Focused Proof-search in the Logic of Bunched Implications

Alexander Gheorghiu and Sonia Marin

University College London, United Kingdom
{alexander.gheorghiu.19, s.marin}@ucl.ac.uk

Abstract. The logic of Bunched Implications (BI) freely combines additive and multiplicative connectives, including implications. It allows for a natural interpretation of formulas as the shared and separated resources required for the construction of a goal; nonetheless, despite its well-studied proof theory, proof-search in BI has always been a difficult problem. Focusing is a restriction of the proof-search space that can capture various goal-directed proof-search procedures; in this paper, we study these procedures in BI. To show that focused proof-search is complete, we first reformulate the traditional sequent calculus for BI using the simpler data-structure of nested sequents; then we propose a polarised and focused variant of this calculus that we show is sound and complete via a cut-elimination argument. This establishes an operational semantics for focused proof-search in the logic of Bunched Implications.

1 Introduction

The Logic of Bunched Implications (BI) [26] is noted for its handling of formulas as resources required for the derivation of a goal. It diverges from related logics e.g., Linear Logic (LL) [18], by classifying the resources as shared or separated, as opposed to restricted or unrestricted. Consequently, the logic has natural applications in systems modelling [28]; and, a particular theory of BI called Separation Logic [32] has found industrial use in program verification e.g., [3]. Despite having a well-developed and well-studied proof theory [29], comparatively little formal study has been made on proof-search in BI.\(^1\)

A large class of goal-directed proof-search procedures are captured by the focusing principle which generalises the uniform proof-search procedure underpinning the operational semantics of logic programming languages [24,1,14,8]. Indeed, focused proofs have been used to give both a general account of proof in related logics [23,25,21], as well as structural operational semantics for computations used in industry e.g., algorithms which solve constraint satisfaction problems [15]. In the following we demonstrate that the focusing principle holds for the traditional sequent calculus LBI via the completeness of an operational semantics which we call the focused system. This subsumes the completeness

\(^1\) Decidability of the logic has been demonstrated via the finite model property [16], but the existence of an effective proof-search procedure remains an open problem.
theorem for the hereditary Harrop fragment with respect to (unit-simple) uniform proofs [2]. Proof-search within related systems with less well-behaved proof theory has been better studied; e.g., Boolean BI [27,19,4], which does not have a known sequent calculus, but only display/labelled systems.

Syntactically BI combines additive and multiplicative conjunctions i.e., $\land$ and $\ast$, and corresponding intuitionistic implications i.e., $\rightarrow$ and $\rightarrow \ast$. Rather than distinguishing the two types of resources via modalities, two context-formers are used; consequently, contexts in BI are not lists, nor multisets, but instead are bunches: binary trees whose leaves are formulas and internal nodes context-formers. In particular, additive composition $(\Gamma; \Delta)$ admits the structural rules of weakening and contraction, whereas multiplicative composition $(\Gamma, \Delta)$ denies them. The principal technical challenges when studying proof-search in BI arise from the interaction between the additive and multiplicative fragments and the complex structure of contexts. We overcome these challenges by restricting the application of these structural rules in LBI as well as working with a representation of bunches as nested multisets.

The focusing principle was introduced for Linear Logic [1] and is characterised by alternating focused and unfocused phases of goal-directed proof-search: the unfocused phase comprises rules which are safe to apply i.e., rules where provability is invariant; conversely, the focused phase comprises the reduction of a formula and its sub-formulas where potentially invalid sequents may arise. The mechanism enforcing the structure is a partition of the set of formulas into two classes: positive and negative. For negative formulas provability is invariant with respect to the application of a right rule, and for positive formulas, of a left rule; however, in the contrasting case, the application of a left rule to a negative formula, or of a right rule to a positive formula, may result in invalid sequents. During focused proof-search the unfocused phases are comprised of safe reductions performed eagerly; meanwhile, the focused phases are comprised of the reduction of a formula and its sub-formulas until only safe reductions are left. We say that the focusing principle holds when every provable sequent has a focused proof.

The original proof of the focusing principle in Linear Logic was via long and tedious permutations of rules [1]. In this paper, we use for BI a different methodology, originally presented in [20], which has since been implemented in a variety of logics [21,6,7] and proof systems [14]. The method is as follows: given a sequent calculus, first one polarises the syntax according to the positive/negative behaviours; second, one gives a focused variation of the sequent calculus where the control flow of proof-search is managed by polarisation; third, one shows that this system admits cuts (the only non-analytic rule); and, finally, one shows that in the presence of cut the given sequent calculus may be simulated in the focused one. When the polarised system is complete, the focusing principle holds.

In LBI certain rules e.g., the structural rules, have no natural placement in either the focused or the unfocused phases of proof-search. Thus, a design choice is made: to eliminate/constrain these rules, or to permit them without restriction. Eliminating and/or constraining them gives better control on proof-search;
on the other hand, allowing them freely achieves a more well-behaved proof theoretic meta-theory. In this paper, we choose the former as our motivation is to study computational behaviour of proof-search in BI. The only case where confinement is not possible is the exchange rule. In standard sequent calculi the exchange rule is made implicit by working with a more convenient data-structure e.g., multisets as opposed to lists; however, the additional structure of bunches in BI means that a more complex alternative is required. The solution presented is to use nested multisets of two types (additive and multiplicative) corresponding to the two different context-formers/conjunctions.

In Section 2 we present the logic of Bunched Implications; in particular, Section 2.1 and Section 2.2 contain the background on BI i.e., the syntax and sequent calculus respectfully; meanwhile, Section 2.3 gives representation of bunches as nested multisets. Section 3 contains the focused system: first, in Section 3.1 we introduce the polarised syntax; second, in Section 3.2 we introduce the focused sequents calculus and some metatheory, most importantly the cut-admissibility result; finally, in Section 3.3 we give the completeness theorem, from which the validity of focusing principle follows as a corollary. We conclude in Section 4 with some further discussion and future directions.

2 Re-presentations of BI

2.1 Traditional Syntax

The natural setting for BI is at the level of sequents (as opposed to simply formulas), and it can be seen as the free combination of the syntax of intuitionistic logic and the multiplicative fragment of intuitionistic linear logic, which imposes the presence of two distinct context-formers. That is to say, the two conjunctions $\land$ and $\ast$ are represented at the meta-level as context-formers $;$ and $,$ respectively, which are both commutative and associative, but do not distribute.

**Definition 1 (Formulas).** Let $P$ be a denumerable set of propositional letters. The formulas of BI are defined by the following grammar, where $A \in P$,

$$\varphi, \psi ::= \top | \bot | \top^* | A | (\varphi \land \psi) | (\varphi \lor \psi) | (\varphi \to \psi) | (\varphi \ast \psi) | (\varphi \ast \ast \psi)$$

Let $\circ$ be a connective; if $\circ \in \{\land, \lor, \to, \top\}$ then it is an additive connective; and, if $\circ \in \{\ast, \ast \ast, \top\}$ then it is a multiplicative connective. The set of all formulas is denoted $F$.

Formulas are combined into two-sorted trees called bunches.

**Definition 2 (Bunch).** A bunch is constructed with the following grammar, where $\varphi \in F$:

$$\Delta ::= \varphi | \varnothing_+ | \varnothing_- | (\Delta; \Delta) | (\Delta, \Delta)$$

The symbols $\varnothing_+$ and $\varnothing_-$ are the additive and multiplicative units respectively; and, the symbols $;$ and $,$ are the additive and multiplicative context-formers respectively. A bunch is basic if it is a formula, $\varnothing_+$, or $\varnothing_-;$ and complex otherwise.
The set of all bunches is denoted $\mathbb{B}$, the set of complex bunches with principal context-former additive by $\mathbb{B}^+$, and the set of complex bunches with principal context-former multiplicative by $\mathbb{B}^\times$.

For two bunches $\Delta, \Delta' \in \mathbb{B}$ if $\Delta'$ is a sub-tree of $\Delta$, it is called a sub-bunch. We may use $\Delta(\Delta')$ to denote that $\Delta'$ is a sub-bunch of $\Delta$, in which case $\Delta(\Delta'\circ \delta)$ is the result of replacing the occurrence of $\Delta'$ by $\Delta''$. If $\delta$ is a basic sub-bunch of $\Delta$, then the context-former $\circ$ is said to be its principal context-former in $\Delta(\Delta'\circ \delta)$ (and $\Delta(\delta \circ \Delta')$).

