langle\psi\rangle$). Okada [16] proposed a sufficient condition for such an interpretation: for any formula φ, $\varphi \in [\![\varphi]\!]$ and $[\![\varphi]\!] \in \langle\varphi\rangle$. Then, the desired property on the interpretation follows via a chain of inclusions:

$$\varphi \in [\![\varphi]\!] \subseteq [\![\psi]\!] \subseteq \langle\psi\rangle.$$

Note that the set of all contexts does not satisfy this condition: as we have seen, the empty set a counter-example. It is the least element w.r.t set inclusion, but $\mathsf{False} \notin \emptyset$, so we cannot set $[\![\mathsf{False}]\!] = \emptyset$. This suggests that, instead of considering arbitrary sets of contexts, we need to refine the powerset algebra somehow. A good starting point would be to consider the carrier of the algebra containing just the sets of the form $\langle\varphi\rangle$, i.e. $\mathcal{C} = \{\langle\varphi\rangle \mid \varphi \in \mathit{Frml}\}$ (by analogy with the Lindenbaum-Tarski algebra, which consists only of elements of the form $[\varphi]$). We can then interpret bottom as $\bot = \langle\mathsf{False}\rangle$, and it indeed will be the least element in the algebra.

Looking at other connectives, we cannot interpret disjunction as set-theoretic union, because the union $\langle\varphi\rangle \cup \langle\psi\rangle$ cannot always be written as $\langle\chi\rangle$, for some formula χ. That is, we cannot actually show that $\mathcal{C}$, as given above, is closed under unions, so $\cup$ is not a well-defined operation on $\mathcal{C}$.

How should we then interpret disjunction if not as the union of sets? If we cannot use the set union, we will use the “next best thing”: the smallest set in $\mathcal{C}$ that actually contains the union. Formally, we set:

$$X \vee Y = \bigcap\{Z \in \mathcal{C} \mid X \cup Y \subseteq Z\}.$$

This definition is still not without issues: for this operation to be defined, we need to ensure that $\mathcal{C}$ is closed under arbitrary intersections. It turns out that we can achieve this by modifying the carrier of $\mathcal{C}$ and “baking in” the closedness under arbitrary intersections. Such a construction is obtained in a generic way as a subalgebra of the powerset algebra generated by a particular *closure operator*, as we will see in Sections 5 and 6.

In the remainder of the paper we develop this construction in details. But first, to make the matters concrete, we recall the BI sequent calculus and properties of its cut-free fragment (Section 3), and the algebraic semantics for BI (Section 4).

3 Sequent Calculus for BI

In this section we briefly recall the sequent calculus formulation of BI [1], and some of the properties of its cut-free fragment. The formulas of BI are obtained from the following grammar:

$$\begin{aligned} \varphi, \psi ::= {} & \mathsf{True} \mid \mathsf{False} \mid \mid \phi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \\ & \mid \mathsf{Emp} \mid \varphi * \psi \mid \varphi \mathrel{-\!\!*} \psi \mid a \qquad (a \in \mathit{Atom}) \end{aligned}$$

BI extends intuitionistic propositional logic with separating conjunction ($*$), magic wand ($-\!\!*$, adjoint to separating conjunction), and the empty proposition (Emp, unit for separating conjunction). We also include atomic propositions drawn from a fixed set *Atom*.

The sequent calculus for BI is given in Figure 1. It operates on the sequents of the form $\Delta \vdash \varphi$, where φ is a formula and Δ is a *bunch* – a tree composed of binary nodes labeled with $\boldsymbol{,}$ and $\boldsymbol{;}$, and leaves being either formulas or empty bunches $\varnothing_m$ and $\varnothing_a$. Morally, we view bunches as equivalence classes of such trees modulo commutative monoid laws for $(\boldsymbol{,}, \varnothing_m)$, and $(\boldsymbol{;}\ , \varnothing_a)$. These are given using structural congruence $\equiv$, the rules for which are also given in

Equivalence of bunches

$$\Delta_1 \,,\, \Delta_2 \equiv \Delta_2 \,,\, \Delta_1 \qquad \Delta_1 \,;\, \Delta_2 \equiv \Delta_2 \,;\, \Delta_1$$

$$\Delta_1 \,,\, (\Delta_2 \,,\, \Delta_3) \equiv (\Delta_1 \,,\, \Delta_2) \,,\, \Delta_3 \qquad \Delta_1 \,;\, (\Delta_2 \,;\, \Delta_3) \equiv (\Delta_1 \,;\, \Delta_2) \,;\, \Delta_3$$

$$\Delta \,,\, \varnothing_m \equiv \Delta \qquad \Delta \,;\, \varnothing_a \equiv \Delta \qquad \frac{\Delta \equiv \Delta'}{\Gamma(\Delta) \equiv \Gamma(\Delta')}$$

Structural rules

$$\overset{\text{AX}}{\frac{a \in Atom}{a \vdash a}} \qquad \overset{\text{EQUIV}}{\frac{\Delta' \vdash \varphi \qquad \Delta \equiv \Delta'}{\Delta \vdash \varphi}} \qquad \overset{\text{W};}{\frac{\Delta(\Delta_1) \vdash \varphi}{\Delta(\Delta_1 \,;\, \Delta_2) \vdash \varphi}}$$

$$\overset{\text{C};}{\frac{\Delta(\Delta_1 \,;\, \Delta_1) \vdash \varphi}{\Delta(\Delta_1) \vdash \varphi}} \qquad \overset{\text{CUT}}{\frac{\Delta' \vdash A \qquad \Delta(A) \vdash B}{\Delta(\Delta') \vdash B}}$$

Multiplicatives

$$\overset{\mathsf{Emp}\text{R}}{\varnothing_m \vdash \mathsf{Emp}} \qquad \overset{\mathsf{Emp}\text{L}}{\frac{\Delta(\varnothing_m) \vdash \varphi}{\Delta(\mathsf{Emp}) \vdash \varphi}} \qquad \overset{*\text{R}}{\frac{\Delta_1 \vdash \varphi \qquad \Delta_2 \vdash \psi}{\Delta_1 \,,\, \Delta_2 \vdash \varphi * \psi}} \qquad \overset{*\text{L}}{\frac{\Delta(\varphi \,,\, \psi) \vdash \chi}{\Delta(\varphi * \psi) \vdash \chi}}$$

$$\overset{-\!\!*\text{R}}{\frac{\Delta \,,\, \varphi \vdash \psi}{\Delta \vdash \varphi -\!\!* \psi}} \qquad \overset{-\!\!*\text{L}}{\frac{\Delta_1 \vdash \varphi \qquad \Delta(\Delta_2 \,,\, \psi) \vdash \chi}{\Delta(\Delta_1 \,,\, \Delta_2 \,,\, \varphi -\!\!* \psi) \vdash \chi}}$$

Additives

$$\overset{\mathsf{True}\text{R}}{\varnothing_a \vdash \mathsf{True}} \qquad \overset{\mathsf{True}\text{L}}{\frac{\Delta(\varnothing_a) \vdash \varphi}{\Delta(\mathsf{True}) \vdash \varphi}} \qquad \overset{\wedge\text{R}}{\frac{\Delta_1 \vdash \varphi \qquad \Delta_2 \vdash \psi}{\Delta_1 \,;\, \Delta_2 \vdash \varphi \wedge \psi}} \qquad \overset{\wedge\text{L}}{\frac{\Delta(\varphi \,;\, \psi) \vdash \chi}{\Delta(\varphi \wedge \psi) \vdash \chi}}$$

$$\overset{\to\text{R}}{\frac{\Delta \,;\, \varphi \vdash \psi}{\Delta \vdash \varphi \to \psi}} \qquad \overset{\to\text{L}}{\frac{\Delta_1 \vdash \varphi \qquad \Delta(\Delta_2 \,;\, \psi) \vdash \chi}{\Delta(\Delta_1 \,;\, \Delta_2 \,;\, \varphi \to \psi) \vdash \chi}} \qquad \overset{\mathsf{False}\text{L}}{\Delta(\mathsf{False}) \vdash \varphi}$$

$$\overset{\vee\text{R1}}{\frac{\Delta \vdash \varphi}{\Delta \vdash \varphi \vee \psi}} \qquad \overset{\vee\text{R2}}{\frac{\Delta \vdash \psi}{\Delta \vdash \varphi \vee \psi}} \qquad \overset{\vee\text{L}}{\frac{\Delta(\varphi) \vdash \chi \qquad \Delta(\psi) \vdash \chi}{\Delta(\varphi \vee \psi) \vdash \chi}}$$

Fig. 1 BI sequent calculus.

Figure 1. We could have defined provability on such equivalence classes, but we opt for using explicit context conversions using EQUIV.

Most of the structural rules and the left rules can be applied to formulas that occur nested inside some bunch with a hole $\Delta(-)$. We refer to such bunches with holes as *bunched contexts*. For example, in the application of the rule $\wedge$L below we use the bunched context $(p \mathbin{,} [-])$:

$$\wedge\mathrm{L}\ \frac{p \mathbin{,} (p \mathbin{;}\ q) \vdash p * q}{p \mathbin{,} (p \wedge q) \vdash p * q.}$$

3.1 Cut-free Provability

Let us write $\Delta \vdash_{\mathsf{cf}} \varphi$ if $\Delta \vdash \varphi$ is derivable *without* the CUT rule. In the rest of this section we prove invertibility of several rules in the cut-free fragment of BI. Those derived rules will be useful to us when constructing the algebraic model in Section 6.

The first observation about the sequent calculus, is that we have formulated the "axiom" rule $\varphi \vdash \varphi$ only for atomic formulas $a \in \mathit{Atom}$. This will significantly simplify some of the proofs (for example, Lemma 3), but does not limit the expressivity of the system, as witness by the following lemma.

Proposition 1 (Identity expansion, `seqcalc_id_ext`). *For every formula φ we can derive a sequent $\varphi \vdash_{\mathsf{cf}} \varphi$.*

Proof By induction on the structure of φ. □

For the construction presented in this paper we need to show that a number of rules are invertible in the cut-free sequent calculus. Specifically, we need to show that $-\!\!*$R, $\rightarrow$R, $*$L, $\wedge$L, EmpL, and TrueL are invertible.

Lemma 2 (`wand_r_inv` and `impl_r_inv`). *The following rules are admissible:*

$$\begin{array}{c} -\!\!*\text{R-INV} \\ \dfrac{\Delta \vdash_{\mathsf{cf}} \varphi -\!\!* \psi}{\Delta \mathbin{,} \varphi \vdash_{\mathsf{cf}} \psi} \end{array} \qquad \begin{array}{c} \rightarrow\text{R-INV} \\ \dfrac{\Delta \vdash_{\mathsf{cf}} \varphi \rightarrow \psi}{\Delta \mathbin{;}\ \varphi \vdash_{\mathsf{cf}} \psi} \end{array}$$

Proof By induction on the derivations $\Delta \vdash_{\mathsf{cf}} \varphi -\!\!* \psi$ and $\Delta \vdash_{\mathsf{cf}} \varphi \rightarrow \psi$. □

At the end of the day, the proof of Lemma 2 by induction on derivations is not very complicated, because the form of the context on the left-hand side of the sequent is relatively simple. It is easy to show that the left rules can commute with $-\!\!*$R and $\rightarrow$R. By contrast, showing that the rules $*$L and $\wedge$L are invertible is more involved for several reasons.

First of all, just like for other sequent calculi with explicit contraction, structural induction on the proof is not strong enough. Consider the following derivation of $\varphi * \psi \vdash_{\mathsf{cf}} \chi$:

$$\frac{\varphi * \psi \,;\, \varphi * \psi \vdash_{\mathsf{cf}} \chi}{\varphi * \psi \vdash_{\mathsf{cf}} \chi}$$

Since $\varphi*\psi$ occurs twice in the premise, we need to apply the induction hypothesis twice. From the first application of the induction hypothesis we get a proof

$$\varphi * \psi \,;\, (\varphi \,,\, \psi) \vdash_{\mathsf{cf}} \chi,$$

but this proof is not a strict subderivation of the original derivation. Therefore, we cannot use the induction hypothesis second time to obtain a proof of $(\varphi \,,\, \psi) \,;\, (\varphi \,,\, \psi) \vdash_{\mathsf{cf}} \chi$.

In order to circumvent this, we do induction on the *height* of the derivation, strengthening the statement to:

Lemma 3 (sep_l_inv). *If there is a derivation of $\Delta(\varphi * \psi) \vdash_{\mathsf{cf}} \chi$ of height n, then there is a derivation of $\Delta(\varphi \,,\, \psi) \vdash_{\mathsf{cf}} \chi$ of height strictly less than n.*

Similarly, if there is a derivation of $\Delta(\varphi \wedge \psi) \vdash_{\mathsf{cf}} \chi$ of height n, then there is a derivation of $\Delta(\varphi \,;\, \psi) \vdash_{\mathsf{cf}} \chi$ of height strictly less than n.

Note that this lemma would be false if we would have included an axiom rule for arbitrary formulas: there would be a proof $\varphi * \psi \vdash_{\mathsf{cf}} \varphi * \psi$ of height 0, but the smallest proof of $\varphi \,,\, \psi \vdash_{\mathsf{cf}} \varphi * \psi$ is of height 1. That is why we have restricted the axiom rule to atomic formulas, and got the general form of the axiom rule as a derived statement (Proposition 1).

Similarly, by induction on the derivation height, we show that EmpL and TrueL are invertible. We only care about the derivation height for the purposes of induction, so we summarize the results on invertible rules in the following lemma.

Lemma 4. *The following rules are admissible:*

$$\wedge\text{L-INV}\ \frac{\Delta(\varphi \wedge \psi) \vdash_{\mathsf{cf}} \chi}{\Delta(\varphi \,;\, \psi) \vdash_{\mathsf{cf}} \chi} \qquad *\text{L-INV}\ \frac{\Delta(\varphi * \psi) \vdash_{\mathsf{cf}} \chi}{\Delta(\varphi \,,\, \psi) \vdash_{\mathsf{cf}} \chi} \qquad \top\text{L-INV}\ \frac{\Delta(\mathsf{True}) \vdash_{\mathsf{cf}} \chi}{\Delta(\varnothing_a) \vdash_{\mathsf{cf}} \chi} \qquad \mathsf{Emp}\text{L-INV}\ \frac{\Delta(\mathsf{Emp}) \vdash_{\mathsf{cf}} \chi}{\Delta(\varnothing_m) \vdash_{\mathsf{cf}} \chi}$$

Let us write $(\Delta)^*$ for interpretation of Δ as a formula: we substitute every occurrence of $,$ in Δ with $*$, every occurrence of $\varnothing_m$ with Emp, and similarly for the additive connectives. For example, $((\varnothing_m \,;\, \varphi) \,,\, \psi)^*$ is $(\mathsf{Emp} \wedge \varphi) * \psi$. Clearly, there is a derivation from $\Delta \vdash_{\mathsf{cf}} \chi$ to $(\Delta)^* \vdash_{\mathsf{cf}} \chi$, by repeated applications of $*$L, $\wedge$L, TrueL and EmpL. For the other direction we have the following.