**Example 3.** Let $\Delta = (\varphi,(\chi;\emptyset_+)); (\psi;(\psi;\emptyset_\times))$, this may be written as $\Delta(\varphi,(\chi;\emptyset_+))$ such that $\Delta(\varphi;\varphi) = (\varphi;\varphi); (\psi;(\psi;\emptyset_\times))$.

**Definition 4 (Bunched Sequent).** A bunched sequent is a pair of a bunch $\Delta$, called the context, and a formula $\varphi$, denoted $\Delta \Rightarrow \varphi$.

Bunches are intended to be considered up-to congruence ($\equiv$); analogously to other calculi where contexts are lists, but considered modulo permutation to become multisets. It is the least relation satisfying:

- Commutative monoid equations for $;$ with unit $\emptyset_+$,
- Commutative monoid equations for $\cdot$, with unit $\emptyset_\times$,
- Coherence i.e., if $\Delta' \equiv \Delta''$ then $\Delta(\Delta') \equiv \Delta(\Delta'')$.

It is useful to have a measure on sub-bunches which can identify their distance from the root node.

**Definition 5 (Rank).** If $\Delta'$ is a sub-bunch of $\Delta$, then $\rho(\Delta')$ is the number of context-former alternations between the principal context-former of $\Delta'$, and the principal context-former of $\Delta$.

Let $\Delta$ be a complex bunch, we use $\Delta' \in \Delta$ to denote that $\Delta'$ is a (proper) top-most sub-bunch i.e., $\Delta$ is a sub-bunch satisfying $\Delta \neq \Delta'$ but $\rho(\Delta') = 0$.

**Example 6.** Let $\Delta$ be as in Example 3, then $\rho(\emptyset_+) = 2$ whereas $\rho(\emptyset_\times) = 0$; and, $(\varphi,(\chi,\emptyset_\times)) \in \Delta$. Consider the parse-tree of $\Delta$:

```
  [ varphi ]
    ^
   / \           \\
  [ chi ] [ emptyset_+ ] [ psi ] [ emptyset_\times ]
```

Reading upward from $\emptyset_+$ one encounters first $;$ which changes into $\cdot$, and then back to $;$ so the rank is 2; whereas counting up from $\emptyset_\times$ one only encounters $;$ so the rank is 0.
2.2 Sequent Calculus

There is both a natural deduction system [26] and a sequent calculus for BI [29]. The sequent calculus is a more natural setting for proof-search since analytic proofs are complete.

Definition 7 (System LBI). The bunched sequent calculus LBI is composed of the rules in Figure 1.

The classification of $\land$ as additive may seem dubious upon reading the $\land R$ rule, but the designation arises from the use of the structural rules; i.e., the $\land R$ and $\to R$ rules may be replaced by additive variants without loss of generality. The presentation in Figure 1 is as in [29] and simply highlights the nature of the additive and multiplicative context-formers. Nonetheless, the choice of rule does affect proof-search behaviours; the consequences are discussed in more detailed in Section 3.1.

Lemma 8 (Cut-elimination [29]). If $\varphi$ has a LBI-proof, then it has a cut-free LBI-proof.

Throughout, unless specified otherwise, take proof to mean cut-free proof. Moreover, if $L$ is a sequent calculus we use $\vdash_L \Delta \Rightarrow \varphi$ to denote that there is an $L$-proof of $\Delta \Rightarrow \varphi$. Further, if $R$ is a rule, then we may denote $L + R$ to denote the sequent calculus combining the rules of $L$ with $R$.

The following will allow for generalised axioms to be used in proof-construction later on.
Lemma 9. $\vdash_{LBI} \varphi \Rightarrow \varphi$

Proof. Follows from induction on complexity of $\varphi$. $\square$

The remainder of this section is the meta-theory required to control the structural rules, which pose the main issue to the study of proof search in BI.

Lemma 10. The following rules are derivable in $LBI$, and replacing $W$ with them does not affect the completeness of the system:

\[
\begin{align*}
\frac{\Delta \Rightarrow \varphi}{\Delta, \Delta \Rightarrow \varphi} & \quad \frac{\Delta' \Rightarrow \psi}{\Delta', \psi \Rightarrow \varphi} & \quad \frac{\Delta'' \Rightarrow \chi}{\Delta', \varphi \Rightarrow \Delta''} \\
\frac{\Delta, A \Rightarrow A'}{\Delta, A \Rightarrow A'} & \quad \frac{\Delta; \varphi \Rightarrow \Delta; \psi}{\Delta; \varphi \Rightarrow \psi} & \quad \frac{\Delta; \varphi \Rightarrow \chi}{\Delta; \varphi \Rightarrow \chi}
\end{align*}
\]

Proof. Derivations follow by use of the structural rules. Let $LBI'$ to be $LBI$ without $W$ but with these new rules (retaining also $\ast R$, $\Rightarrow L$, $\Rightarrow R$, and $Ax$), then $W$ is admissible in $LBI'$ using standard permutation argument. $\square$

One may regard the above modification to $LBI$ as forming a new calculus, but since all the new rules are derivable it is really a restriction of the calculus.

### 2.3 Nested Calculus

Originally, sequents in the sequent calculi for classical and intuitionistic logics ($LK$ and $LJ$ respectively) were introduced as lists, and a formal exchange rule was required to permute elements when needed for a logical rule to be applied [17]. However, in practice, the rule is often suppressed, and contexts are simply presented as multisets of formulas. This reduces the number of steps/choices being made during proof-search without increasing the complexity of the underlying data structure. Bunches have considerably more structure than lists, but a quotient with respect to congruence can be made resulting in two-sorted nested multisets; this was first suggested in [13], though never formally realised.

**Definition 11 (Two-sorted Nest).** Nests ($\Gamma$) are formulas or multisets, ascribed either additive ($\Sigma$), or multiplicative ($\Pi$) kind, containing nests of the opposite kind i.e.,

\[
\begin{align*}
\Gamma := & \, \varphi \mid \Sigma \mid \Pi \\
\Sigma := & \, \varphi \mid \{\Pi_1, ..., \Pi_n\}_+ \\
\Pi := & \, \varphi \mid \{\Sigma_1, ..., \Sigma_n\}_x
\end{align*}
\]

The constructors are multiset constructors which may be empty in which case the nests are denoted $\varnothing_+$ and $\varnothing_x$ respectively. No multiset is a singleton; and the set of all nests is denoted $\mathbb{B}/\equiv$.

We write $A \in \Gamma$ to denote either that $A = \Gamma$, if $\Gamma$ is a formula, or that $A$ is an element of the multiset $\Gamma$ otherwise. Furthermore, we write $A \subseteq \Gamma$ to denote $\forall \gamma \in \mathbb{B}/\equiv$ if $\gamma \in A$ then $\gamma \in \Gamma$. The notation $\Gamma\{A\}_+$ (resp. $\Gamma\{A\}_x$), denotes that $A$ is a sub-nest of $\Gamma$ of additive (resp. multiplicative) kind; we may use $\Gamma\{A\}$ when the kind is not specified. In either case $\Gamma\{A'\}$ denotes the substitution of $A$ for $A'$. A promotion in the syntax tree may be required after a substitution either to handle a singleton or an improper alternation of constructor types.
Example 12. The following inclusions are valid,
\[ \{\varphi, \chi\} + \subseteq \left\{ \{\varphi, \chi\} +, \psi, \varnothing_\times \right\}_+ = \Gamma\{\{\varphi, \chi\} +\} \]

It follow that \( \Gamma\{\{\varphi, \varphi\} +\} = \{\varphi, \varphi, \psi, \varnothing_\times \} +\). Note the absence of the \( \{\cdot\} +\) constructor after substitution, this is due to a promotion in the syntax tree to avoid having two nested additive constructors. Similarly, since \( \varnothing_\times \) denotes the empty multiset of multiplicative kind, substituting \( \chi \) with it gives \( \{\varphi, \psi, \varnothing, \varnothing_\times \} +\); i.e., first the improper \( \{\varphi, \varnothing_\times \} +\) becomes \( \{\varphi\} +\); then, the resulting singleton \( \{\varphi\} +\) is promoted to \( \varphi \).

Typically we will only be interested in fragments of sub-nests so we have the following abuse of notation, where \( \circ \in \{+, \times\} \):
\[ \Gamma\{\{\Pi_1, \ldots, \Pi_i\} \circ, \Pi_{i+1}, \ldots, \Pi_n\}_\circ := \Gamma\{\Pi_1, \ldots, \Pi_n\}_\circ \]

The notion of rank has a natural analogue in this setting.

**Definition 13 (Depth, Rank).** Let \( \circ \in \{+, \times\} \) be a nest, we define the depth on \( \mathbb{B} \) as follows:
\[ \delta(\varphi) := 0 \quad \delta(\{\Gamma_1, \ldots, \Gamma_n\}_\circ) := \max\{\delta(\Gamma_1), \ldots, \delta(\Gamma_n)\} + 1 \]

The equivalence of the two presentations i.e., bunches and nests, follows from a moral\(^2\) inverse between a nestifying function \( \eta \) and a bunching function \( \beta \). The transformation \( \beta \) is simply going from a tree with arbitrary branching to a binary one, and \( \eta \) is the reverse.

**Definition 14 (Canonical Translation).** The canonical translation \( \eta : \mathbb{B} \to \mathbb{B}/\equiv \) is defined recursively as follows,
\[
\eta(\Delta) := \begin{cases} 
\Delta & \text{if } \Delta \in \mathbb{F} \cup \{\varnothing_+, \varnothing_\times\} \\
\{\eta(\Delta') \in \mathbb{B}/\equiv | \Delta' \in \Delta \text{ and } \Delta' \in \mathbb{B}_+\} + & \text{if } \Delta \in \mathbb{B}_+ \\
\{\eta(\Delta') \in \mathbb{B}/\equiv | \Delta' \in \Delta \text{ and } \Delta' \in \mathbb{B}_\times\} \times & \text{if } \Delta \in \mathbb{B}_\times 
\end{cases}
\]

The canonical translation \( \beta : \mathbb{B}/\equiv \to \mathbb{B} \) is defined recursively as follows,
\[
\beta(\Gamma) := \begin{cases} 
\Gamma & \text{if } \Gamma \in \mathbb{F} \cup \{\varnothing_+, \varnothing_\times\} \\
\beta(\Pi_1); (\beta(\Pi_2); ...) & \text{if } \Gamma = \{\Pi_1, \Pi_2, \ldots\} + \\
(\beta(\Sigma_1), (\beta(\Sigma_2), \ldots)) & \text{if } \Gamma = \{\Sigma_1, \Sigma_2, \ldots\} \times 
\end{cases}
\]

Example 15. Applying \( \eta \) to the bunch in Example 3 gives the nest in Example 12:

\[
\begin{tikzpicture}
\node (X) at (0,0) {$\varnothing_\times$};
\node (P) at (1,-1) {$\psi$};
\node (Q) at (1,-2) {$\varnothing_\times$};
\node (R) at (0,-2) {$\psi$};
\node (S) at (0,-3) {$\varnothing_\times$};
\node (T) at (0,-4) {$\varnothing_\times$};
\node (U) at (1,-4) {$\psi$};
\node (V) at (1,-5) {$\varnothing_\times$};
\node (W) at (0,-5) {$\varnothing_\times$};
\node (A) at (1,-6) {$\varnothing_\times$};
\node (B) at (0,-6) {$\varnothing_\times$};
\node (C) at (0,-7) {$\varnothing_\times$};
\node (D) at (1,-7) {$\varnothing_\times$};
\node (E) at (1,-8) {$\varnothing_\times$};
\node (F) at (0,-8) {$\varnothing_\times$};
\node (G) at (1,-9) {$\varnothing_\times$};
\node (H) at (0,-9) {$\varnothing_\times$};
\node (I) at (1,-10) {$\varnothing_\times$};
\node (J) at (0,-10) {$\varnothing_\times$};
\node (K) at (1,-11) {$\varnothing_\times$};
\node (L) at (0,-11) {$\varnothing_\times$};
\node (M) at (1,-12) {$\varnothing_\times$};
\node (N) at (0,-12) {$\varnothing_\times$};
\node (O) at (1,-13) {$\varnothing_\times$};
\node (P) at (0,-13) {$\varnothing_\times$};
\node (Q) at (1,-14) {$\varnothing_\times$};
\node (R) at (0,-14) {$\varnothing_\times$};
\node (S) at (1,-15) {$\varnothing_\times$};
\node (T) at (0,-15) {$\varnothing_\times$};
\node (U) at (1,-16) {$\varnothing_\times$};
\node (V) at (0,-16) {$\varnothing_\times$};
\node (W) at (1,-17) {$\varnothing_\times$};
\node (X) at (0,-17) {$\varnothing_\times$};
\node (Y) at (1,-18) {$\varnothing_\times$};
\node (Z) at (0,-18) {$\varnothing_\times$};
\node (AA) at (1,-19) {$\varnothing_\times$};
\node (BB) at (0,-19) {$\varnothing_\times$};
\node (CC) at (1,-20) {$\varnothing_\times$};
\node (DD) at (0,-20) {$\varnothing_\times$};
\node (EE) at (1,-21) {$\varnothing_\times$};
\node (FF) at (0,-21) {$\varnothing_\times$};
\node (GG) at (1,-22) {$\varnothing_\times$};
\node (HH) at (0,-22) {$\varnothing_\times$};
\node (II) at (1,-23) {$\varnothing_\times$};
\node (JJ) at (0,-23) {$\varnothing_\times$};
\node (KK) at (1,-24) {$\varnothing_\times$};
\node (LL) at (0,-24) {$\varnothing_\times$};
\node (MM) at (1,-25) {$\varnothing_\times$};
\node (NN) at (0,-25) {$\varnothing_\times$};
\node (OO) at (1,-26) {$\varnothing_\times$};
\node (PP) at (0,-26) {$\varnothing_\times$};
\node (QQ) at (1,-27) {$\varnothing_\times$};
\node (RR) at (0,-27) {$\varnothing_\times$};
\node (SS) at (1,-28) {$\varnothing_\times$};
\node (TT) at (0,-28) {$\varnothing_\times$};
\node (UU) at (1,-29) {$\varnothing_\times$};
\node (VV) at (0,-29) {$\varnothing_\times$};
\node (WW) at (1,-30) {$\varnothing_\times$};
\node (XX) at (0,-30) {$\varnothing_\times$};
\node (YY) at (1,-31) {$\varnothing_\times$};
\node (ZZ) at (0,-31) {$\varnothing_\times$};
\node (AAA) at (1,-32) {$\varnothing_\times$};
\node (BBB) at (0,-32) {$\varnothing_\times$};
\node (CCC) at (1,-33) {$\varnothing_\times$};
\node (DDD) at (0,-33) {$\varnothing_\times$};\end{tikzpicture}
\]

\(^2\) In the sense that bunches are intended to be considered modulo congruence.
Lemma 16. The translations are inverses up-to congruence i.e.,

1. if $\Delta \in \mathbb{B}$ then $(\beta \circ \eta)(\Delta) \equiv \Delta$;
2. if $\Gamma \in \mathbb{B} \equiv \Gamma$ then $(\eta \circ \beta)(\Gamma) \equiv \Gamma$;
3. let $\Delta, \Delta' \in \mathbb{B}$, then $\Delta \equiv \Delta'$ if and only if $\eta(\Delta) = \eta(\Delta')$.

Proof. The first two statements follow by induction on the depth (either for bunches or nests), where one must take care to consider the case of a context consisting entirely of units. The third statement employs the first in the forward direction, and proceeds by induction on depth in the reverse direction. \hfill \Box

Definition 17 (System $\eta$LBI). The nested sequent calculus $\eta$LBI is composed of the rules in Figure 2, where the metavariables denote possibly empty nests.