Corollary 5 (collapse_l_inv). *The following rule is admissible:*

$$\frac{\Delta'((\Delta)^*) \vdash_{\mathsf{cf}} \chi}{\Delta'(\Delta) \vdash_{\mathsf{cf}} \chi}$$

Proof By induction on Δ, using Lemma 4. □

4 Algebraic Semantics for BI

We interpret the BI sequent calculus in the algebraic structures known as BI algebras, which are bounded Heyting algebras with a compatible residuated monoidal structure.

Definition 6. *A BI algebra $\mathcal{B}$ is a tuple $(B, \bot, \top, \wedge, \vee, \rightarrow, \mathsf{Emp}, *, -\!\!*)$ where*

- $(B, \bot, \top, \wedge, \vee, \rightarrow)$ *is a bounded Heyting algebra, i.e. a bounded distributive lattice with the Heyting implication satisfying*

$$a \wedge b \leq c \iff a \leq b \rightarrow c$$

- $* : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ *is a monotone commutative and associative function;*
- $\mathsf{Emp} : \mathcal{B}$ *is a unit element for* $*$*;*
- $-\!\!* : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ *is a binary operation satisfying*

$$a * b \leq c \iff a \leq b -\!\!* c$$

Definition 7. *Let $\mathcal{B}$ be an arbitrary BI algebra. Given an interpretation $i : Atom \rightarrow \mathcal{B}$ of atomic propositions, we interpret formulas of BI in $\mathcal{B}$ in the usual tautological way:*

$$\begin{aligned}
[\![\mathsf{Emp}]\!] &= \mathsf{Emp} & [\![\mathsf{True}]\!] &= \top \\
[\![\varphi * \psi]\!] &= [\![\varphi]\!] * [\![\psi]\!] & [\![\varphi \wedge \psi]\!] &= [\![\varphi]\!] \wedge [\![\psi]\!] \\
[\![\varphi -\!\!* \psi]\!] &= [\![\varphi]\!] -\!\!* [\![\psi]\!] & [\![\varphi \rightarrow \psi]\!] &= [\![\varphi]\!] \rightarrow [\![\psi]\!] \\
[\![\varphi \vee \psi]\!] &= [\![\varphi]\!] \vee [\![\psi]\!] & [\![\mathsf{False}]\!] &= \bot \\
[\![a]\!] &= i(a)
\end{aligned}$$

Theorem 8 (Soundness, seq_interp_sound). *If $\Delta \vdash \varphi$ is derivable, then $[\![(\Delta)^*]\!] \leq [\![\varphi]\!]$ holds in any BI algebra.*

Proof By induction on the derivation. □

4.1 BI Algebras from Monoids

In practice, a lot of BI algebras arise as predicates over a partial commutative monoid. Let $(M, \cdot, e)$ be a partial commutative monoid, i.e. a commutative monoid for which the multiplication function $\cdot$ is only partially defined, and for which the monoidal laws hold only for defined elements. We write $x \cdot y = \bot$ if composition of x and y is undefined. Then the powerset $\mathcal{P}(M)$ is a (complete) Heyting algebra, and it forms a BI algebra $(\mathcal{P}(M), \emptyset, M, \cap, \cup, \to, \mathbf{0}, \bullet, -\!\!\bullet)$ with the following operators:

$$\begin{aligned}
\mathbf{0} &\triangleq \{e\} \\
X \bullet Y &\triangleq \{x \cdot y \mid x \in X, y \in Y, x \cdot y \neq \bot\} \\
X -\!\!\bullet\, Y &\triangleq \{z \mid \forall x \in X.\, z \cdot x \neq \bot \implies z \cdot x \in Y\}.
\end{aligned}$$

BI algebra from the monoid of contexts.

Let us write *Bunch* for the set of bunches, modulo the equivalence $\equiv$ (from Figure 1). We can endow the set *Bunch* of bunches with the structure of a monoid. Composition of two contexts Δ and Δ' is just putting them next to each other using $\boldsymbol{,}$:

$$\Delta \cdot \Delta' \triangleq (\Delta \boldsymbol{,} \Delta')$$

then, up to equivalence of bunches, $\varnothing_m$ is the unit element. Using the powerset construction we get a BI algebra $\mathcal{P}(\mathit{Bunch})$.

This model is very much "freely generated" from syntax, but it is not very useful, as it does not involve any notion of provability (only equivalence of contexts). In this next sections we are going to refine this model, in order to obtain a submodel which can be used to prove completeness and cut-elimination.

5 Moore Closures on BI Algebras

For cut elimination, we will be interested in subalgebras of the powerset algebra $\mathcal{P}(M)$ for some partial commutative monoid M; specifically subalgebras arising from a particular closure operator. For the rest of this section we fix a partial commutative monoid M.

Definition 9. *A Moore collection is a family of sets $\mathcal{C} \subseteq \mathcal{P}(M)$ that is closed under arbitrary intersections:*

$$(\forall i \in I.\, A_i \in \mathcal{C}) \implies \bigcap_{i \in I} A_i \in \mathcal{C}.$$

If $X \in \mathcal{C}$ we say that X is closed.

Alternatively, a Moore collection can be given in terms of a closure operator $\mathsf{cl}(-)$ satisfying the following conditions:

- $X \subseteq \mathsf{cl}(X)$;
- monotonicity: $X \subseteq Y \implies \mathsf{cl}(X) \subseteq \mathsf{cl}(Y)$;
- idempotence: $\mathsf{cl}(\mathsf{cl}(X)) = \mathsf{cl}(X)$.

Given a Moore collection $\mathcal{C}$ we define the associated closure operator as $\mathsf{cl}(X) = \bigcap\{Y \in \mathcal{C} \mid X \subseteq Y\}$. In the other direction, given a closure operator we define $\mathsf{cl}(-)$-closed sets as $\mathcal{C} = \{X \mid \mathsf{cl}(X) = X\}$.

Some basic theory behind posets with such a closure operator is given in [20]. Here, we recall only the results that we will be needing. First of all, we are going to use the following rule often.

Lemma 10 (`cl_adj`). *The closure operator satisfies the following adjunction rule:*

$$\frac{X \subseteq Y \quad \textit{in } \mathcal{P}(M)}{\mathsf{cl}(X) \subseteq Y \quad \textit{in } \mathcal{C}}$$

for a closed set Y.

Since $\mathcal{C}$ is closed under intersections, $X \cap Y$ is a meet of two closed sets X and Y. However, given two closed sets, their union $X \cup Y$ is not always closed. Instead, we interpret join as $\mathsf{cl}(X \cup Y)$.

Proposition 11. *The collection $\mathcal{C}$ is a complete bounded lattice: the least upper bound is given by $\bigvee_{i \in I} X_i = \mathsf{cl}(\bigcup_{i \in I} X_i)$. In particular, the bottom element of $\mathcal{C}$ is $\mathsf{cl}(\emptyset)$.*

It is not necessarily the case that $\mathcal{C}$ has Heyting implication, but if it does, then we can describe it in terms of the implication on $\mathcal{P}(M)$ and a dual of the closure operator.

Proposition 12 (`impl_from_int`). *For a set $X \in \mathcal{P}(M)$, we write $\mathsf{int}(X)$ for the largest closed set contained in X:*

$$\mathsf{int}(X) = \bigvee\{Y \in \mathcal{C} \mid Y \subseteq X\}.$$

Then for closed sets X and Y,

$$X \to Y = \mathsf{int}(X \supset Y)$$

where $X \supset Y$ denotes implication in $\mathcal{P}(M)$.

Proof We reason as follows:

$$\begin{aligned}\mathsf{int}(X \supset Y) &= \bigvee\{Z \in \mathcal{C} \mid Z \subseteq X \supset Y\} \\ &= \bigvee\{Z \in \mathcal{C} \mid Z \cap X \subseteq Y\} \\ &= \bigvee\{Z \in \mathcal{C} \mid Z \subseteq X \to Y\} = X \to Y\end{aligned}$$

□

In light of the previous propositions, we can see that some Heyting algebra structure on $\mathcal{C}$ arises from the same operations on $\mathcal{P}(M)$. Can we similarly lift the BI operations? Let us denote the residuated monoidal structure (defined as in Section 4.1) on $\mathcal{P}(M)$ as $(\mathbf{0}, \bullet, -\!\bullet)$. In the rest of this section we describe how to lift this structure to $\mathcal{C}$.

5.1 BI Algebra Structure on Closed Sets

A sufficient condition for $\mathcal{C}$ to be a BI algebra is the following.

Definition 13. *We say that the closure operator is* strong *if for any X and Y*

$$\mathsf{cl}(X) \bullet Y \subseteq \mathsf{cl}(X \bullet Y)$$

If $\mathsf{cl}(-)$ is strong, then we define the BI operators on $\mathcal{C}$ as follows:

$$\begin{aligned} \mathsf{Emp} &= \mathsf{cl}(\mathbf{0}) \\ X * Y &= \mathsf{cl}(X \bullet Y) \\ X -\!\!* Y &= \mathsf{cl}(X -\!\bullet Y) \end{aligned}$$

We shall verify that with these connectives $\mathcal{C}$ is a BI algebra.

Proposition 14 (`wand_intro_r`, `wand_elim_l'`)**.** *There is an adjunction between $*$ and $-\!\!*$:*

$$X * Y \subseteq Z \iff X \subseteq Y -\!\!* Z.$$

Proof We reason as follows.

$$\begin{aligned} & X * Y \subseteq Z && \text{(def. of } *) \\ \iff\; & \mathsf{cl}(X \bullet Y) \subseteq Z && (Z \text{ is closed}) \\ \iff\; & X \bullet Y \subseteq Z && \text{(adjunction)} \\ \iff\; & X \subseteq Y -\!\bullet Z && \\ \implies\; & X \subseteq \mathsf{cl}(Y -\!\bullet Z) && \text{(def. of } -\!\!*) \\ \iff\; & X \subseteq Y -\!\!* Z. && \end{aligned}$$

On the other hand,

$$\begin{aligned} & X \subseteq \mathsf{cl}(Y -\!\bullet Z) && \text{(monotonicity of } \bullet) \\ \implies\; & X \bullet Y \subseteq \mathsf{cl}(Y -\!\bullet Z) \bullet Y && \text{(strength of } \mathsf{cl}(-)) \\ \implies\; & X \bullet Y \subseteq \mathsf{cl}((Y -\!\bullet Z) \bullet Y) && \\ \implies\; & X \bullet Y \subseteq \mathsf{cl}(Z) = Z && \\ \iff\; & \mathsf{cl}(X \bullet Y) \subseteq Z. && \end{aligned}$$

□

Proposition 15 (sep_comm', sep_assoc', and emp_sep_1, emp_sep_2). *$(\mathcal{C}, *, \mathsf{Emp})$ is a commutative monoid.*

Proof The commutativity of $*$ is evident from its definition. Let us verify the unit laws:

$$\begin{aligned}\mathsf{Emp} * X &= \mathsf{cl}(\mathsf{cl}(\mathbf{0}) \bullet X) \subseteq \mathsf{cl}(\mathsf{cl}(\mathbf{0} \bullet X)) = X \\ X = \mathbf{0} \bullet X &\subseteq \mathsf{cl}(\mathbf{0}) \bullet X \subseteq \mathsf{cl}(\mathsf{cl}(\mathbf{0}) \bullet X) = \mathsf{Emp} * X.\end{aligned}$$

We reason similarly for associativity of $*$. □

We can summaries these results in the following theorem.

Theorem 16. *Let M be a PCM, and let $\mathsf{cl}(-)$ be a strong closure operator on $\mathcal{P}(M)$, such that $\mathcal{C}$ has Heyting implication. Then the set $\mathcal{C}$ of closed elements is a BI algebra.*

Finally, some times it is more convenient to use an alternative condition in place of closure strength:

Proposition 17. *The closure operator is strong iff $X \mathrel{-\!\!\bullet} Y$ is closed whenever Y is closed, i.e. $\mathcal{C}$ forms an exponential ideal.*

Proof Suppose that $\mathcal{C}$ is an exponential ideal w.r.t $\mathrel{-\!\!\bullet}$. Then we reason as follows:

$$\begin{array}{c} \mathsf{cl}(X) \bullet Y \subseteq \mathsf{cl}(X \bullet Y) \\ \hline \mathsf{cl}(X) \subseteq Y \mathrel{-\!\!\bullet} \mathsf{cl}(X \bullet Y) \\ \hline X \subseteq Y \mathrel{-\!\!\bullet} \mathsf{cl}(X \bullet Y) \qquad \textit{(the r.h.s. is closed)} \\ \hline X \bullet Y \subseteq \mathsf{cl}(X \bullet Y) \end{array}$$

Hence, $\mathsf{cl}(-)$ is strong.

For the other direction, if Y is closed, then

$$\begin{aligned}\mathsf{cl}(X \mathrel{-\!\!\bullet} Y) \subseteq X \mathrel{-\!\!\bullet} Y &\iff \mathsf{cl}(X \mathrel{-\!\!\bullet} Y) \bullet X \subseteq Y \\ &\Longleftarrow \mathsf{cl}((X \mathrel{-\!\!\bullet} Y) \bullet X) \subseteq Y \\ &\iff (X \mathrel{-\!\!\bullet} Y) \bullet X \subseteq Y\end{aligned}$$

□

A remark on (im)predicativity.

In practice, we want to start with some collection $\mathcal{B} \subseteq \mathcal{P}(M)$ of sets, and generate $\mathcal{C}$ freely from arbitrary intersections of elements of $\mathcal{B}$ (think of generating a topology from a closed basis). Then $\mathcal{C}$ is a Moore collection and the associated closure operator can be given as $\mathsf{cl}(X) = \bigcap\{Y \in \mathcal{C} \mid X \subseteq Y\}$. Unfortunately, this definition is impredicative (we define an element of $\mathcal{C}$ by quantifying over elements of $\mathcal{C}$), which, when formalized in type theory, increases the universe level.