Observe the use of metavariable $\Gamma'$ instead of $\Pi$ (resp. $\Sigma$) as sub-contexts. This allows the inferences

\[
\begin{align*}
\{\Sigma_0, \ldots, \Sigma_i\}_x \Rightarrow \varphi, \\
\{\Sigma_0, \ldots, \Sigma_n\} \Rightarrow \varphi \ast \psi
\end{align*}
\]

to be captured by a single figure. In practice it implements the abuse of notation given above:

\[
\{\{\Sigma_0, \ldots, \Sigma_i\}_x, \{\Sigma_{i+1}, \ldots, \Sigma_n\}_x\} \Rightarrow \varphi \ast \psi
\]

Lemma 18 (Soundness and Completeness of $\eta$LBI). Systems LBI and $\eta$LBI are equivalent:

Soundness: If $\vdash_{\etaLBI} \Gamma \Rightarrow \varphi$ then $\vdash_{\LBI} \beta(\Gamma) \Rightarrow \varphi$;
Completeness: If $\vdash_{\LBI} \Delta \Rightarrow \varphi$ then $\vdash_{\etaLBI} \eta(\Delta) \Rightarrow \varphi$.

Example 19. The following is a proof in $\eta$LBI using $W$ and $Ax$ to close the leaves; more precisely, it is a proof in the complete sub-system arising in Lemma 10 using nested sequents:
At no point in this section will we refer to bunches, thus the variable \( \Delta \), which has so far been reserved for elements of \( \mathbb{B} \), is re-appropriated.

### 3.1 Polarisation

Polarity in the focusing principle is determined by the invariance of provability under application of a rule, thus it is determined by the proof rules themselves. Traditionally, the distinction between positive and negative connectives is apparent in them having either synchronous or asynchronous behaviour. For example, the \( *R \) and \( \rightarrow L \) highlight the synchronous behaviour of the multiplicative connectives since the structure of the context affects the applicability of the rule. Note that since \( \rightarrow L \) is a left rule this make \( \rightarrow \) a negative connective. Although it is perhaps proof-theoretically displeasing to incorporate weakening inside the operational rules as in \( \rightarrow L' \) and \( *R' \), it has good computational behaviour during focused proof-search since the reduction of \( \varphi \rightarrow \psi \) can only arise out of an explicit choice made earlier in the computation.

Since each connective has one right rule and one left rule, polarity can simply be assigned by case analysis. For example, consider the inverses of the \( \lor L \) rule i.e.,

\[
\Gamma \{ \varphi \lor \psi \} \Rightarrow \chi \\
\Gamma \varphi \Rightarrow \chi \\
\Gamma \psi \Rightarrow \chi
\]

They are derivable in \( \text{LBI} \) with \text{cut} (the left branch being closed using Lemma 9) and therefore admissible in \( \text{LBI} \) without \text{cut} (by Lemma 8).

\[
\varphi \Rightarrow \varphi \\
\varphi \Rightarrow \varphi \lor \psi \\
\Gamma \varphi \Rightarrow \chi \quad \text{cut} \\
\Gamma \varphi \lor \psi \Rightarrow \chi \\
\psi \Rightarrow \varphi \lor \psi \\
\psi \Rightarrow \chi \\
\Gamma \psi \Rightarrow \chi \\
\Gamma \psi \lor \psi \Rightarrow \chi \quad \text{cut}
\]

This means that provability is invariant upon application of \( \lor L \): it can always be reverted if needed:

\[
\Gamma \varphi \Rightarrow \chi \\
\Gamma \psi \Rightarrow \chi \\
\Gamma \varphi \lor \psi \Rightarrow \chi \quad \lor L \\
\Gamma \psi \lor \psi \Rightarrow \chi
\]

This is one way of justifying the assignment of polarity to connectives. However, dual connectives do not necessarily have dual behaviours in terms of provability invariance, on the left and on the right. For example, consider the four
possible rules for $\land$:

\[
\frac{\Gamma(\phi_i)}{\Gamma(\phi_1 \land \phi_0)} \quad \frac{\Gamma(\phi)}{\Gamma \land \psi}
\]
\[
\frac{\Gamma(\phi, \psi)_+}{\chi} \quad \frac{\Gamma}{\phi \land \psi}
\]
\[
\frac{\Gamma(\phi_1 \land \phi_0)}{\chi} \quad \frac{\Gamma(\phi)}{\phi \land \psi}
\]
\[
\frac{\Gamma(\phi \land \psi)_+}{\chi} \quad \frac{\Gamma}{\phi \land \psi}
\]
\[
\frac{\Gamma}{\phi \land \psi}
\]

All of these rules are sound, and replacing the conjunction rules in LBI with any pair of a left and right rule will result in a sound and complete system. Indeed, the rules are inter-derivable when the structural rules are present, but otherwise they can be paired to form two sets of rules which have essentially different proof-search behaviours. That is, the rules in the top-row make $\land$ negative; conversely, the bottom row makes $\land$ positive. Each conjunction also comes with an associated unit; i.e., $\top^-$ for negative conjunction, and $\top^+$ for positive conjunction. We will add them all to our system in order to have access to those different proof search behaviours at will. Finally, the polarity of the propositional letters can be assigned arbitrarily as long as only one for each.

**Definition 20 (Polarised Syntax).** Let $P \sqcup P^{-}$ be a partition of $P$, and let $A^\pm \in P^\pm$, then the polarised formulas are defined by the following grammar,

\[
\begin{align*}
P, Q &::= L \mid P \lor Q \mid P \ast Q \mid P \land Q \mid \top^+ \mid \top^- \mid \bot \mid A^+ \mid L \\
N, M &::= R \mid P \rightarrow N \mid P \rightarrow N \mid N \land M \mid \top^- \mid R \\
\end{align*}
\]

The set of positive formulas $P$ is denoted $P^+$; the set of negative formulas $N$ is denoted $P^-$; and the set of all polarised formulas is denoted $P^\pm$. The subclassifications $L$ and $R$ are left-neutral and right-neutral formulas respectively.

The shift operators have no logical meaning; they simply mediate the exchange of polarity, and thus the shifting into a new phase of proof-search. Consequently, to reduces cases in subsequent proofs, we will consider formulas of the form $\uparrow N$ and $\downarrow P$, but not $\uparrow \downarrow \downarrow N$ or $\uparrow \downarrow \downarrow P$.

**Definition 21 (Depolarization).** The depolarization function $\llbracket \rrbracket : F^\pm \rightarrow F$ is as follows. Let $\circ \in \{\lor, \ast, \rightarrow, \rightarrow^*\}$, and let $A \in P$, then depolarisation is defined as follows:

\[
\begin{align*}
[A^+] &:= [A^-] := A \\
[\uparrow \phi] &:= [\downarrow \phi] := [\phi] \\
[\downarrow] &:= \bot \\
[\top^+] &:= [\top^-] := \top \\
[\top^*] &:= [\top^-] := \top \quad [\phi \circ \psi] := [\varphi] \circ [\psi] \\
[\phi \land^+ \psi] &:= [\varphi \land \psi] := [\phi \land \psi]
\end{align*}
\]

Since proof-search is controlled by polarity, the construction of sequents in the focused system must be handled carefully to avoid ambiguity.

**Definition 22 (Polarised Sequents).** Positive contexts denoted by $\Gamma^+$, shifted by $\overline{\Gamma}$, and focused by $\hat{\Gamma}$, are defined according to the following grammars:

\[
\begin{align*}
\Gamma^+ &::= \Sigma^+ \mid \Pi^+ \\
\overline{\Gamma} &::= \overline{\Sigma} \mid \overline{\Pi} \\
\hat{\Gamma} &::= \widehat{\Sigma} \mid \widehat{\Pi} \\
\end{align*}
\]

\[
\begin{align*}
\Pi^+ &::= P \mid \{\Pi^+_1, \ldots, \Pi^+_n\}_+ \\
\overline{\Pi} &::= L \mid \{\overline{\Pi}^+_1, \ldots, \overline{\Pi}^+_n\}_+ \\
\hat{\Pi} &::= \langle N \rangle \mid \{\hat{\Pi}, \overline{\Pi}^+_1, \ldots, \overline{\Pi}^+_n\}_+ \\
\end{align*}
\]
A pair of a polarised nest and a polarised formula is a polarised sequent if it falls into one of the following cases:

\[ \Gamma^+ \Rightarrow N \mid \tilde{T} \Rightarrow \langle P \rangle \mid \tilde{T}\{\langle N \rangle\} \Rightarrow R \]

The decoration \( \langle \varphi \rangle \) indicates that the formula is in focus; i.e., it is a positive formula on the right, or a negative formula on the left. Of the three possible cases for well-formed polarised sequents, the first may be called *unfocused*, with the particular case of being *shifted* (neutral) when of the form \( \tilde{T} \Rightarrow R \); and the latter two may be called *focused*.

**Definition 23 (Depolarised Nest).** The depolarisation map extends to polarised nests \([ ] : \mathbb{B}/\equiv \Rightarrow \mathbb{B}/\equiv\) as follows:

\[
\begin{align*}
[\{\Pi_1, \ldots, \Pi_n\}]^+ &= \{[\Pi_1], \ldots, [\Pi_n]\}^+ \\
[\{\Sigma_1, \ldots, \Sigma_n\}] \times &= \{[\Sigma_1], \ldots, [\Sigma_n]\} \times
\end{align*}
\]

### 3.2 Focused Calculus

We may now give the focused system i.e., the operational semantics for focused proof-search in LBI. All the rules, with the exception of \( P \) and \( N \), are the polarised versions of the rules from \( \eta \text{LBI} \). The \( P \) and \( N \) rules are present for the simulation in the proof of the completeness theorem. In related works e.g., [7,6], the analogous rules are eliminated by initially working with a weaker notion of focused proof-search - it is reasonable to suppose that the same may be true for BI, but we leave this to future investigation.

**Definition 24 (System fBI).** The focused system fBI is composed of the rules on Figure 3.

Note the absence of a cut-rule, this is because the above system is intended to encapsulate precisely focused proof-search. Below we show that a cut-rule is indeed admissible, but proofs in fBI + cut do not necessarily correspond to focused proofs. Here the distinction between the methodologies for establishing the focusing principle becomes present since one may show completeness without leaving fBI by a permutation argument instead of a cut-elimination one.

Soundness follows immediately from the depolarisation map i.e., the interpretation of polarised sequents as nested sequents, and hence proofs in fBI actually are focused proofs in \( \eta \text{LBI} \).

**Theorem 25 (Soundness of fBI).** \( \vdash \text{fBI} \Gamma \Rightarrow \vdash \eta \text{LBI} [\Gamma] \)

*Proof.* Every rule in fBI except the shift rules, \( P \) axiom, and \( N \) axioms, become a rule in \( \eta \text{BI} \) when the antecedent(s) and consequent are depolarised. Instance of the shift rule can be ignored since the depolarised versions of the consequent and antecedents are the same. Finally, the depolarised versions \( P \) and \( N \) follow from Lemma 9 with the use of some weakening. \( \Box \)
Example 26. Consider the following proof in fBi, we suppose here that propositional letters A and C are negative, but B is positive.

\[
\text{(1)} \quad \Gamma \dashv\vdash (A \rightarrow B) \quad (A) \rightarrow A \\
\quad \text{and} \quad \Gamma \dashv\vdash (A \rightarrow B) \\
\quad \text{and} \quad \Gamma \dashv\vdash (A \rightarrow C) \\
\quad \text{and} \quad \Gamma \dashv\vdash (C \rightarrow C) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\top \rightarrow \bot) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\bot \rightarrow \top) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\bot \rightarrow \bot) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\top \rightarrow \top)
\]

\[
\text{(2)} \quad \Gamma \dashv\vdash (A \rightarrow B) \\
\quad \text{and} \quad \Gamma \dashv\vdash (A \rightarrow C) \\
\quad \text{and} \quad \Gamma \dashv\vdash (C \rightarrow C) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\top \rightarrow \bot) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\bot \rightarrow \top) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\bot \rightarrow \bot) \\
\quad \text{and} \quad \Gamma \dashv\vdash (\top \rightarrow \top)
\]
It is a witness of focused proof given in Example 19. Observe that the only non-deterministic choices are which formula to focus on e.g., steps (1) and (2) where different choices have been made for the sake of demonstration. The point of focusing is that only at such points do choices which affect termination occur. The assignment of polarity to the propositional letters is what forced the shape of the proof e.g., if $B$ has been negative the above would not have been well-formed. This phenomenon has been observed for other focused systems, e.g., in [8].

We now introduce the tool which will allow us to show that if there is a unstructured proof of a sequent, then there is necessarily a focused one.

**Definition 27.** All instances of the following rule where the sequents are well-formed are instances of cut, where $\varphi'$ denotes that $\varphi$ possibly has a shift:

$$
\frac{\Delta \Rightarrow \varphi \quad \Gamma(\varphi') \Rightarrow \chi}{\Gamma(\Delta) \Rightarrow \chi} \text{ cut}
$$

Admissibility follows from the usual argument, but within the focused system; i.e., through the upward permutation of cuts until they are eliminated in the axioms or are reduced in some other measure.

**Definition 28 (Good and Bad Cuts).** Let $D$ be a fBl + cut proof, a cut is a quadruple $\langle \mathcal{L}, \mathcal{R}, \mathcal{C}, \varphi \rangle$ where $\mathcal{L}$ and $\mathcal{R}$ are the premises to a cut concluding $\mathcal{C}$ in $D$, and $\varphi$ is the cut-formula. They are classified as follows:

**Good** - If $\varphi$ is principal in both $\mathcal{L}$ and $\mathcal{R}$.

**Bad** - If $\varphi$ is not principal in one of $\mathcal{L}$ and $\mathcal{R}$.

**Type 1:** If $\varphi$ is not principal in $\mathcal{L}$.

**Type 2:** If $\varphi$ is not principal in $\mathcal{R}$.

**Definition 29 (Cut Ordering).** The cut-complexity is the complexity of $\varphi$; the cut-level of a cut $\langle \mathcal{L}, \mathcal{R}, \mathcal{C}, \varphi \rangle$ in a proof $D$ the sum of the heights of the subproofs concluding $\mathcal{L}$ and $\mathcal{R}$; and, the cut-duplicity is the number of contraction above the cut. The cut-rank is the tuple $\langle \text{cut-complexity}, \text{cut-duplicity}, \text{cut-level} \rangle$.

Let $D$ and $D'$ be two fBl + cut proofs, let $\sigma$ and $\sigma'$ denote their multiset of cuts respectively. Proofs are ordered by $D \prec D' \iff \sigma < \sigma'$, where $<$ is the multiset ordering derived from the lexicographic ordering on cut-rank.

It follows from a result in [11] that the ordering on proofs is a well-order, since the ordering on cuts is a well-order.

**Lemma 30 (Good Cuts Elimination).** Let $D$ be a fBl + cut proof of $S$; there is a fBl + cut proof $D'$ of $S$ containing no good cuts such that $D' \preceq D$.

**Proof.** Let $D$ be as in hypothesis, if it contains no good cuts then $D = D'$ gives the desired proof. Otherwise, there is at least one good cut $\langle \mathcal{L}, \mathcal{R}, \mathcal{C}, \varphi \rangle$. Let $\partial$ be the sub-proof in $D$ concluding $\mathcal{C}$, then there is a transformation $\partial \mapsto \partial'$ where $\partial'$ is a fBl + cut proof of $S$ with $\partial' \prec \partial$ such that the set of good cuts in $\partial'$ is
smaller (with respect to $\prec$) than the set of good cuts in $\partial$. Since $\prec$ is a well-order indefinitely replacing $\partial$ with $\partial'$ in $\mathcal{D}$ for various cuts yields the desired $\mathcal{D}'$. The transformations are given in Appendix B, with some examples below. The key step is that a cut of a certain cut-complexity is replaced by cuts of lower cut complexity, possibly increasing the cut-duplicity or cut-level of other cuts in the proof.

\[
\frac{\{\Delta, \uparrow A_+\}_+ \Rightarrow \{A_+\}, \{\Delta', A_+\}_+ \Rightarrow \{A_+\}}{\{\Delta, \Delta', \uparrow A_+\}_+ \Rightarrow \{A_+\}}
\]

\[
\frac{\{\Delta'' \uparrow A_+\}_+ \Rightarrow \{A_+\}}{\{\Delta, \Delta', \Delta'' \uparrow A_+\}_+ \Rightarrow \{A_+\}}
\]

**Lemma 31 (Bad Cuts Elimination).** Let $\mathcal{D}$ be a fB! + cut proof of $S$ that contains only one cut which is bad, then there is a fB! + cut proof $\mathcal{D}'$ of $S$ such that $\mathcal{D}' \prec \mathcal{D}$.