That means that we cannot use the closure operator to *define* the set $\mathcal{C}$, i.e. the set $\{X \mid X = \mathsf{cl}(X)\}$ will have a higher universe level than $\mathcal{C}$. To circumvent this, we can instead define the closure operator equivalently by quantifying not over all the closed sets, but only over the basic closed sets: $\mathsf{cl}(X) = \bigcap\{Y \in \mathcal{B} \mid X \subseteq Y\}$. Then we can define $\mathcal{C}$ to be the set of elements satisfying $X = \mathsf{cl}(X)$.

6 Cut-elimination via a Syntactic Model

In this section we construct a special BI algebra $\mathcal{C} \subseteq \mathcal{P}(\mathit{Bunch})$ that has the following property: if $[\![\varphi]\!] \leq [\![\psi]\!]$ holds in $\mathcal{C}$, then $\varphi \vdash_{\mathsf{cf}} \psi$. By composing this with the soundness theorem, we will obtain the cut-elimination result.

6.1 Principal Closed Sets

We are going to construct $\mathcal{C}$ as a particular Moore collection on $\mathcal{P}(\mathit{Bunch})$. To define when a predicate X is closed (e.g. when $X \in \mathcal{C}$), we start with *principal closed elements*, and generate $\mathcal{C}$ as families of intersections of principal closed sets.

Definition 18. *A* principal *closed set is a set of the form:*

$$\langle\varphi\rangle = \{\Delta \mid \Delta \vdash_{\mathsf{cf}} \varphi\}$$

for a formula φ.

We can then generate closed sets by closing the collection of principal closed sets under arbitrary intersections:

$$\mathsf{cl}(X) \triangleq \bigcap\{\langle\varphi\rangle \mid X \subseteq \langle\varphi\rangle\} = \bigcap\{\langle\varphi\rangle \mid \forall\Delta \in X.\, \Delta \vdash_{\mathsf{cf}} \varphi\}.$$

We then define the collection $\mathcal{C}$ of closed sets as

$$\mathcal{C} \triangleq \{X \mid X = \mathsf{cl}(X)\}.$$

Then every element of $\mathcal{C}$ can be written as some intersection $\bigcap_{i \in I}\langle\varphi_i\rangle$.

Let us briefly describe some useful properties of closed sets:

Proposition 19 (`C_inhabited`, `C_weaken`, `C_contract`, and `C_collapse`)**.** *Let X be a closed set. Then the following holds.*

1. $\mathsf{False} \in X$;
2. $\Delta \in X \implies (\Delta \,;\, \Delta') \in X$;
3. $(\Delta \,;\, \Delta) \in X \implies \Delta \in X$;
4. $\Delta \in X \iff (\Delta)^* \in X$.

Proof For the first point, observe that $\mathsf{False} \vdash_{\mathsf{cf}} \varphi$, so $\mathsf{False} \in \langle\varphi\rangle$ for any formula φ.

For the second point, let X be $\bigcap_{i \in I} \langle \varphi_i \rangle$. Then, $\Delta \in X \iff \forall i \in I.\, \Delta \vdash_{\mathsf{cf}} \varphi_i$. If $\Delta \in X$, then, using weakening:

$$\frac{\Delta \vdash_{\mathsf{cf}} \varphi_i}{\Delta \,;\, \Delta' \vdash_{\mathsf{cf}} \varphi_i}$$

for any $i \in I$. Hence, $\Delta;\ \Delta' \in X$.

Similarly for the other two cases, using contraction, and the left rules, and Corollary 5. □

As an example of a calculation in $\mathcal{C}$, we show the following characterization of meets.

Proposition 20 (`C_and_eq`)**.** *The following holds in $\mathcal{C}$:*

$$X \wedge Y = \mathsf{cl}(\{\Delta \,;\, \Delta' \mid \Delta \in X, \Delta' \in Y\})$$

Proof For the inclusion from left to right: suppose that $\Delta \in X \cap Y$. Then,

$$\begin{aligned}(\Delta \,;\, \Delta) &\in \{\Delta \,;\, \Delta' \mid \Delta \in X, \Delta' \in Y\} \\ &\subseteq \mathsf{cl}(\{\Delta \,;\, \Delta' \mid \Delta \in X, \Delta' \in Y\}).\end{aligned}$$

From Proposition 19 we get

$$\Delta \in \mathsf{cl}(\{\Delta \,;\, \Delta' \mid \Delta \in X, \Delta' \in Y\}).$$

For the inclusion from right to left: it suffices to show:

$$\{\Delta \,;\, \Delta' \mid \Delta \in X, \Delta' \in Y\} \subseteq X \cap Y.$$

If $\Delta \in X$ and $\Delta' \in Y$, then $\Delta \,;\, \Delta' \in X \cap Y$ by Proposition 19. □

6.2 BI Structure

In order to apply Theorem 16 and obtain a BI algebra structure on $\mathcal{C}$, we have to ensure that the Heyting implication of closed sets is closed, and that $X \multimap\!\bullet\, Y \in \mathcal{C}$ whenever $Y \in \mathcal{C}$.

For the following lemma we will use the fact that the $\mathrel{-\!\!*}$R is invertible and Corollary 5.

Lemma 21 (`wand_is_closed`)**.** *If Y is closed, then so is $X \multimap\!\bullet\, Y$; furthermore, it can be described as:*

$$X \multimap\!\bullet\, Y = \{\Delta \mid \forall \Delta' \in X.\, (\Delta \,,\, \Delta') \in Y\}.$$

Proof It is straightforward to check that $X \multimap\!\bullet\, Y$ defined as above is indeed a right adjoint to the $\bullet$ operation. Thus, it remains to show that $X \multimap\!\bullet\, Y$ is closed.

Since Y is closed, it can be written as an intersection of some family of principal closed sets: $Y = \bigcap_{j \in J} \langle \varphi_j \rangle$. Then, we claim,

$$X \multimap\!\bullet\, Y = \bigcap_{(\Delta', j) \in X \times J} \langle (\Delta')^* \mathrel{-\!\!*} \varphi_j \rangle.$$

Direction from left to right: let $\Delta \in X \multimapdotbothB Y$, and let $(\Delta', j) \in X \times J$. We are to show: $\Delta \vdash_{\mathsf{cf}} (\Delta')^* \mathrel{-\!\!*} \varphi_j$. We argue as follows:

$$\dfrac{\dfrac{\Delta \,,\, \Delta' \vdash_{\mathsf{cf}} \varphi_j}{\Delta \,,\, (\Delta')^* \vdash_{\mathsf{cf}} \varphi_j}}{\Delta \vdash_{\mathsf{cf}} (\Delta')^* \mathrel{-\!\!*} \varphi_j}$$

Direction from right to left: suppose that

$$\Delta \in \bigcap_{(\Delta', j) \in X \times J} \langle (\Delta')^* \mathrel{-\!\!*} \varphi_j \rangle,$$

and let $\Delta' \in X$. We are to show $\Delta \,,\, \Delta' \vdash_{\mathsf{cf}} \varphi_j$ for any $j \in J$. By the assumption we have

$$\Delta \vdash_{\mathsf{cf}} (\Delta')^* \mathrel{-\!\!*} \varphi_j.$$

We then reason similarly as in the previous direction, but using inversions Lemma 2 and corollary 5:

$$\dfrac{\dfrac{\Delta \vdash_{\mathsf{cf}} (\Delta')^* \mathrel{-\!\!*} \varphi_j}{\Delta \,,\, (\Delta')^* \vdash_{\mathsf{cf}} \varphi_j}}{\Delta \,,\, \Delta' \vdash_{\mathsf{cf}} \varphi_j}$$

□

We can give a similar characterization of the Heyting implication in $\mathcal{C}$:

Proposition 22 (`has_heyting_impl`)**.** *For every closed X, Y, the Heyting implication is closed and can be described as:*

$$X \to Y = \{\Delta \mid \forall \Delta' \in X, (\Delta \,;\, \Delta') \in Y\}.$$

Proof Using Proposition 20, it is straightforward to check that $X \to Y$ as defined above is a right adjoint to the meet operation $\cap$. The proof of closedness follows the proof of Lemma 21. □

To sum up, by Theorem 16 we have a BI algebra $\mathcal{C}$ in which operations are defined as follows:

$$\mathsf{Emp} = \mathsf{cl}(\{\varnothing_m\}) \qquad \top = \mathit{Bunch}$$

$$\bot = \mathsf{cl}(\emptyset) \qquad X \vee Y = \mathsf{cl}(X \cup Y) \qquad X * Y = \mathsf{cl}(\{\Delta \,,\, \Delta' \mid \Delta \in X, \Delta' \in Y\})$$

$$X \wedge Y = \mathsf{cl}(\{\Delta \,;\, \Delta' \mid \Delta \in X, \Delta' \in Y\})$$

$$X \mathrel{-\!\!*} Y = \{\Delta \mid \forall \Delta' \in X. (\Delta \,,\, \Delta') \in Y\}$$

$$X \to Y = \{\Delta \mid \forall \Delta' \in X. (\Delta \,;\, \Delta') \in Y\}$$

6.3 Fundamental Property of $\mathcal{C}$

We can interpret formulas in the model $\mathcal{C}$ by picking the interpretation of atomic propositions to be $[\![a]\!] = \langle a \rangle$. Now we are ready to prove the main theorem: if $[\![\varphi]\!] \subseteq [\![\psi]\!]$, then $\varphi \vdash_{\mathsf{cf}} \psi$. To obtain this, we prove the following property, due to Okada [16].

Lemma 23 (`okada_property`)**.** *For any formula φ,*

$$\varphi \in [\![\varphi]\!] \subseteq \langle \varphi \rangle$$

(where the leftmost instance of φ is a bunch consisting of a single leaf with the formula φ).

Proof By induction on φ.

Case $\varphi = \mathsf{False}$. We have $[\![\mathsf{False}]\!] = \mathsf{cl}(\emptyset)$. Clearly, $\mathsf{cl}(\emptyset) \subseteq \langle \varphi \rangle$, because $\langle \varphi \rangle$ is closed and $\emptyset \subseteq \langle \varphi \rangle$. By Proposition 19 we have $\mathsf{False} \in [\![\mathsf{False}]\!]$.

Case $\varphi = \mathsf{True}$. We have $[\![\mathsf{True}]\!] = \mathit{Bunch} = \langle \mathsf{True} \rangle$.

Case $\varphi = \mathsf{Emp}$. In order to show $[\![\mathsf{Emp}]\!] = \mathsf{cl}(\{\varnothing_m\}) \subseteq \langle \mathsf{Emp} \rangle$, it suffices to show $\{\varnothing_m\} \subseteq \langle \mathsf{Emp} \rangle$, by the characterization of the closure operator. That inclusion holds because $\varnothing_m \vdash_{\mathsf{cf}} \mathsf{Emp}$. In order to show $\mathsf{Emp} \in \mathsf{cl}(\{\varnothing_m\})$, it suffices to show $\varnothing_m \in \mathsf{cl}(\{\varnothing_m\})$ by Proposition 19, which holds trivially.

Case $\varphi = \psi_1 * \psi_2$. In order to show the set inclusion $[\![\psi_1 * \psi_2]\!] = \mathsf{cl}([\![\psi_1]\!] \bullet [\![\psi_2]\!]) \subseteq \langle \psi_1 * \psi_2 \rangle$, it suffices to show $[\![\psi_1]\!] \bullet [\![\psi_2]\!] \subseteq \langle \psi_1 * \psi_2 \rangle$, by the characterization of the closure operator. If $(\Delta_1, \Delta_2) \in [\![\psi_1]\!] \bullet [\![\psi_2]\!]$, then, by the induction hypothesis $\Delta_i \vdash_{\mathsf{cf}} \psi_i$, and we can reason as follows:

$$\frac{\Delta_1 \vdash_{\mathsf{cf}} \psi_1 \qquad \Delta_2 \vdash_{\mathsf{cf}} \psi_2}{\Delta_1, \Delta_2 \vdash_{\mathsf{cf}} \psi_1 * \psi_2}$$

Hence, $(\Delta_1, \Delta_2) \in \langle \psi_1 * \psi_2 \rangle$.

As for the element inclusion $\psi_1 * \psi_2 \in \mathsf{cl}([\![\psi_1]\!] \bullet [\![\psi_2]\!])$, note that by Proposition 19 it suffices to show $(\psi_1, \psi_2) \in \mathsf{cl}([\![\psi_1]\!] \bullet [\![\psi_2]\!])$, which is evident from the induction hypotheses.

Case $\varphi = \psi_1 \wedge \psi_2$. In order to show the set inclusion, suppose that $\Delta \in [\![\psi_1 \wedge \psi_2]\!] = [\![\psi_1]\!] \cap [\![\psi_2]\!]$. Then, by the induction hypothesis, $\Delta \in \langle \psi_1 \rangle \cap \langle \psi_2 \rangle$, and we can reason as follows:

$$\cfrac{\cfrac{\Delta \vdash_{\mathsf{cf}} \psi_1 \qquad \Delta \vdash_{\mathsf{cf}} \psi_2}{\Delta \,;\, \Delta \vdash_{\mathsf{cf}} \psi_1 \wedge \psi_2}}{\Delta \vdash_{\mathsf{cf}} \psi_1 \wedge \psi_2}$$

As for the element inclusion $\psi_1 \wedge \psi_2 \in [\![\psi_1]\!] \cap [\![\psi_2]\!]$, we argue as follows. By the induction hypothesis, $\psi_1 \in [\![\psi_1]\!]$. By Proposition 19 (item 1), $(\psi_1 ;\ \psi_2) \in [\![\psi_1]\!]$, and by Proposition 19 (item 3), $\psi_1 \wedge \psi_2 \in [\![\psi_1]\!]$. Similarly we can show $\psi_1 \wedge \psi_2 \in [\![\psi_2]\!]$.