**Proof.** Without loss of generality suppose the cut is the last inference in the proof, then it may be replaced by other cuts whose cut-level or cut-duplicity is smaller.

First we consider bad cuts when $\mathcal{L}$ and $\mathcal{R}$ are both axioms; there are no Type 1 bad cuts on axioms as the formula is always principle; the Type 2 bad cuts can trivially be permuted upwards or ignored e.g.,

\[
\frac{\Delta \Rightarrow \varphi, \{\Delta', \psi\}_x \Rightarrow \chi}{\Gamma\{\Delta, \Delta', \{\Delta'' \uparrow A_+\}_+\} \Rightarrow \chi}
\]

For the remaining cases the cuts are commutative in the sense that they may be permuted upward thereby reducing the cut-level. The cases are given in Appendix A, an example is given below:

\[
\frac{\Lambda\{\{\mathcal{N}_i\}\} \Rightarrow \mathcal{N}}{\mathcal{N}} \qquad \frac{\Lambda\{\{\mathcal{N}_i \land \mathcal{N}_j\}\} \Rightarrow \mathcal{N} \land \mathcal{L}}{\mathcal{N}} \qquad \frac{\Lambda\{\{\mathcal{N}_i\}\} \Rightarrow \mathcal{N} \land \mathcal{L}}{\mathcal{N}}
\]

\[
\frac{\Lambda\{\{\mathcal{N}_i \land \mathcal{N}_j\}\} \Rightarrow \mathcal{N} \land \mathcal{L}}{\mathcal{N} \land \mathcal{L}} \qquad \frac{\Lambda\{\{\mathcal{N}_i\}\} \Rightarrow \mathcal{N} \land \mathcal{L}}{\mathcal{N} \land \mathcal{L}}
\]
The exceptional cases are the interaction with contraction where the cut is replaced by cuts of possibly equal cut-level, but cut-duplicity decreases:

\[
\Delta' \Rightarrow N \quad \Rightarrow \quad \Gamma((\delta N), \delta N) \Rightarrow \chi \\
\Gamma((\delta N), \delta N) \Rightarrow \chi \\
\Delta' \Rightarrow N \quad \Rightarrow \quad \Gamma((\delta \Delta'), \delta \Delta') \Rightarrow \chi \\
\Gamma((\delta \Delta'), \delta \Delta') \Rightarrow \chi \\
\Gamma((\Delta'), \Delta') \Rightarrow \chi 
\]

\[
\Delta' \Rightarrow N \quad \Rightarrow \quad \Gamma((\delta \Delta'), \delta \Delta') \Rightarrow \chi \\
\Gamma((\Delta'), \Delta') \Rightarrow \chi 
\]

**Theorem 32 (Cut-elimination in fBI).** \(\vdash_{\text{fBI}} \Gamma \Rightarrow \varphi \) if and only if \(\vdash_{\text{fBI+cut}} \Gamma \Rightarrow \varphi \)

**Proof.** (\(\Rightarrow\)) Trivial as any fBl-proof is a fBl+cut-proof. (\(\Leftarrow\)) Let \(D\) be a fBl+cut-proof of \(\varphi\), if it has no cuts then it is a fBl-proof so we are done. Otherwise, there is at least one cut, and we proceed by well-founded induction on the ordering of proofs and sub-proofs of \(D\) with respect to \(\prec\).

**Base Case.** Assume \(D\) is minimal with respect to \(\prec\) with at least one cut; without loss of generality, by Lemma 30, assume the cut is bad. It follows from Lemma 31 that there is a proof strictly smaller in \(\prec\)-ordering, but this proof must be cut-free as \(D\) is minimal.

**Inductive Step.** Let \(D\) be as in the hypothesis, then by Lemma 30 there is a proof \(\partial\) of \(\varphi\) containing no good cuts such that \(\partial \preceq D\). Either \(\partial\) is cut-free and we are done, or it contains bad cuts. Consider the topmost cut, and denote the sub-proof by \(\partial\), it follows from Lemma 31 that there is a proof \(\partial'\) of of the same sequent such that \(\partial' \prec \partial\). Hence, by inductive hypothesis, there is a cut-free proof the sequent and replacing \(\partial\) by this proof in \(D\) gives a proof of 

\(\Gamma \Rightarrow \varphi\) strictly smaller in \(\prec\)-ordering, thus by inductive hypothesis there is a cut-free proof as required.

\(\Box\)

### 3.3 Completeness of fBI

The completeness theorem of the focused system i.e., the operational semantics, is with respect to an interpretation i.e., a polarisation. Indeed, any polarisation may be considered e.g., both \((\downarrow A \ast B_+) \land^+ \downarrow A\) and \((A_+ \ast \downarrow B\) \land^+ A_+ are valid polarisations of the formulas \((A \ast B) \land A\). Taking arbitrary \(\varphi\) the process is as follows. First, fix a polarised syntax i.e., a partition of the propositional letters into positive and negative sets:

- If \(\varphi\) is a propositional atom, it must be polarised by default;
- If \(\varphi = \top\), then choose polarisation \(\top^+\) or \(\top^-\);
– If \( \varphi = \psi_1 \land \psi_2 \), first polarise \( \psi_1 \) and \( \psi_2 \), then choose an additive conjunction and combine accordingly, using shifts to ensure the formula is well-formed;
– If \( \varphi = \psi_1 \circ \psi_2 \) where \( \circ \in \{*, \circ, \to\} \), then polarise \( \psi_1 \) and \( \psi_2 \) and combine with \( \circ \) accordingly, using shifts where necessary.

**Example 33.** Suppose \( A \) is negative and \( B \) is positive, then \( (A \ast B) \land A \) may be polarised by choosing the additive conjunction to be positive resulting in \( (\downarrow A \ast B) \land^+ \downarrow A \) (when \( \downarrow (A \ast \downarrow B) \land^+ A \) would not be well-formed). Choosing to shift one can ascribe a negative polarisation \( \uparrow (\downarrow A \ast B) \land^+ \downarrow A \).

The above generates the set of all such polarised formulas when all possible choices are explored. The free assignment of polarity to formulas means numerous focusing procedures are captured by the completeness theorem.

**Lemma 34 (Completeness of \( fBI + \text{cut} \)).** For any unfocused \( \Gamma \Rightarrow \psi \), if \( \vdash_{\eta LBI} [\Gamma \Rightarrow N] \) then \( \vdash_{fBI + \text{cut}} \Gamma \Rightarrow N \).

*Proof.* We show that every rule in \( fBI \) is admissible in \( \eta LBI \). Consequently, the remaining cases are given in Appendix C; below we give an examples on how to simulate a positive and negative rule. Where it does not matter e.g., for inactive nests, we do not distinguish the polarised and unpolarised versions. Note that each of the simulations can be concluded due to the presence of the \( P \) and \( N \) axioms.

\[
\frac{\Gamma \Rightarrow \psi}{\{\Gamma, \Delta\}_{x}, \Delta'} \Rightarrow \psi^* \quad \text{is simulated by} \quad \frac{\Gamma \Rightarrow \psi^*}{\{\Gamma, \Delta\}_{x}} \Rightarrow \psi \quad \text{simulated by} \quad \frac{\Delta \Rightarrow \psi}{\Delta', \varphi \Rightarrow \psi} \Rightarrow \chi \quad \text{simulated by} \quad \frac{\downarrow\psi^+ \Rightarrow \varphi^-}{\downarrow\varphi^- \Rightarrow \uparrow\psi^-} \Rightarrow \chi \quad \text{simulated by} \quad \frac{\Delta \Rightarrow \varphi^+}{\Delta', \varphi^- \Rightarrow \chi} \Rightarrow \chi \quad \text{simulated by} \quad \frac{\Delta \Rightarrow \varphi^+}{\Delta', \varphi^- \Rightarrow \chi} \Rightarrow \chi \quad \text{simulated by} \quad \frac{\Delta \Rightarrow \varphi^+}{\Delta', \varphi^- \Rightarrow \chi} \Rightarrow \chi \quad \text{simulated by} \quad \frac{\Delta \Rightarrow \varphi^+}{\Delta', \varphi^- \Rightarrow \chi} \Rightarrow \chi \quad \text{simulated by} \quad \frac{\Delta \Rightarrow \varphi^+}{\Delta', \varphi^- \Rightarrow \chi} \Rightarrow \chi
\]

\[\square\]

**Theorem 35 (Completeness of \( fBI \)).** For any unfocused \( \Gamma \Rightarrow \psi \), if \( \vdash_{\eta LBI} [\Gamma \Rightarrow \psi] \) then \( \vdash_{fBI} \Gamma \Rightarrow \psi \).
Proof. It follows from Lemma 34 that there is a proof of $\Gamma \Rightarrow \varphi$ in fBI + cut, and then it follows from Lemma 32 that there is a proof of $\Gamma \Rightarrow \varphi$ in fBI, as required.

Given an arbitrary sequent the above theorem guarantees the existence of a focused proof, thus the focusing principle holds for $\eta\text{LBI}$ and therefore for LBI.

4 Conclusion

By proving the completeness of a focused sequent calculus for the logic of Bunched Implications, we have demonstrated that it satisfies the focusing principle; that is, any polarisation of a BI-provable sequent can be proved following a focused search procedure. This required a careful analysis of how to restrict the usage of structural rules. In particular, we had to fully develop the congruence-invariant representation of bunches as nested multisets (originally proposed in [13]) to treat the exchange rule within bunched structures.

Proof-theoretically the completeness of the focused systems suggests a syntactic naturality of LBI, though the $P$ and $N$ axioms leave something to be desired. Computationally, these axioms are unproblematic as during search it makes sense to terminate a branch as soon as possible; however, unless they may be eliminated it means that the focusing principle holds in BI only up to a point. Restricting attention to the reductive logic of BI, it may be that focused proof-search can be used to implement efficient and effective proof-search procedures. In related works e.g., [7] the analogous problem is overcome by first considering a weak focused system e.g., one where the structural rules are not controlled and safe-reduction may be performed inside focused phases if desired. Completeness of (strong) focusing is achieved by appealing to a synthetic system. It seems reasonable to suppose the same can be done for BI, resulting in a more proof-theoretically satisfactory focused calculus, exploring this possibility is a natural extension of the work on fBI.

The methodology employed for proving the focusing principle can be interpreted as soundness and completeness of an operational semantics for goal-directed search. The robustness of this technique is demonstrated by its efficacy in modal [7,6] and substructural logics [22], including now bunched ones. Although BI may be the most employed bunched logic, there are a number of others e.g., relevant logics [31] and the family of bunched logics [12], for which the focusing principle should be studied. However, without the presence of a cut-free sequent calculus goal-directed search becomes unclear, and currently such calculi do not exist for the two main variants of BI i.e., Boolean BI [29] and Classical BI [5]. On the other hand, large families of bunched and substructural logics have been given hypersequent calculi [9,10]. Effective proof-search procedures have been established for the hypersequent calculi in the substructural case [30], but not the bunched one, and focused proof-search for neither. There is a technical challenge in focusing these systems as one must not only decide which formula to reduce, but also which sequent.
In the future it will be especially interesting to see how focused search, when combined with the expressiveness of BI, increases its modelling capabilities. Indeed, the dynamics of proof-search can be used to represent models of computation within (propositional) logics e.g., the undecidability of Linear Logic involves simulating two-counter machines [22]. With respect to focused proof-search in particular, focused systems have been used to emulate proofs for other logics [23]; and to give structural operational semantics for systems used in industry e.g., algorithms for solving constraint satisfaction problems [15].

References

1. Jean-marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2:297–347, 1992.
2. Pablo Armein. *Bunched Logic Programming*. PhD thesis, Queen Mary College, University of London, 2002.
3. Josh Berdine, Cristiano Calcagno, and Peter W. O’Hearn. Smallfoot: Modular automatic assertion checking with separation logic. In *Proceedings of the 4th International Conference on Formal Methods for Components and Objects*, FMCO’05, page 115–137, Berlin, Heidelberg, 2005. Springer-Verlag.
4. James Brotherston. A unified display proof theory for bunched logic. *Electronic Notes in Theoretical Computer Science*, 265:197 – 211, 2010. Proceedings of the 26th Conference on the Mathematical Foundations of Programming Semantics (MFPS 2010).
5. James Brotherston and Cristiano Calcagno. Classical bi: Its semantics and proof theory. *Logical Methods in Computer Science*, 6, 05 2010.
6. Kaustuv Chaudhuri, Sonia Marin, and Lutz Straßburger. Focused and synthetic nested sequents. In *Foundations of Software Science and Computation*, volume 9634, pages 390–407, 04 2016.
7. Kaustuv Chaudhuri, Sonia Marin, and Lutz Straßburger. Modular focused proof systems for intuitionistic modal logics. In *FSCD 2016 - 1st International Conference on Formal Structures for Computation and Deduction*, 2016.
8. Kaustuv Chaudhuri, Frank Pfenning, and Greg Price. A logical characterization of forward and backward chaining in the inverse method. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, pages 97–111, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
9. Agata Ciabattoni, Nikolaos Galatos, and Kazushige Terui. Algebraic proof theory for substructural logics: Cut-elimination and completions. *Annals of Pure and Applied Logic*, 163(3):266–290, Mar 2012.
10. Agata Ciabattoni and Revantha Ramanayake. Bunched hypersequent calculi for distributive substructural logics. In Thomas Eiter and David Sands, editors, *LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 46 of *EPiC Series in Computing*, pages 417–434. EasyChair, 2017.
11. Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. In Hermann A. Maurer, editor, *Automata, Languages and Programming*, pages 188–202, Berlin, Heidelberg, 1979. Springer Berlin Heidelberg.
12. Simon Docherty. Bunched logics: A uniform approach.
13. Kevin Donnelly, Tyler Gibson, Neel Krishnaswami, Stephen Magill, and Sungwoo Park. The inverse method for the logic of bunched implications. In Franz Baader and Andrei Voronkov, editors, Logic for Programming, Artificial Intelligence, and Reasoning, pages 466–480, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
14. Roy Dyckhoff and Stéphane Lengrand. Ljq: A strongly focused calculus for intuitionistic logic. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, Logical Approaches to Computational Barriers, pages 173–185, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
15. Mahfuza Farooque, Stéphane Graham-Lengrand, and Assia Mahboubi. A bisimulation between dpill(t) and a proof-search strategy for the focused sequent calculus. LFMTP ’13, page 3–14, New York, NY, USA, 2013. Association for Computing Machinery.
16. D. Galmiche, D. Mery, and D. J. Pym. The semantics of bi and resource tableaux. Mathematical Structures in Computer Science, 15(6):1033–1088, 2005.
17. Gerhard Gentzen. The Collected Papers of Gerhard Gentzen. Amsterdam: North-Holland Pub. Co., 1969.
18. Jean-Yves Girard. Linear logic. Theoretical Computer Science, 50(1):1 – 101, 1987.
19. ZHÉ HOU, RAJEEV GARÉ, and ALINE TIU. A labelled sequent calculus for BBI: proof theory and proof search. Journal of Logic and Computation, 28(4):809–872, 06 2015.
20. Olivier Laurent. A proof of the focalization property of linear logic. https://perso.ens-lyon.fr/olivier.laurent/llfoc.pdf, 04 2004.
21. Chuck Liang and Dale Miller. Focusing and polarization in linear, intuitionistic, and classical logics. Journal of Theoretical Computer Science, 410(46):4747–4768, November 2009.
22. Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. Annals of Pure and Applied Logic, 56(1):239 – 311, 1992.
23. Sonia Marin, Dale Miller, and Marco Volpe. A focused framework for emulating modal proof systems. Number 11 in Advances in Modal Logic, pages 469–488, 2016.
24. Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. Annals of Pure and Applied Logic, 51(1):125 – 157, 1991.
25. Dale Miller and Elaine Pimentel. A formal framework for specifying sequent calculus proof systems. Theoretical Computer Science, 474:98–116, February 2013.
26. Peter O’Hearn and David Pym. The logic of bunched implications. The Bulletin of Symbolic Logic, 5(2):215–244, 1999.
27. Jonghyun Park, Jeongbong Seo, and Sungwoo Park. A theorem prover for boolean bi. In Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL ’13, page 219–232, New York, NY, USA, 2013. Association for Computing Machinery.
28. David Pym. Resource semantics: Logic as a modelling technology. ACM SIGLOG News, 6(2):5–41, April 2019.
29. David J. Pym. The Semantics and Proof Theory of the Logic of Bunched Implications, volume 26 of Applied Logic Series. Springer Netherlands, Dordrecht, 2002.
30. Revantha Ramanayake. Extended kripke lemma and decidability for hypersequent substructural logics. LICS ’20, page 795–806, New York, NY, USA, 2020. Association for Computing Machinery.
31. S. Read. *Relevant Logic: A Philosophical Examination of Inference*. B. Blackwell, 1988.