Case $\varphi = \psi_1 \mathrel{-\!\!*} \psi_2$. In order to show $[\![\psi_1 \mathrel{-\!\!*} \psi_2]\!] = [\![\psi_1]\!] \mathrel{-\!\!*} [\![\psi_2]\!] \subseteq \langle \psi_1 \mathrel{-\!\!*} \psi_2 \rangle$, suppose that $\Delta \in [\![\psi_1]\!] \mathrel{-\!\!*} [\![\psi_2]\!]$. We are to show $\Delta \vdash_{\mathsf{cf}} \psi_1 \mathrel{-\!\!*} \psi_2$. By the induction hypothesis, $\psi_1 \in [\![\psi_1]\!]$; hence

$$(\Delta \,,\, \psi_1) \in [\![\psi_2]\!] \subseteq \langle \psi_2 \rangle.$$

We can then reason using the right rule for $-\!\!*$:

$$\frac{\Delta\,\mathbf{,}\,\psi_1 \vdash_{\mathsf{cf}} \psi_2}{\Delta \vdash_{\mathsf{cf}} \psi_1 -\!\!* \psi_2}$$

In order to show $\psi_1 -\!\!* \psi_2 \in [\![\psi_1]\!] -\!\!* [\![\psi_2]\!]$, suppose that $\Delta \in [\![\psi_1]\!]$. We are to show $(\Delta\,\mathbf{,}\,\psi_1 -\!\!* \psi_2) \in [\![\psi_2]\!]$. Let us write $[\![\psi_2]\!]$ as $\bigcap_{i\in I}\langle\varphi_i\rangle$. Then our goal can be reduced to showing

$$\Delta\,\mathbf{,}\,\psi_1 -\!\!* \psi_2 \vdash_{\mathsf{cf}} \varphi_i$$

for any $i \in I$. We argue as follows, using the left rule for $-\!\!*$:

$$\frac{\Delta \vdash_{\mathsf{cf}} \psi_1 \qquad \psi_2 \vdash_{\mathsf{cf}} \varphi_i}{\Delta\,\mathbf{,}\,\psi_1 -\!\!* \psi_2 \vdash_{\mathsf{cf}} \varphi_i}$$

where the first assumption holds because $\Delta \in [\![\psi_1]\!] \subseteq \langle\psi_1\rangle$ and the second assumption holds because $\psi_2 \in \langle\psi_2\rangle$.

Case $\varphi = \psi_1 \to \psi_2$. Similarly to the case $\varphi = \psi_1 -\!\!* \psi_2$, using the characterization of the Heyting implication in $\mathcal{C}$.

Case $\varphi = \psi_1 \vee \psi_2$. In order to show $[\![\psi_1 \vee \psi_2]\!] = [\![\psi_1]\!] \vee [\![\psi_2]\!] \subseteq \langle\psi_1 \vee \psi_2\rangle$, it suffices to show $[\![\psi_1]\!] \subseteq \langle\psi_1 \vee \psi_2\rangle$ and $[\![\psi_2]\!] \subseteq \langle\psi_1 \vee \psi_2\rangle$. To show that $[\![\psi_i]\!] \subseteq \langle\psi_1 \vee \psi_2\rangle$, for $i = 1, 2$, it suffices to show $\langle\psi_i\rangle \subseteq \langle\psi_1 \vee \psi_2\rangle$. We show that using the right rules for disjunction.

To show $\psi_1 \vee \psi_2 \in [\![\psi_1 \vee \psi_2]\!] = \mathsf{cl}([\![\psi_1]\!] \cup [\![\psi_2]\!])$, we appeal to the definition of $\mathsf{cl}(-)$: Let φ be a formula such that $[\![\psi_1]\!] \cup [\![\psi_2]\!] \subseteq \langle\varphi\rangle$. We are to show $\psi_1 \vee \psi_2 \in \langle\varphi\rangle$, i.e. $\psi_1 \vee \psi_2 \vdash_{\mathsf{cf}} \varphi$. By assumption we have $\psi_i \in [\![\psi_i]\!]$, for $i = 1, 2$, and, hence $\psi_i \in \langle\varphi\rangle$, or, equivalently, $\psi_i \vdash_{\mathsf{cf}} \varphi$. We obtain the desired result using $\vee$L. □

Theorem 24 (`C_interp_cf`). *If $[\![(\Delta)^*]\!] \leq [\![\varphi]\!]$ holds in $\mathcal{C}$, then $\Delta \vdash_{\mathsf{cf}} \varphi$.*

Proof By Lemma 23, we have $(\Delta)^* \in [\![(\Delta)^*]\!]$, and hence $(\Delta)^* \in [\![\varphi]\!]$. By Proposition 19 we have furthermore have $\Delta \in [\![\varphi]\!]$ which is equivalent to $\Delta \vdash_{\mathsf{cf}} \varphi$. □

As a consequence, we get the cut admissibility:

Theorem 25 (`cut`). *The* CUT *rule is admissible in the cut-free fragment $\vdash_{\mathsf{cf}}$ of BI.*

Proof Suppose $\Delta \vdash_{\mathsf{cf}} \psi$ and $\Gamma(\psi) \vdash_{\mathsf{cf}} \varphi$. We are to show that $\Gamma(\Delta) \vdash_{\mathsf{cf}} \varphi$. By Theorem 24 it suffices to show that $[\![(\Gamma(\Delta))^*]\!] \leq [\![\varphi]\!]$ holds in $\mathcal{C}$.

From soundness we have that $[\![(\Delta)^*]\!] \leq [\![\psi]\!]$. By induction on Γ we can show that $[\![(\Gamma(\Delta))^*]\!] \leq [\![(\Gamma(\psi))^*]\!]$, from which we obtain

$$[\![(\Gamma(\Delta))^*]\!] \leq [\![(\Gamma(\psi))^*]\!] \leq [\![\varphi]\!].$$

□

Overview.

In the next sections we will be looking at adjusting the construction of $\mathcal{C}$ to extensions of BI. At this point we would like to give an overview of the argument, and see what kind of conditions we need.

- To show that the closure operator $\mathsf{cl}(-)$ is strong, we had to use invertibility of certain rules. Firstly, we used the fact that BI satisfies a strong form of the deduction theorem for both implications (the rules $\to$R and $-\!\!*$R are invertible). Secondly, we used the fact that the left rules are invertible for connectives that form bunches (EmpL, TrueL, $\wedge$L, $*$L).
- Additionally, we need to verify that all the rules of sequent calculus are validated in $\mathcal{C}$.
- Finally, we need to show that Okada's property (Lemma 23) holds in $\mathcal{C}$.

This list gives us a sort of roadmap for extending the cut elimination argument. For every rule that we want to add to BI, we need to re-verify the invertibility of certain rules, and that the rule is validated in $\mathcal{C}$. If we want to add a new connective to the system, we need to additionally come up with the interpretation of this connective on $\mathcal{C}$, and re-verify Okada's property.

7 Extending the Logic: Analytic Structural Rules

An important extension of BI is *affine BI*, which extends the sequent calculus of Figure 1 with the weakening rule for $\mathbf{,}$:

$$\text{W}\mathbf{,}\quad \frac{\Delta(\Delta_1) \vdash \varphi}{\Delta(\Delta_1 \mathbf{,} \Delta_2) \vdash \varphi}$$

An algebraic structure for interpreting affine BI is a BI algebra in which the following inequality holds: $p * q \leq p$. Can we extend the argument presented so far to cover affine BI? As we discussed at the end of the previous section, because we are adding a new rule, we have to make sure that the analogues of Lemma 2 and Lemma 4 still hold (the appropriate rules are invertible), and that $\mathcal{C}$ validates the inequality $X * Y \subseteq X$.

To verify that $X * Y \subseteq X$ it suffices to verify that $X \bullet Y \subseteq X$, since X is closed. Let us write $X = \bigcap_{i \in I} \langle \varphi_i \rangle$. Suppose that $\Delta_1 \in X, \Delta_2 \in Y$. We are to show that $\Delta_1 \mathbf{,} \Delta_2 \vdash_{\mathsf{cf}} \varphi_i$ for any i; however we know that $\Delta_1 \vdash_{\mathsf{cf}} \varphi_i$ by the assumption, and the desired result follows by W$\mathbf{,}$.

This kind of argument for W$\mathbf{,}$ can be generalized to infinitely many structural rules of a particular shape, which we call, following [21], *analytic structural rules*. In the remainder of this section we show how to define such analytic structural rules, and we prove cut elimination for BI extended with any combination of such rules.

7.1 Analytic Structural Rules and Bunched Terms

Structural rules are rules of the shape

$$\frac{\Pi(T_1[\Delta_1,\ldots,\Delta_n]) \vdash \varphi \qquad \ldots \qquad \Pi(T_m[\Delta_1,\ldots,\Delta_n]) \vdash \varphi}{\Pi(T[\Delta_1,\ldots,\Delta_n]) \vdash \varphi}$$

where $T_1, \ldots, T_m, T$ are *bunched terms* – bunches built out of connectives $\mathbin{,}, \mathbin{;}$, and variables $x_1, \ldots, x_n$. The notation $T_i[\vec{\Delta}]$ stands for the bunch obtained from T_i by replacing all the variables x_j with Δ_j. A structural rule is *analytic* if the bunched term T in the conclusion is *linear*, that is, every variable in T occurs at most once.

We identify a structural rule with a tuple $(\{T_1, \ldots, T_m\}, T)$. For example, the rule W, above is represented with a tuple $(\{x_1\}, x_1 \mathbin{,} x_2)$. If L is a set of tuples corresponding to analytic structural rules, we write BI+L for a sequent calculus of BI extended with the structural rules from L.

For the rest of this section, we fix a finite collection L of analytic structural rules and the extended system BI+L. We write $\vdash_{\mathsf{cf}}$ for cut-free provability in BI+L, and we denote by $\mathcal{C}$ the BI algebra constructed in Section 6, but for BI+L-provability.

Firstly, we need to check that the construction of $\mathcal{C}$ works out. We need to verify that the rules $\to$L, $-\!*$L, $\wedge$L, $*$L, TrueL, EmpL are still invertible, in presence of the additional rules from L. For that, we just follow the proof of Lemma 4.

7.2 Interpretation of Structural Rules in $\mathcal{C}$

Additionally, we need to verify that $\mathcal{C}$ validates all the rules from L.

Each bunched term $T[x_1, \ldots, x_n]$ can be interpreted as a function $[\![T]\!] : A^n \to A$ on any BI algebra A. For example, a (non-linear) bunched term $(x_1 \mathbin{,} x_2) \mathbin{;} x_1$ gives rise to a mapping $(X_1, X_2) \mapsto (X_1 * X_2) \wedge X_1$.

In order to interpret a structural rule given by a tuple $(\{T_1, \ldots, T_m\}, T)$ in a BI algebra A, we require that the following inequality holds in A for any $a_1, \ldots, a_n \in A$:

$$[\![T]\!](a_1, \ldots, a_n) \leq [\![T_1]\!](a_1, \ldots, a_n) \vee \cdots \vee [\![T_m]\!](a_1, \ldots, a_n).$$

In this case, we say that A validates the structural rule. For example, recall that the weakening rule W, for , is represented as $(\{x_1\}, (x_1 \mathbin{,} x_2))$. Then the associated inequality is:

$$[\![x_1 \mathbin{,} x_2]\!](p, q) \leq [\![x_1]\!](p, q) \qquad \Longleftrightarrow \qquad p * q \leq p.$$

Lemma 26 (`seq_interp_sound`). *If a BI algebra A validates the rules in L, then $\Delta \vdash \varphi$ implies $[\![\Delta]\!] \leq [\![\varphi]\!]$ in A.*

Proof For the case of a structural rule $(\{T_1, \ldots, T_m\}, T)$, we assume that $[\![T_i]\!](\bar{a}) \leq [\![\varphi]\!]$ holds for any $1 \leq i \leq m$. Then, $\bigvee_{1 \leq i \leq m} [\![T_i]\!](\bar{a}) \leq [\![\varphi]\!]$. Since the rule is validated in A we have

$$[\![T]\!](\bar{a}) \leq \bigvee_{1 \leq i \leq m} [\![T_i]\!](\bar{a}) \leq [\![\varphi]\!].$$

□

In order to show that $\mathcal{C}$ validates all the rules from L, we need the following lemmas about $[\![T]\!]$. For the algebra $\mathcal{C}$ we have the following description:

Lemma 27 (`bterm_C_refl`). *Let $X_1, \ldots, X_n \in \mathcal{C}$, and $\Delta_i \in X_i$ for $1 \leq i \leq n$. Then for any bunched term T,*

$$T[\vec{\Delta}] \in [\![T]\!](\vec{X})$$

Proof By induction on T. □

Lemma 28 (`blinterm_C_desc'`). *For any $X_1, \ldots, X_n \in \mathcal{C}$ and any linear bunched term T we have*

$$[\![T]\!](X_1, \ldots, X_n) = \mathsf{cl}(\{T[\Delta_1, \ldots, \Delta_n] \mid \Delta_i \in X_i, 1 \leq i \leq n\})$$

Proof In view of Lemma 27 it suffices to show that the left-hand side is included in the right-hand side. This is done by induction on T. We show only the case for $\mathbin{\boldsymbol{,}}$, as the other case is similar. If $T(\vec{x}) = F(\vec{x}) \mathbin{\boldsymbol{,}} U(\vec{x})$, then, since T is linear, we can write it down as

$$T(\vec{y}\vec{z}) = F(\vec{y}) \mathbin{\boldsymbol{,}} U(\vec{z})$$

for some factorization $\vec{y}\vec{z} = \vec{x}$, and for some linear terms F and U. By the induction hypothesis we have

$$[\![T]\!](\vec{Y}\vec{Z}) = \mathsf{cl}(\mathsf{cl}(\{F[\vec{\Gamma}] \mid \vec{\Gamma} \in \vec{Y}\}) \bullet \mathsf{cl}(\{U[\vec{\Gamma}] \mid \vec{\Gamma} \in \vec{Z}\})).$$

In order to show the inclusion into $\mathsf{cl}(\{T[\vec{\Delta}] \mid \vec{\Delta} \in \vec{Y}\vec{Z}\})$ it suffices to show

$$\{F[\vec{\Gamma}] \mid \vec{\Gamma} \in \vec{Y}\} \bullet \{U[\vec{\Gamma}] \mid \vec{\Gamma} \in \vec{Z}\} \subseteq \{T[\vec{\Delta}] \mid \vec{\Delta} \in \vec{Y}\vec{Z}\}.$$

Let $\vec{\Gamma} \in \vec{Y}$ and $\vec{\Theta} \in \vec{Z}$. Then, $\vec{\Gamma}\vec{\Theta} \in \vec{X}$, and, hence $F[\vec{\Gamma}] \mathbin{\boldsymbol{,}} U[\vec{\Theta}] = T[\vec{\Gamma}\vec{\Theta}]$, which concludes the proof the inclusion. □

With the two lemmas at hand we can prove that $\mathcal{C}$ is a model of BI+L.

Lemma 29 (`C_extensions`). *Every rule from the set L is validated in $\mathcal{C}$.*

Proof Suppose that $(\{T_1, \ldots, T_m\}, T)$ is a simple structural rule from L. We have to show $[\![T]\!](\vec{X}) \subseteq \mathsf{cl}(\bigcup_{1 \leq i \leq m} [\![T_i]\!](\vec{X}))$. By Lemma 28, it suffices to show

$$\{T[\Delta_1, \ldots, \Delta_n] \mid \vec{\Delta} \in \vec{X}\} \subseteq \mathsf{cl}(\bigcup_{1 \leq i \leq m} [\![T_i]\!](\vec{X}))$$

where $\vec{\Delta} \in \vec{X}$ is a shorthand for $\Delta_i \in X_i$ for all $1 \leq i \leq n$.

Suppose that φ is such that $\bigcup_{1\leq i\leq n}[\![T_i]\!](\vec{X}) \subseteq \langle\varphi\rangle$. We are to show that $T[\vec{\Delta}] \vdash_{\mathsf{cf}} \varphi$, for any $\vec{\Delta} \in \vec{X}$. By Lemma 27, we have $T_i[\vec{\Delta}] \in [\![T_i]\!](\vec{X}) \subseteq \langle\varphi\rangle$. So we get $T_i[\vec{\Delta}] \vdash_{\mathsf{cf}} \varphi$, from which we can conclude that $T[\vec{\Delta}] \vdash_{\mathsf{cf}} \varphi$ □

We can summarize the results of this section in the following theorem.

Theorem 30 (`cut`). *Let L be an arbitrary set of analytic structural rules. Then the* CUT *rule is admissible in the cut-free fragment $\vdash_{\mathsf{cf}}$ of BI+L.*

8 Analytic Completion

In the previous section we have seen that the semantic proof of cut-elimination works well for extensions of BI with analytic structural rules. However, a lot of usual extensions of BI are not given in terms of analytic structural rules. For example, the following "restricted weakening" rule, corresponding to an equation $p \leq p * p$, is not in an analytic form:

$$\text{W-RESTR}\ \frac{\Pi(\Delta) \vdash \varphi}{\Pi(\Delta \,,\, \Delta) \vdash \varphi}$$

because the bunch meta-variable Δ occurs twice in the conclusion. In this section we show that any structural rule, e.g. any rule of the shape

$$\frac{\Pi(T_1[\Delta_1,\ldots,\Delta_n]) \vdash \varphi \qquad \ldots \qquad \Pi(T_m[\Delta_1,\ldots,\Delta_n]) \vdash \varphi}{\Pi(T[\Delta_1,\ldots,\Delta_n]) \vdash \varphi},$$

can be transformed into an equivalent analytic rule through the process known as *analytic completion*. Analytic completion was developed for extensions of Lambek calculus by Ciabattoni et al [21]; we adapt the analytic completion to the setting of BI. We describe this process informally first, before proving its correctness.

8.1 Analytic completion procedure

For the remainder of the section, let us fix the following (non-analytic) structural rule as a running example:

$$\text{RS}\ \frac{\Pi(\Delta_1 \,,\, \Delta_2) \vdash \varphi}{\Pi((\Delta_1 \,,\, \Delta_1) \,;\, \Delta_2) \vdash \varphi}$$

which corresponds to the equation $(p*p)\wedge q \leq p*q$. The corresponding bunched terms are $T[X_1, X_2] = (X_1 \,,\, X_1) \,;\, X_2$ for the conclusion and $T_1[X_1, X_2] = X_1 \,,\, X_2$ for the premise.

The analytic completion proceeds in two steps: first, making the conclusion linear, thus turning a rule into an analytic one; second, adjusting the premises to match the newly linearized conclusion.

Linearizing the conclusion.

Let us describe the linearization step first, i.e. making the term T in the conclusion linear.

Definition 31 (`linearize_bterm`, `linearize_bterm_ren`). *Given a term T, a linearization of T is a term $T^\bullet$ obtained by replacing every occurrence of a variable in T with a fresh variable. The linearization comes with an associated renaming function r_T mapping each fresh variable of $T^\bullet$ to an "old" variable of T.*

Continuing with the example of the structural rule RS, the term in the conclusion is $T[X_1, X_2] = (X_1 \mathbin{,} X_1) \mathbin{;} X_2$. Linearizing it we get $T^\bullet[Y_1, Y_2, Y_3] = (Y_1 \mathbin{,} Y_2) \mathbin{;} Y_3$. The associated renaming function is then given by $r_T(Y_1) = r_T(Y_2) = X_1$ and $r_T(Y_3) = X_2$. Since the choices of variable names do not actually matter, we can assume that all the variables come from the same pool $X_1, X_2, \ldots$, and the renaming function operates on indices. In the previous example, we could write $r_T(1) = r_T(2) = 1$ and $r_T(3) = 2$.

The relation between the linearized term and the associated renaming function is established in the following lemma:

Lemma 32 (`linearize_bterm_act_ren`). *Let T be a bunched term with n variables, and let $T^\bullet$ be its linearization with $k > n$ variables.*

$$T[X_1, \ldots, X_n] = T^\bullet[X_{r_T(1)}, \ldots, X_{r_T(k)}].$$

In our running example RS, we have $T[X_1, X_2] = T^\bullet[X_1, X_1, X_2]$.

A formal definition of the linearization function depends on the exact representation of terms. For these reasons we defer it to Section 10.4, where we describe the Coq representation of terms and the implementation of the analytic completion.

Adjusting the premises.

Now that we have linearized the bunched term in the conclusion, we need to adjust the premises in such a way that we obtain an equivalent structural rule. The question is, how do we update the premises in such a way that the resulting structural rule is equivalent to the one we started with? One way to being answering this question is to see in what ways we can apply the original structural rule to the linearized conclusion?

In the running example RS, the linearized conclusion is $T^\bullet[X_1, X_2, X_3] = (X_1 \mathbin{,} X_2) \mathbin{;} X_3$; or, written as a propositional formula, $(x_1 * x_2) \wedge x_3$. We cannot apply the inequality $(p*p) \wedge q \leq p*q$ directly to that formula, because we cannot unify x_1 and x_2. Note, however, that we can obtain a common proposition $x_1 \vee x_2$ from both of those propositions. This leads us to the following chain of inequalities

$$(x_1 * x_2) \wedge x_3 \leq \big((x_1 \vee x_2) * (x_1 \vee x_2)\big) \wedge x_3 \leq (x_1 \vee x_2) * x_3 = (x_1 * x_3) \vee (x_2 * x_3),$$

where the second inequality corresponds to the application of the original rule RS. Note that in the conclusion we have $T_1[x_1, x_3] \vee T_1[x_2, x_3]$.

As we can see, in place of the variable x_1 in the original bunched term T, we plugged in the disjunction $x_1 \vee x_2$, because both x_1 and x_2 are renamed into x_1 by the renaming function r_T. Then, when applying the inequality, instead of the original premise $T_1[x_1, x_2]$, we get a disjunction $T_1[x_1, x_3] \vee T_1[x_2, x_3]$ where r_T renames x_1 and x_2 into x_1 and x_3 into x_2. To see this more clearly, we can rewrite the obtained proposition as:

$$\bigvee \{T_1[x_i, x_j] \mid r_T(i) = 1, r_T(j) = 2\}.$$

If this is the result of applying the original rule to the linearized conclusion, then we can just take this result to be the new set of premises: $T_1[x_1, x_3]$ and $T_1[x_2, x_3]$. For example, the resulting analytic version of RS is

$$\text{RS'}\quad \frac{\Pi(\Theta_1 \,,\, \Theta_3) \vdash \varphi \qquad \Pi(\Theta_2 \,,\, \Theta_3) \vdash \varphi}{\Pi((\Theta_1 \,,\, \Theta_2) \,;\, \Theta_3) \vdash \varphi}$$

Here we used the variable names Θ, to further contrast RS' with the original rule RS. The original rule RS can be recovered by picking $\Theta_1 = \Theta_2 = \Delta_1$ and $\Theta_3 = \Delta_2$, as described by the linearization function.

Using this example as an insight, we define the procedure $\mathsf{transform}(T_i)$ which transforms a term T_i from the premise into a set of terms that we will use as the premises of the analytic rule. More specifically, we transform each term T_i in the premise into a set of terms as follows:

$$\begin{aligned} &T'[X_1, \ldots, X_k] \in \mathsf{transform}(T_i) \iff \\ &\quad \exists \sigma.\, T'[X_1, \ldots, X_k] = T_i^{\bullet}[X_{\sigma(1)}, \ldots, X_{\sigma(n)}] \wedge \forall j.\, \sigma(j) \in r_T^{-1}(r_{T_i}(j))\end{aligned}$$

We implemented a direct (intentional) and computable definition of this procedure in Coq, which we detail in Section 10.4. For now, we can assume that we can write $\mathsf{transform}(-)$ as a computable function.

Definition 33 (Analytic completion)**.** *Given a structural rule*

$$\frac{\Pi(T_1[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \qquad \ldots \qquad \Pi(T_m[\Delta_1, \ldots, \Delta_n]) \vdash \varphi}{\Pi(T[\Delta_1, \ldots, \Delta_n]) \vdash \varphi},$$

we define its analytic completion to be the analytic structural rule

$$\frac{\{\Pi(T'_j[\Delta_1,\ldots,\Delta_k]) \vdash \varphi \mid 1 \le i \le m, T'_j \in \mathsf{transform}(T_i)\}}{\Pi(T^\bullet[\Delta_1,\ldots,\Delta_k]) \vdash \varphi}$$

8.2 Correctness

In order to show that the obtained rule is equivalent to the one we started with, we will require the following lemmas.

Lemma 34 (linearize_bterm_act)**.** *For any term T we have*

$$T^\bullet[X_1,\ldots,X_k] \le T[\bigvee\{X_j \mid j \in r_T^{-1}(1)\},\ldots,\bigvee\{X_j \mid j \in r_T^{-1}(n)\}]$$

Proof By induction on the structure of the term. □

In our running example, we used Lemma 34 to conclude $(x_1 * x_2) \wedge x_3 \le \big((x_1 \vee x_2) * (x_1 \vee x_2)\big) \wedge x_3$.

With this we can prove soundness.

Proposition 35 (Soundness, analytic_completion_sound)**.** *Suppose that the original structural rule is satisfied in an algebra $\mathcal{A}$. Then the analytic completion of that rule is also satisfied in the same algebra.*

Proof We are to show

$$[\![T^\bullet]\!][X_1,\ldots,X_k] \le \bigvee_{1\le i\le m,\ T'_j \in \mathsf{transform}(T_i)} [\![T'_j]\!][X_1,\ldots,X_k],$$

where $\vec{X} \in \mathcal{A}$. Then, by Lemma 34 we have

$$[\![T^\bullet]\!][X_1,\ldots,X_k] \le [\![T]\!][Y_1,\ldots,Y_n],$$

where $Y_i = \bigvee\{X_j \mid j \in r_T^{-1}(i)\}$.

We know that the original rule holds in $\mathcal{A}$. Then,

$$[\![T]\!][Y_1,\ldots,Y_n] \le [\![T_i]\!][Y_1,\ldots,Y_n],$$

for some premise term T_i.

Then, by induction on the term T_i we can prove

$$[\![T_i]\!][Y_1,\ldots,Y_n] \le \bigvee_{T'_j \in \mathsf{transform}(T_i)} [\![T'_j]\!][X_1,\ldots,X_k].$$

The idea for this proof is to select, for each disjunction $\bigvee\{X_j \mid j \in r_T^{-1}(l)\}$ associated to the l-th variable a representative X_j^l for $j \in r_T^{-1}(l)$ that actually holds in the disjunction. This choice of X_j^l for each $1 \le j \le n$ will determine the renaming σ, which signifies which premise from the set $\mathsf{transform}(T_i)$. □

For completeness we will also need the following lemma

Lemma 36 (transformed_premise_act_ren). *Let $T'_i \in \mathsf{transform}(T_i)$. Then,*

$$T_i[X_1, \ldots, X_n] = T'_i[X_{r_T(1)}, \ldots, X_{r_T(k)}].$$

Proof By definition, if $T'_i \in \mathsf{transform}(T_i)$, then

$$T'_i[X_{r_T(1)}, \ldots, X_{r_T(k)}] = {T_i}^{\bullet}[X_{r_T(\sigma(1))}, \ldots, X_{r_T(\sigma(n))}],$$

for σ satisfying $\sigma(j) \in r_T^{-1}(r_{T_i}(j))$. Then, for any j we have $r_T(\sigma(j)) = r_{T_i}(j)$. Therefore we have

$${T_i}^{\bullet}[X_{r_T(\sigma(1))}, \ldots, X_{r_T(\sigma(n))}] = {T_i}^{\bullet}[X_{r_{T_i}(1)}, \ldots, X_{r_{T_i}(n)}] = T_i[X_1, \ldots, X_k],$$

where the last equality follows from Lemma 32 □

Proposition 37 (Completeness, analytic_completion_complete). *Suppose that the analytic completion of a structural rule is satisfied in an algebra $\mathcal{A}$. Then that rule is also satisfied in the same algebra.*

Proof We are to show

$$[\![T]\!][Y_1, \ldots, Y_n] \leq \bigvee_{1 \leq i \leq m} [\![T_i]\!][Y_1, \ldots, Y_n]$$

where $\vec{Y} \in \mathcal{A}$. Then, by Lemma 32, we have

$$[\![T]\!][Y_1, \ldots, Y_n] = [\![T^{\bullet}]\!][Y_{r_T(1)}, \ldots, Y_{r_T(k)}].$$

From the assumption that the analytic closure of the rule holds, we then have

$$[\![T^{\bullet}]\!][Y_{r_T(1)}, \ldots, Y_{r_T(k)}] \leq [\![T'_j]\!][Y_{r_T(1)}, \ldots, Y_{r_T(k)}]$$

for some $T'_j \in \mathsf{transform}(T_i)$ for some $1 \leq i \leq m$.

Finally, by Lemma 36, we have

$$[\![T]\!][Y_1, \ldots, Y_n] \leq [\![T'_j]\!][Y_{r_T(1)}, \ldots, Y_{r_T(k)}] = T_i[Y_1, \ldots, Y_n].$$

□

Combining the results above with Theorem 30, we get the following theorem, summarizing the results of this section.

Theorem 38. *Let L be an arbitrary set of structural rules. Then the* CUT *rule is admissible in the cut-free fragment $\vdash_{\mathsf{cf}}$ of BI+L.*

9 Extending the Logic: an S4 Modality

In this section we look at a different kind of extension to BI, the one obtained by "freely" adding an (intuitionistic) S4-like modality. This amounts to adding the following rules (usual for intuitionistic formulation of S4 sequent calculus [22]):

$$\Box\text{R}\ \frac{\Box\Delta \vdash A}{\Box\Delta \vdash \Box A} \qquad \Box\text{L}\ \frac{\Delta(A) \vdash B}{\Delta(\Box A) \vdash B}$$

where $\Box\Delta$ is the same as Δ, but with boxes $\Box$ put in front of all the formulas, e.g.

$$\Box(\varnothing_m \mathbin{;} (\varphi \mathbin{,} \psi) \mathbin{;} \chi) \triangleq \varnothing_m \mathbin{;} (\Box\varphi \mathbin{,} \Box\psi) \mathbin{;} \Box\chi.$$

We denote the extended system (the sequent calculus from Figure 1 with the rules $\Box$R, $\Box$L above) as BIS4. We can verify that the relevant rules are still invertible (a version of Lemma 4 and Lemma 2 for BIS4).

Interpreting the modality.

As per the roadmap at the end of Section 6 we need to interpret the modality $\Box$ on $\mathcal{C}$ somehow. The usual way of interpreting a $\Box$ modality in intuitionistic setting is with an interior operator (c.f. the notion of a CS4 algebra [23, Definition 3]).

Definition 39. *A* BIS4 algebra *is a tuple* $(\mathcal{B}, \Box)$ *where* $\mathcal{B}$ *is a BI algebra and* $\Box : \mathcal{B} \to \mathcal{B}$ *is a monotone function satisfying:*

1. $\Box p \leq p$;
2. $\Box p \leq \Box\Box p$;
3. $\top = \Box\top$
4. $\mathsf{Emp} = \Box\mathsf{Emp}$;
5. $\Box p \wedge \Box q \leq \Box(p \wedge q)$;
6. $\Box p * \Box q \leq \Box(p * q)$.

We define the interior operator $\Box$ on $\mathcal{C}$ as:

$$\Box X \triangleq \mathsf{cl}(\{\Box\Delta \mid \Delta \in X\}).$$

In order to show that $\mathcal{C}$ satisfies the conditions from Definition 39, we will use the following lemmas.

Lemma 40 (`box_l_inv`)**.** *The following rule is admissible:*

$$\textsc{$\Box$-idemp}\ \frac{\Gamma(\Box\Box\Delta) \vdash \varphi}{\Gamma(\Box\Delta) \vdash \varphi}$$

Proof By induction on the height of the derivation, similar to the proof of Lemma 4. □

Lemma 41 (`C_necessitate`, `C_bunch_box_idemp`)**.** *Let X be a closed set.*

- *If $\Delta \in X$, then $\Box\Delta \in X$.*
- *If $\Box\Box\Delta \in X$, then $\Box\Delta \in X$.*

Proof By examining the definitions of $\Box$ and $\mathsf{cl}(-)$, using Lemma 40 for the second item. □

Lemma 42 (C_alg_box)**.** *$(\mathcal{C}, \Box)$ is a BIS4 algebra.*

Proof The conditions (1) and (2) follow from Lemma 41. The conditions (3) and (4) can be shown by examining the definitions of all the connectives involved.

The condition (6) can be shown as follows. To show $\Box X * \Box Y \subseteq \Box(X * Y)$, we reason as follows:

$$\begin{aligned}&\Box X * \Box Y = \mathsf{cl}(\mathsf{cl}(\{\Box\Delta \mid \Delta \in X\}) \bullet \mathsf{cl}(\{\Box\Delta \mid \Delta \in Y\})) \subseteq \\ &\mathsf{cl}(\mathsf{cl}(\{\Box\Delta \mid \Delta \in X\} \bullet \mathsf{cl}(\{\Box\Delta \mid \Delta \in Y\}))) = \\ &\mathsf{cl}(\{\Box\Delta \mid \Delta \in X\} \bullet \mathsf{cl}(\{\Box\Delta \mid \Delta \in Y\})).\end{aligned}$$

To show that

$$\mathsf{cl}(\{\Box\Delta \mid \Delta \in X\} \bullet \mathsf{cl}(\{\Box\Delta \mid \Delta \in Y\})) \subseteq \Box(X * Y)$$

it suffices to show that

$$\{\Box\Delta \mid \Delta \in X\} \bullet \mathsf{cl}(\{\Box\Delta \mid \Delta \in Y\}) \subseteq \Box(X * Y).$$

And, since

$$\{\Box\Delta \mid \Delta \in X\} \bullet \mathsf{cl}(\{\Box\Delta \mid \Delta \in Y\}) \\ \subseteq \mathsf{cl}(\{\Box\Delta \mid \Delta \in X\} \bullet \{\Box\Delta \mid \Delta \in Y\}),$$

it suffices to show

$$\{\Box\Delta \mid \Delta \in X\} \bullet \{\Box\Delta \mid \Delta \in Y\} \subseteq \Box(X * Y).$$

Let Δ be such that $\Delta = \Box\Delta_1 \, , \Box\Delta_2$, for $\Delta_1 \in X$, $\Delta_2 \in Y$. Then $\Delta = \Box(\Delta_1 \, , \Delta_2)$, with $\Delta_1 \, , \Delta_2 \in X * Y$.

Finally, the condition (5) is shown similarly. □

All it remains to verify is that Okada's property (Lemma 23) still holds. Since we have added only the $\Box$ modality we need to check one extra case:

Lemma 43. *Assume that φ is such that $\varphi \in [\![\varphi]\!] \subseteq \langle\varphi\rangle$. Then*

$$\Box\varphi \in [\![\Box\varphi]\!] \subseteq \langle\Box\varphi\rangle.$$

Proof In order to show the first inclusion, note that by the hypothesis, we have $\varphi \in [\![\varphi]\!]$. Hence,

$$\Box\varphi \in \{\Box\Delta \mid \Delta \in [\![\varphi]\!]\} \subseteq \Box[\![\varphi]\!].$$

To show the second inclusion it suffices to show

$$\{\Box\Delta \mid \Delta \in [\![\varphi]\!]\} \subseteq \langle\Box\varphi\rangle.$$

So, let us assume $\Delta \in [\![\varphi]\!]$. By the induction hypothesis we have $\Delta \vdash_{\mathsf{cf}} \varphi$, and, hence $\Box\Delta \vdash_{\mathsf{cf}} \Box\varphi$. Which gives us the desired result $\Box\Delta \in \langle\Box\varphi\rangle$. □

Theorem 44 (cutelim_s4.cut)**.** *The* CUT *rule is admissible in the cut-free fragment $\vdash_{\mathsf{cf}}$ of BIS4.*

10 The Coq Formalization

As we mentioned, the results of this paper has been formalized in the Coq proof assistant. In this section we describe some of the design choices and trade-offs that we made.

10.1 Sequent calculus and bunch decomposition

Instead of formalizing sequent calculus with the cut rule and deriving a cut-free sequence calculus from that, we opted for formalizing just the cut-free sequent calculus and proving that cut it admissible in that system. The sequent calculus is then given as a standard inductive family of propositions in Coq.

The trickiest proofs to formalize about the sequent calculus were the admissibility of the inverted rules (Lemma 4) in the cut-free sequent calculus. Firstly, as was mentioned in Section 3, those admissibility proofs proceed by induction on the height of the derivation. To handle this in the Coq formalization, we use an auxiliary relation `proves : bunch → formula → nat → Prop` which includes the (upper bound on the) height of the derivation. Our reasoning behind this definition is that if we were to define a proof height function and do induction on its value, we would have to formulate our goal (and the proof) in a rather unwieldy way: we would have to package together the context, the formula, and the derivation into a Σ-type: `Σ (Δ : bunch) (φ : formula), proves Δ φ`.

Secondly, even with induction on proof height, in the proof of Lemma 4 we often end with a situation where we have a bunch Δ that can be decomposed multiple ways that we need to related to each other. For example, in the proof of invertibility of $*$L, we want to obtain a proof of $\Delta_0(\varphi \mathbin{,} \psi) \vdash \chi$ from a proof of $\Delta_0(\varphi * \psi) \vdash \chi$. Suppose that the last applied rule in the proof was weakening

$$\frac{\Delta_1(\Gamma_1) \vdash_{\mathsf{cf}} \chi}{\Delta_1(\Gamma_1 \mathbin{;} \Gamma_2) \vdash_{\mathsf{cf}} \chi}$$

with $\Delta_1(\Gamma_1 \mathbin{;} \Gamma_2) = \Delta_0(\varphi * \psi)$. In order to apply the induction hypothesis we have to locate the formula $\varphi * \psi$ somewhere in the bunch $\Delta_1(\Gamma_1)$. The formula may appear either in Γ_1, Γ_2, or be part of the bunched context $\Delta_1(\cdot)$, depending on the relation between Δ_0 and Δ_1. This is an example of informal observation that comes up often in the BI sequent calculus because all the left rules (and structural rules) can be applied deep inside an arbitrary bunch. As such, reasoning about what appears where in bunched contexts is of importance.

In order to reason about situations like this in Coq, we define an auxiliary inductive system $\Delta \rightsquigarrow \langle \Pi(-) \mid \Delta' \rangle$ that captures exactly when $\Delta = \Pi(\Delta')$. The rules for the decomposition of bunches is given in Figure 2.

Lemma 45 (`bunch_decomp_iff`). $\Delta = \Pi(\Delta')$ *if and only if* $\Delta \rightsquigarrow \langle \Pi \mid \Delta' \rangle$.

Using this inductive system we can prove the following lemmas about decomposition of contexts, that we use for formalizing proofs from Section 3:

$$\Delta \rightsquigarrow \langle (-) \mid \Delta \rangle \qquad \frac{\Delta_1 \rightsquigarrow \langle \Pi(-) \mid \Delta \rangle}{\Delta_1 \,,\, \Delta_2 \rightsquigarrow \langle \Pi(-) \,,\, \Delta_2 \mid \Delta \rangle}$$

$$\frac{\Delta_2 \rightsquigarrow \langle \Pi(-) \mid \Delta \rangle}{\Delta_1 \,,\, \Delta_2 \rightsquigarrow \langle \Delta_1 \,,\, \Pi(-) \mid \Delta \rangle} \qquad \frac{\Delta_1 \rightsquigarrow \langle \Pi(-) \mid \Delta \rangle}{\Delta_1 \,;\, \Delta_2 \rightsquigarrow \langle \Pi(-) \,;\, \Delta_2 \mid \Delta \rangle}$$

$$\frac{\Delta_2 \rightsquigarrow \langle \Pi(-) \mid \Delta \rangle}{\Delta_1 \,;\, \Delta_2 \rightsquigarrow \langle \Delta_1 \,;\, \Pi(-) \mid \Delta \rangle}$$

Fig. 2 Inductive rules for decomposition of bunches.

Lemma 46 (`fill_is_frml`). *If $\Pi(\Delta) = \varphi$ then Π is an empty context and $\Delta = \varphi$.*

Lemma 47 (`bunch_decomp_ctx`). *If $\Pi(\Delta) \rightsquigarrow \langle \Pi'(-) \mid \varphi \rangle$ then one of the two conditions hold:*

- *The formula φ appears in Δ itself. That is, there is $\Pi_0(-)$ such that $\Delta \rightsquigarrow \langle \Pi_0(-) \mid \varphi \rangle$ and $\Pi'(-) = \Pi(\Pi_0(-))$.*
- *Or the formula φ appears in the context $\Pi(-)$. Then we can think of $\Pi'(-)$ as a context with two holes, one of which is already filled with Δ. Formally we represent this situation as follows. There are functions Π_0, Π_1 from bunches to bunched contexts, such that:*
  - *For any bunch Λ, we have $\Pi(\Lambda) \rightsquigarrow \langle \Pi_0(\Lambda)(-) \mid \varphi \rangle$.*
  - *For any bunch Λ, we have $\Pi'(\Lambda) \rightsquigarrow \langle \Pi_1(\Lambda)(-) \mid \Delta \rangle$.*
  - *For any bunches Λ, Λ', we have $\Pi_0(\Lambda)(\Lambda') = \Pi_1(\Lambda')(\Lambda)$.*

In order to prove the invertibility of relevant rules for BIS4 (Section 9), including Lemma 40 we related bunch decomposition and the $\Box$ modality:

Lemma 48 (`bunch_decomp_box`). *If $\Box\Delta = \Pi(\Box\varphi)$, then there is a bunched context Π' such that*

- *$\Delta = \Pi'(\varphi)$;*
- *for any Γ, $\Box\Pi'(\Gamma) = \Pi(\Box\Gamma)$.*

10.2 Algebraic semantics

As for the algebraic semantics, we used a slightly modified formalization of BI algebras from the Iris Coq library [24, 25]. The original formulation BI algebras in Iris also includes a *persistence modality* [26], which behaves quite differently from an S4-like modality that we use in Section 9. To our knowledge, the proof theory of this modality has not been studied and there is no sequent calculus for this logic. This formalization contains a lot of useful lemmas and makes heavy use of setoids, which allows us to easily formulate the model $\mathcal{P}(\mathit{Bunch})$ of predicates on bunches quotiented by bunch equivalence.

```
Inductive bterm (V : Type) : Type :=
| Var (x : V)
| TComma (T1 T2 : bterm V)
| TSemic (T1 T2 : bterm V).
Fixpoint term_fv (T : bterm V) : gset V :=
  match T with
  | Var x ⇒ {[ x ]}
  | TComma T1 T2 ⇒ term_fv T1 ∪ term_fv T2
  | TSemic T1 T2 ⇒ term_fv T1 ∪ term_fv T2
  end.
Fixpoint linear_bterm
  (T : bterm V) : Prop :=
  match T with
  | Var x ⇒ True
  | TComma T1 T2
  | TSemic T1 T2 ⇒
    term_fv T1 ## term_fv T2 ∧
    linear_bterm T1 ∧ linear_bterm T2
  end.
Definition bterm_alg_act {PROP : bi} (T : bterm V) (Xs : V →
PROP) : PROP.
Instance bterm_fmap : FMap bterm.
(* Class FMap (F : Type → Type) := A → B → F A → F B *)
Definition bterm_gset_fold : bterm (gset V) → bterm V.
```

Fig. 3 Coq definitions of bunched terms.

We also use setoids extensively in other parts of the formalization. We found it useful to declare the bunch equivalence $\Delta_1 \equiv \Delta_2$ as a rewritable relation with relevant `Proper` typeclass instances. In addition to that, we make use of various typeclasses from the std++ library [27], like `LeftId` and `Comm`, for registering the unital and commutativity laws of bunch equivalence with the Coq's `rewrite` system. For example, this allows us to easily turn a statement $(\Delta_1 \mathbin{,} \varnothing_m) \mathbin{;} \Delta_2 \in X$ to $\Delta_2 \mathbin{,} \Delta_2 \in X$, for a closed set $X \in \mathcal{C}$.

10.3 Bunched terms and structural rules

The sequent calculus (and, consequently, the algebra $\mathcal{C}$) is parameterized by a collection of analytic structural rules (as in Section 7). Recall that an analytic structural rule is a tuple $(\vec{T}, T)$ of bunched terms, such that T is a linear term representing the context in the conclusion, and the rules $\vec{T}$ represent the contexts in the premises.

Bunched terms.

Let us first look at the API for bunched terms that we have developed. Bunched terms `bterm V` are defined in Coq inductively and are parameterized by the type `V` of variables. The definition of `bterm V` and its corresponding API is given in Figure 3. The `bterm_alg_act` function interprets a

```
Definition structural_rule := (list (bterm nat) * bterm nat)%type.
Definition is_analytical (s : structural_rule) := linear_bterm (snd s).
Definition rule_valid (s : structural_rule) (PROP : bi) : Prop :=
  ∀ (Xs : nat → PROP),
    bterm_alg_act (snd s) Xs ⊢
      ∃ Ti' : {Ti : bterm nat | Ti ∈ fst s}, bterm_alg_act (proj1_sig Ti')
```

Fig. 4 Coq definitions of bunched terms.

bunched term `T` in a BI algebra `PROP`, given an interpretation `Xs` for all the variables. The `bterm_fmap` typeclass instance allows us to write terms like `fmap (fun n ⇒ n + 1) (TComma (Var 0) (Var 1))`, which computes to `TComma (Var 1) (Var 2)`. The function `term_fv` returns a set[2] of variables used in a term. The predicate `linear_bterm` determines whether a term is linear (i.e. every variable in the term occurs exactly once). In the definition we use the predicate `X ## Y` stating that the sets `X` and `Y` are disjoint.

Finally, the function `bterm_gset_fold` converts a term with sets of variables to a set of terms with variables, by exhausting different picks of variables from each set. For example, it would turn a term $(\{X_1, X_2\} \mathbin{,} \{X_1\}) \mathbin{;} \{X_2\}$ into a set $\{((X_1 \mathbin{,} X_1) \mathbin{;} X_2), ((X_2 \mathbin{,} X_1) \mathbin{;} X_2)\}$. This function will be used in the formalization of analytic completion, and demonstrates the advantage of having the syntax of terms parameterized by a type of variables.

Structural rules

The Coq definitions of (analytic) structural rules, and their validity in a particular BI algebra, are given in Figure 4. As one can see, we represent a structural rule as a pair $(\vec{T}, T_0)$, where $\vec{T}$ is a list of bunched terms in the premise of the structural rule, and T_0 is the bunched term representing the conclusion of the rule. For our purposes, we will mainly be using `nat` as a type of variables for bunched terms. The proposition `rule_valid` describes when a rule $(\vec{T}, T_0)$ holds in an algebra `PROP`; the existential quantifier represents disjunction over a set, and we use Coq's dependent product types to accurately describe the domain of the quantification.

The sequent calculus we define is parameterized by a collection of `structural_rule`'s. In order to prove the invertibility of relevant rules (as in Section 7), we need an auxiliary lemma, similar to those in Section 10.1, relating bunch decomposition and linear bunched terms:

Lemma 49 (`bterm_ctx_act_decomp`). *If T is a linear bunched term with variables $x_1, \dots, x_n$, and $T[\vec{\Delta}] = \Pi(\varphi)$ for some bunched context Π, then there is a variable x_j occurring in T, and a context Π' such that*

- $\Delta_j = \Pi'(\varphi)$*;*

[2]We use the type `gset A` of finite sets from the std++ library. The name stands for "generic sets", meaning that `gset A` is generic in the type `A` of elements, under two conditions: the type `A` should be countable and should have decidable equality. In the rest of the text we assume that the type `V` of variables in bunched terms satisfies these conditions.

- *for any bunch* Γ,

$$T[\Delta_1, \ldots, \Delta_{j-1}, \Pi'(\Gamma), \Delta_{j+1}, \ldots, \Delta_n] = \Pi(\Gamma).$$

We demonstrate how to use that lemma for showing the invertibility of the rule $*$L:

Proof (of Lemma 3) We proceed by induction on the proof height and inversion of the derivation. There are many cases to consider, based on the structure of the derivation, which we divide into several groups.

Right rules. Suppose that the last applied rule was a right rule. For some rules (EmpR, TrueR, $-\!*$R, $\to$R, $\vee$R1, $\vee$R2) the process is relatively straightforward. For example, if the last applied rule was

$$\frac{\Delta'(\varphi * \psi) \,;\, \chi_1 \vdash_{\mathsf{cf}} \chi_2}{\Delta'(\varphi * \psi) \vdash_{\mathsf{cf}} \chi_1 \to \chi_2}$$

then we apply the induction hypothesis to obtain a derivation $\Delta'(\varphi \,,\, \psi) \,;\, \chi_1 \vdash_{\mathsf{cf}} \chi_2$ from which we obtain $\Delta'(\varphi \,,\, \psi) \vdash_{\mathsf{cf}} \chi_1 \to \chi_2$.

For other right rules ($*$R, $\wedge$R) we need to reason additionally about contexts. Suppose, for example, that the last applied rule was

$$\frac{\Delta_1 \vdash_{\mathsf{cf}} \chi_1 \qquad \Delta_2 \vdash_{\mathsf{cf}} \chi_2}{\Delta_1 \,,\, \Delta_2 \vdash_{\mathsf{cf}} \chi_1 * \chi_2}$$

where $\Delta(\varphi * \psi) = \Delta_1 \,,\, \Delta_2$. Since $\Delta_1 \,,\, \Delta_2 \rightsquigarrow \langle \Delta(-) \mid \varphi * \psi \rangle$, we know that there exists $\Delta'(-)$ such that one of the two cases hold:

1. $\Delta_1 \rightsquigarrow \langle \Delta' \mid \varphi * \psi \rangle$ and $\Delta(-) = \Delta'(-) \,,\, \Delta_2$;
2. or $\Delta_2 \rightsquigarrow \langle \Delta' \mid \varphi * \psi \rangle$ and $\Delta(-) = \Delta_1 \,,\, \Delta'(-)$.

In both of those cases we use the induction hypothesis on one of the subproofs.

Left rules. Suppose the last applied rule was $*$L:

$$\frac{\Pi(\varphi' \,,\, \psi') \vdash_{\mathsf{cf}} \chi}{\Pi(\varphi' * \psi') \vdash_{\mathsf{cf}} \chi}$$

and $\Pi(\varphi' * \psi') = \Delta(\varphi * \psi)$. By Lemma 46 we can consider several cases. First of all, if $\Pi(-) = \Delta(-)$ and $\varphi' * \psi' = \varphi * \psi$, then $\varphi = \varphi'$ and $\psi = \psi'$ and we get the result that we want immediately from the assumption.

On the other hand, if $\varphi * \psi$ appears inside Π itself we have functions Π_0, Π_1 such that

- $\Pi(\Lambda) = \Pi_0(\Lambda)(\varphi * \psi)$ for any Λ;
- $\Delta(\Lambda) = \Pi_1(\Lambda)(\varphi' * \psi')$ for any Λ;
- $\Pi_0(\Lambda)(\Lambda') = \Pi_1(\Lambda')(\Lambda)$.

Then, since we have a proof of

$$\Pi_0(\varphi' \,,\, \psi')(\varphi * \psi) = \Pi(\varphi' \,,\, \psi') \vdash_{\mathsf{cf}} \chi,$$

we can apply the induction hypothesis and obtain a proof of

$$\Pi_0(\varphi' \,,\, \psi')(\varphi \,,\, \psi) \vdash_{\mathsf{cf}} \chi.$$

Since $\Pi_0(\varphi' \,,\, \psi')(\varphi \,,\, \psi) = \Pi_1(\varphi \,,\, \psi)(\varphi' \,,\, \psi')$, we can apply $*$L to get a derivation of

$$\Delta(\varphi \,,\, \psi) = \Pi_1(\varphi \,,\, \psi)(\varphi' * \psi') \vdash_{\mathsf{cf}} \chi.$$

The other left rules are handled similarly.

Structural rules. The rules AX and CUT are not applicable. For the other structural rules, we proceed the same as in the case of left rules, relying on Lemma 46. ☐

10.4 Analytic completion

We finish this section by describing the formalization of the analytic completion procedure from Section 8. For the rest of this section fix a structural rule $(\vec{T}, T)$.

Linearizing bunched terms.

Given a bunched term T we define it's linearization as follows:

```
Fixpoint linearize_pre (T : bterm nat) (idx : nat)
  : nat * (gmap nat nat) * (bterm nat) :=
  match T with
  | Var x ⇒ (idx + 1, {[ idx := x ]}, Var idx)
  | TComma T1 T2 ⇒
      let '(idx1,m1,T1') := linearize_pre T1 idx in
      let '(idx2,m2,T2') := linearize_pre T2 idx1 in
      (idx2, m1 ∪ m2, TComma T1' T2')
  | TSemic T1 T2 ⇒
      let '(idx1,m1,T1') := linearize_pre T1 idx in
      let '(idx2,m2,T2') := linearize_pre T2 idx1 in
      (idx2, m1 ∪ m2, TSemic T1' T2')
  end.
Definition linearize_bterm (T : bterm nat) : bterm nat
  := snd (linearize_pre T 0).
Definition linearize_bterm_ren (T : bterm nat) : gmap nat nat
  := snd (fst (linearize_pre T 0)).
```

The linearization procedure is defined recursively on the structure on T, replacing each variable with a fresh name and keeping track of that action. The main linearization function is parametrized by the starting index `idx` for fresh variables, and it returns, in addition to the linearized term `T'`, *1)* the updated index `idx'`, which is used in recursive calls, and *2)* a renaming mapping[3] `m`. Almost all the proofs about the linearization function will proceed by first generalizing the starting index for fresh variables from 0 to an arbitrary natural number n, and then proceeding by induction on the structure of the term.

Transformation of premises and analytic completion.

Recall from Section 8, that we need to define a (computable) function $\mathsf{transform}(T_i)$ for each premise T_i, satisfying the following condition:

$$\begin{aligned}&T'[X_1, \ldots, X_k] \in \mathsf{transform}(T_i) \iff \\ &\quad \exists \sigma.\, T'[X_1, \ldots, X_k] = {T_i}^{\bullet}[X_{\sigma(1)}, \ldots, X_{\sigma(n)}] \wedge \forall j.\, \sigma(j) \in r_T^{-1}(r_{T_i}(j))\end{aligned}$$

In Coq, we define this function, as well as the analytic completion of the rule as follows, where `(Ts,T)` is the representation of the structural rule in question:

[3] We use the type `gmap` of finite maps from natural numbers to natural numbers, from the std++ library. The notation `{[idx := x]}` stands for a singleton finite map sending `idx` to `x`. By `m1 ∪ m2` we mean the union of two finite maps.

```
Definition ren_inverse : gmap nat (gset nat) :=
  map_preimage (linearize_bterm_ren T).

Definition transform_premise (Tz : bterm nat) : gset (bterm nat) :=
    let Tz_lin := linearize_bterm Tz in
    let r_Tz := linearize_bterm_ren Tz in
    let ren := λ j, ren_inverse !!! (r_Tz !!! j) in
    bterm_gset_fold (fmap ren Tz_lin).

Definition analytic_completion : structural_rule :=
    (mjoin (map (elements ∘ transform_premise) Ts), linearize_bterm T).
```

The function `ren_invserse` is a finite map, corresponding to the inverse r_T^{-1} of the renaming function. We use the notation `m !!! i` to denote the total lookup function for the map `m`, returning a dummy element if the key `i` is not present in the map.

The function `transform_premise` first computes a bunched term `fmap ren Tz_lin : bterm (gset nat)`, in which variables correspond to sets of variables from the original term. On paper, we might write this as ${T_z}^\bullet[r_T^{-1}(r_{T_z}(1)), \ldots, r_T^{-1}(r_{T_z}(n))]$. We then use the function `bterm_gset_fold` to obtain a set of terms $\{{T_z}^\bullet[Y_1, \ldots, Y_n] \mid \forall j.\, Y_j \in r_T^{-1}(r_{T_z}(j))\}$.

Finally, `analytic_completion` puts it all together, by transforming every premise into a list of premises (`elements ∘ transform_premise`), and then "collapses" the results into one big list of premises (`mjoin`).

11 Related Work

There has been a long line of work on formalizing cut elimination and other meta-theoretical properties of logics in proof assistants. Here, we mention a few recent ones. Pfenning [28] formalized cut elimination for intuitionistic and classical propositional logic in Elf, using only structural induction and avoiding termination measures. Chaudhuri, Lima, and Reis [29] have formalized cut elimination for various fragments of linear logic in Abella. Xavier, Olarte, Reis, and Nigam [30] have formalized cut elimination and completeness of focusing for first-order linear logic in Coq, along with some other meta-theoretical properties. In [31], Dawson and Goré describe their framework for formalizing sequent calculus with explicit structural rules in Isabelle/HOL. They apply their framework for the provability logic GL and formalize the cut elimination argument for it. Their framework was later ported Coq [32] and used to formalize cut elimination for a modal logic Kt. Another proof of cut elimination for GL was formalized in Coq [33]; the authors noticed during the formalization process that the proof can be simplified in several parts.

Tews [34] used Coq to formalize Pattinson's and Schröder's proof [35] of cut elimination for coalgebraic modal logics. During his formalization effort, Tews has uncovered a number of fixable gaps in the proof. As the author puts it themselves:

> A formalization of this extent does always uncover a number of typos and errors in the formalized work. It is a clear sign for the quality and accurateness of the pen-and-paper proofs of Pattinson and Schröder that I found only 4 errors beyond the level of nitpicking.

The formalized proofs mentioned above are syntactic. A formalized semantic proof of cut elimination for the $(\forall, \to, \bot)$ fragment of intuitionistic FOL was given by Herbelin and Lee [36], using Kripke models. The only similar formalization that we are aware is the formalization by Larchey-Wendling [37] of the Okada's semantic proof of cut elimination for linear logic [16, 17]. A similar formalization of cut elimination for implicational relevance logic was used by the author used part of a larger formalization [38]. In personal communication Larchey-Wendling mentioned that he has adapted the aforementioned phase semantics proof to the logic of Bunched Implications, but was not completely satisfied with it.

After Okada's proof, related methods for proving cut elimination were discussed for various logics. For example, Belardinelli, Jipsen, and Ono [39] use intermediate structures (Gentzen structures) to interpret sequent calculi and prove cut elimination for various substructural variants of the Lambek calculus. This method was generalized to handle non-associative logics (i.e. without the exchange rule) [40]. Ciabattoni, Galatos, and Terui [21] prove semantic cut elimination for a wide ride of hypersequent calculi for nonclassical logics.

Galatos and Jipsen [19] introduced the framework of residuated frames which they use to prove cut elimination (and other related properties) for many extensions of Lambek calculus with arbitrary structural rules. The authors later extended their framework [18] to cover extensions of distributive Lambek calculus and BI. [4]

Our proof can be seen as a simplification of the Galatos and Jipsen's method. Instead of making heavy use of the residuated frames, our proof goes directly through algebraic semantics. While this is a less general framework, it still allows us to extend the proof to cover, e.g. modal extensions of BI, which were not covered by the residuated frames framework. We conjecture that the algebra we construct in Section 6 is isomorphic to the Galois algebra constructed in [18, Section 4].

12 Conclusion and Future Work

In this paper we have presented a fully formalized semantic-based proof of cut elimination for the logic of bunched implications. We show that this proof can be extended to cover various extensions of BI, and demonstrated which parts of the proof have to be modified, and which remain unchanged.

As for future work, we see several ways of going forward. Firstly, we can look at extensions of BI. For example, we can probably extend the construction

[4]The residuated frames framework was used to derive other meta-theoretical properties, such as the finite model property. Unfortunately, the finite model property proof in [18] does not hold. The argument there relies on a version of the Curry's lemma (limiting a number of contractions that can occur in a given sequent in a proof search) which does not hold in BI (see [41]).

presented here to cover first-order/predicate BI. The algebra $\mathcal{C}$ is already complete (has all the meets and joins), so it is suitable for interpreting quantifiers. Unfortunately, formalizing this would require dealing with variable binders, which we decided to forgo in this paper. It would also be natural to look at extensions such as GBI [18], extensions of BI with various modalities that are used in separation logic [26, 42], or the recently proposed polarized sequent calculus for BI [43].

Secondly, it would be interesting to go from logic to type theory. The algebra $\mathcal{C}$ is a subalgebra of predicates $\mathit{Bunch} \to \mathit{Prop}$, where Prop is the type of propositions. One can imagine it is possible to consider instead presheaves $\mathit{Bunch} \to \mathit{Set}$, and look for a categorification of $\mathcal{C}$ – a reflexive subcategory of the category of presheaves, which is universal for cut-free provability. That might give us some insight into the connections to the normalization-by-evaluation method for type theories [44], which is usually based on the category of presheaves.

Acknowledgments

The author would like to thank Jorge Pérez, Revantha Ramanayake, Niels van der Weide, and Dominique Larchey-Wendling, for their insightful comments on the earlier version of this article and for pointing me to some of the related work. The author would also like to thank the anonymous reviewers for providing their invaluable feedback.

This work has been supported by the Dutch Research Council (NWO) under project No. 016.Vidi.189.046 (Unifying Correctness for Communicating Software).

Statements and Declarations

Ethical approval.

Not applicable.

Competing interests.

None.

Authors' contributions.

Not applicable.

Funding.

This work has been supported by the Dutch Research Council (NWO) under project No. 016.Vidi.189.046 (Unifying Correctness for Communicating Software).

Availability of data and materials.

The full formalization of this paper available online at https://github.com/co-dan/BI-cutelim.

References

[1] O'Hearn, P., Pym, D.: The Logic of Bunched Implications. The Bulletin of Symbolic Logic **5**(2), 215–244 (1999). https://doi.org/10.2307/421090

[2] Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: LICS, pp. 55–74. IEEE Computer Society, ??? (2002). https://doi.org/10.1109/LICS.2002.1029817

[3] O'Hearn, P.: Separation logic. CACM **62**(2), 86–95 (2019). https://doi.org/10.1145/3211968

[4] Pym, D., O'Hearn, P., Yang, H.: Possible worlds and resources: The semantics of BI. Theoretical Computer Science **315**(1), 257–305 (2004). https://doi.org/10.1016/j.tcs.2003.11.020

[5] Pym, D.: The Semantics and Proof Theory of the Logic of Bunched Implications. Applied Logic Series. Springer Netherlands, ??? (2002). https://doi.org/10.1007/978-94-017-0091-7

[6] Arisaka, R., Qin, S.: LBI Cut Elimination Proof with BI-MultiCut. In: 2012 Sixth International Symposium on Theoretical Aspects of Software Engineering, pp. 235–238 (2012). https://doi.org/10.1109/TASE.2012.30

[7] Borisavljević, M., Došen, K., Petrić, Z.: On permuting cut with contraction. Mathematical Structures in Computer Science **10**(2), 99–136 (2000). https://doi.org/10.1017/S0960129599003011

[8] Brotherston, J.: Bunched Logics Displayed. Studia Logica **100**(6), 1223–1254 (2012)

[9] Pinto, L., Uustalu, T.: Proof Search and Counter-Model Construction for Bi-intuitionistic Propositional Logic with Labelled Sequents. In: Giese, M., Waaler, A. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods. Lecture Notes in Computer Science, pp. 295–309. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02716-1_22

[10] de Paiva, V., Braüner, T.: Cut-Elimination for Full Intuitionistic Linear Logic (1996)

[11] Bierman, G.: A note on full intuitionistic linear logic. Annals of Pure and Applied Logic **79**(3), 281–287 (1996). https://doi.org/10.1016/

0168-0072(96)00004-8

[12] Brünnler, K., Straßburger, L.: Modular Sequent Systems for Modal Logic. In: Giese, M., Waaler, A. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods. Lecture Notes in Computer Science, pp. 152–166. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02716-1_12

[13] Marin, S., Straßburger, L.: Label-free modular systems for classical and intuitionistic modal logics. In: Advances in Modal Logic 10, Groningen, Netherlands (2014)

[14] Sambin, G., Valentini, S.: The modal logic of provability. The sequential approach. Journal of Philosophical Logic **11**(3), 311–342 (1982). https://doi.org/10.1007/BF00293433

[15] Goré, R., Ramanayake, R.: Valentini's cut-elimination for provability logic resolved. The Review of Symbolic Logic **5**(2), 212–238 (2012). https://doi.org/10.1017/S1755020311000323

[16] Okada, M.: Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic. Theoretical Computer Science **227**(1-2), 333–396 (1999). https://doi.org/10.1016/S0304-3975(99)00058-4

[17] Okada, M.: A uniform semantic proof for cut-elimination and completeness of various first and higher order logics. Theoretical Compututer Science **281**(1), 471–498 (2002). https://doi.org/10.1016/S0304-3975(02)00024-5

[18] Galatos, N., Jipsen, P.: Distributive residuated frames and generalized bunched implication algebras. Algebra universalis **78**(3), 303–336 (2017). https://doi.org/10.1007/s00012-017-0456-x

[19] Galatos, N., Jipsen, P.: Residuated Frames with Applications to Decidability. Transactions of the American Mathematical Society **365**(3), 1219–1249 (2013)

[20] Everett, C.J.: Closure Operators and Galois Theory in Lattices. Transactions of the American Mathematical Society **55**(3), 514–525 (1944). https://doi.org/10.2307/1990306

[21] Ciabattoni, A., Galatos, N., Terui, K.: From Axioms to Analytic Rules in Nonclassical Logics. In: 2008 23rd Annual IEEE Symposium on Logic in Computer Science, pp. 229–240 (2008). https://doi.org/10.1109/LICS.2008.39

[22] Bierman, G., de Paiva, V.: On an Intuitionistic Modal Logic. Studia Logica **65**(3), 383–416 (2000). https://doi.org/10.1023/A:1005291931660

[23] Alechina, N., Mendler, M., de Paiva, V., Ritter, E.: Categorical and Kripke Semantics for Constructive S4 Modal Logic. In: Fribourg, L. (ed.) Computer Science Logic. Lecture Notes in Computer Science, pp. 292–307. Springer, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-44802-0_21

[24] Iris team: The Iris Project website and Coq development. https://iris-project.org/. Accessed: 2021-09-08 (2021)

[25] Krebbers, R., Jourdan, J., Jung, R., Tassarotti, J., Kaiser, J., Timany, A., Charguéraud, A., Dreyer, D.: MoSeL: A general, extensible modal framework for interactive proofs in separation logic. PACMPL **2**(ICFP), 77–17730 (2018). https://doi.org/10.1145/3236772

[26] Bizjak, A., Birkedal, L.: On Models of Higher-Order Separation Logic. Electronic Notes in Theoretical Computer Science **336**, 57–78 (2018). https://doi.org/10.1016/j.entcs.2018.03.016

[27] std++ developers: An extended "standard" library for Coq. https://gitlab.mpi-sws.org/iris/stdpp/. Accessed: 2022-05-27 (2022)

[28] Pfenning, F.: Structural cut elimination: I. intuitionistic and classical logic. Information and Computation **157**(1-2), 84–141 (2000). https://doi.org/10.1006/inco.1999.2832

[29] Chaudhuri, K., Lima, L., Reis, G.: Formalized Meta-Theory of Sequent Calculi for Substructural Logics. Electronic Notes in Theoretical Computer Science **332**, 57–73 (2017). https://doi.org/10.1016/j.entcs.2017.04.005

[30] Xavier, B., Olarte, C., Reis, G., Nigam, V.: Mechanizing Focused Linear Logic in Coq. Electronic Notes in Theoretical Computer Science **338**, 219–236 (2018). https://doi.org/10.1016/j.entcs.2018.10.014

[31] Dawson, J.E., Goré, R.: Generic Methods for Formalising Sequent Calculi Applied to Provability Logic. In: Fermüller, C.G., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning. Lecture Notes in Computer Science, pp. 263–277. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16242-8_19

[32] D'Abrera, C., Dawson, J., Goré, R.: A Formally Verified Cut-Elimination Procedure for Linear Nested Sequents for Tense Logic. In: Das, A., Negri, S. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods. Lecture Notes in Computer Science, pp. 281–298. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_17

[33] Goré, R., Ramanayake, R., Shillito, I.: Cut-Elimination for Provability Logic by Terminating Proof-Search: Formalised and Deconstructed Using Coq. In: Das, A., Negri, S. (eds.) Automated Reasoning with Analytic

Tableaux and Related Methods. Lecture Notes in Computer Science, pp. 299–313. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-86059-2_18

[34] Tews, H.: Formalizing Cut Elimination of Coalgebraic Logics in Coq. In: Galmiche, D., Larchey-Wendling, D. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods. Lecture Notes in Computer Science, pp. 257–272. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40537-2_22

[35] Pattinson, D., Schröder, L.: Cut elimination in coalgebraic logics. Information and Computation **208**(12), 1447–1468 (2010). https://doi.org/10.1016/j.ic.2009.11.008

[36] Herbelin, H., Lee, G.: Forcing-Based Cut-Elimination for Gentzen-Style Intuitionistic Sequent Calculus. In: Ono, H., Kanazawa, M., de Queiroz, R. (eds.) Logic, Language, Information and Computation. Lecture Notes in Computer Science, pp. 209–217. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02261-6_17

[37] Larchey-Wendling, D.: Semantic Cut-Elimination for ILL via relational phase semantics. https://github.com/DmxLarchey/Coq-Phase-Semantics. Accessed: 2021-09-08 (2021)

[38] Larchey-Wendling, D.: Constructive Decision via Redundancy-Free Proof-Search. Journal of Automated Reasoning **64**(7), 1197–1219 (2020). https://doi.org/10.1007/s10817-020-09555-y

[39] Belardinelli, F., Jipsen, P., Ono, H.: Algebraic Aspects of Cut Elimination. Studia Logica **77**(2), 209–240 (2004). https://doi.org/10.1023/B:STUD.0000037127.15182.2a

[40] Galatos, N., Ono, H.: Cut elimination and strong separation for substructural logics: An algebraic approach. Annals of Pure and Applied Logic **161**(9), 1097–1133 (2010). https://doi.org/10.1016/j.apal.2010.01.003

[41] Jipsen, P., Litak, T.: An Algebraic Glimpse at Bunched Implications and Separation Logic. To appear in "Outstanding Contributions: Hiroakira Ono on Residuated Lattices and Substructural Logics". (2018)

[42] Dockins, R., Appel, A.W., Hobor, A.: Multimodal Separation Logic for Reasoning About Operational Semantics. Electronic Notes in Theoretical Computer Science **218**, 5–20 (2008). https://doi.org/10.1016/j.entcs.2008.10.002

[43] Gheorghiu, A., Marin, S.: Focused proof-search in the logic of bunched implications. In: Kiefer, S., Tasson, C. (eds.) Foundations of Software

Science and Computation Structures - 24th International Conference, FOSSACS 2021. Lecture Notes in Computer Science, vol. 12650, pp. 247–267. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-71995-1_13

[44] Altenkirch, T., Hofmann, M., Streicher, T.: Categorical reconstruction of a reduction free normalization proof. In: Pitt, D., Rydeheard, D.E., Johnstone, P. (eds.) Category Theory and Computer Science. Lecture Notes in Computer Science, pp. 182–199. Springer, Berlin, Heidelberg (1995)