32. J. C. Reynolds. Separation logic: a logic for shared mutable data structures. In *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, pages 55–74, 2002.
A  Bad Cuts

We have not included the cases of cutting with a shift rule since the transformation is simply to not perform the shift.

Type 1:

\[
\frac{\Delta(P, Q) \vdash \varphi \quad \Delta(P) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P, Q) \vdash \varphi} \quad \text{cut} \quad \frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \lor Q) \vdash \varphi} \quad \text{cut}
\]

\[
\frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \land Q) \vdash \varphi} \quad \text{cut} \quad \frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi \quad \Delta(Q') \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \land Q) \vdash \varphi} \quad \text{cut}
\]

\[
\frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \lor Q) \vdash \varphi} \quad \text{cut} \quad \frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \land Q) \vdash \varphi} \quad \text{cut}
\]

\[
\frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \land Q) \vdash \varphi} \quad \text{cut} \quad \frac{\Delta(P) \vdash \varphi \quad \Delta(Q) \vdash \varphi}{\Gamma \vdash \varphi \quad \Delta(P \lor Q) \vdash \varphi} \quad \text{cut}
\]
Type 2:

\[
\frac{\Delta \vdash \phi}{\overline{\Gamma}(\Delta) \Rightarrow (P_1 \lor P_2)} \quad \text{cut} \quad \iff \quad \frac{\Delta \vdash \phi}{\overline{\Gamma}(\Delta) \Rightarrow (P_1 \lor P_2)} \quad \text{cut}
\]

\[
\frac{\Delta \vdash \phi}{\overline{\Gamma}(\Delta), (N_1 \land N_2) \Rightarrow N} \quad \text{cut} \quad \iff \quad \frac{\Delta \vdash \phi}{\overline{\Gamma}(\Delta), (N_1 \land N_2) \Rightarrow N} \quad \text{cut}
\]

\[
\overline{\Delta} \vdash \phi \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P)} \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (Q)} \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P \times Q)} \quad \text{cut} \quad \implies \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P \times Q)} \quad \text{cut}
\]

\[
\overline{\Delta} \vdash \phi \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P) \times (P \times Q)} \quad \text{cut} \quad \implies \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P \times Q)} \quad \text{cut}
\]

\[
\overline{\Delta} \vdash \phi \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P)} \quad \text{cut} \quad \implies \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P \times Q)} \quad \text{cut}
\]

\[
\overline{\Delta} \vdash \phi \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P) \times (P \times Q)} \quad \text{cut} \quad \implies \quad \frac{(\overline{\Delta}(\varphi), \overline{\Gamma})}{\Rightarrow (P \times Q)} \quad \text{cut}
\]

\[
\overline{\Delta} \vdash \phi \quad \frac{\overline{\Gamma}(\Delta, (N))}{\Rightarrow M} \quad \text{cut} \quad \iff \quad \frac{\overline{\Gamma}(\Delta, (N))}{\Rightarrow M} \quad \text{cut}
\]

\[
\overline{\Delta} \vdash \phi \quad \frac{\overline{\Gamma}(\Delta, (N))}{\Rightarrow M} \quad \text{cut} \quad \iff \quad \frac{\overline{\Gamma}(\Delta, (N))}{\Rightarrow M} \quad \text{cut}
\]
B Good Cuts

\[
\frac{\{\overline{2}, \uparrow A_{+}\} \Rightarrow (A_{+}) \quad \{\overline{2}', A_{+}\} \Rightarrow (A_{+})}{(\overline{2}, \overline{2}', \uparrow A_{+} \downarrow) \Rightarrow (A_{+})} \quad \mapsto \quad \frac{\{\overline{2}, \overline{2}', \uparrow A_{+}\} \Rightarrow (A_{+})}{(\overline{2}, \overline{2}', \uparrow A_{+} \downarrow) \Rightarrow (A_{+})}
\]

\[
\frac{\{\overline{2}, (A_{-})\} \Rightarrow A_{-} \quad \{\overline{2}', (A_{-})\} \Rightarrow A_{-}}{(\overline{2}, \overline{2}', A_{-}) \Rightarrow A_{-}} \quad \mapsto \quad \frac{\{\overline{2}, \overline{2}', (A_{-})\} \Rightarrow A_{-}}{(\overline{2}, \overline{2}', A_{-}) \Rightarrow A_{-}}
\]

\[
\frac{\{\overline{2}, A_{+}\} \Rightarrow (A_{+}) \quad \{\overline{2}', A_{+}\} \Rightarrow (A_{+})}{(\overline{2}, \overline{2}', A_{+}) \Rightarrow (A_{+})} \quad \mapsto \quad \frac{\{\overline{2}, \overline{2}', A_{+}\} \Rightarrow (A_{+})}{(\overline{2}, \overline{2}', A_{+}) \Rightarrow (A_{+})}
\]

\[
\frac{\{\overline{2}, (A_{-})\} \Rightarrow A_{-} \quad \{\overline{2}', (A_{-})\} \Rightarrow A_{-}}{(\overline{2}, \overline{2}', A_{-}) \Rightarrow A_{-}} \quad \mapsto \quad \frac{\{\overline{2}, \overline{2}', A_{-}\} \Rightarrow A_{-}}{(\overline{2}, \overline{2}', A_{-}) \Rightarrow A_{-}}
\]

\[
\frac{\overline{2} \Rightarrow \top \quad \{\overline{2}, (\top)\} \Rightarrow \top}{(\overline{2}, \overline{2}, \top) \Rightarrow \top} \quad \mapsto \quad \frac{\overline{2} \Rightarrow \top \quad \{\overline{2}, (\top)\} \Rightarrow \top}{(\overline{2}, \overline{2}, \top) \Rightarrow \top}
\]

\[
\frac{\overline{2} \Rightarrow \top \quad \{\overline{2}, (\top)\} \Rightarrow \top}{(\overline{2}, \overline{2}, \top) \Rightarrow \top} \quad \mapsto \quad \frac{\overline{2} \Rightarrow \top \quad \{\overline{2}, (\top)\} \Rightarrow \top}{(\overline{2}, \overline{2}, \top) \Rightarrow \top}
\]

\[
\frac{\{\overline{2}, (A_{-})\} \Rightarrow A_{-} \quad \{\overline{2}', (A_{-})\} \Rightarrow A_{-}}{(\overline{2}, \overline{2}', A_{-}) \Rightarrow A_{-}} \quad \mapsto \quad \frac{\{\overline{2}, \overline{2}', A_{-}\} \Rightarrow A_{-}}{(\overline{2}, \overline{2}', A_{-}) \Rightarrow A_{-}}
\]

\[
\frac{\overline{3}, \overline{3}', P_{+} \Rightarrow N \quad \xrightarrow{R} \quad \overline{3}' = (P) \quad \Gamma(\overline{3}, N)_{x} \Rightarrow M \quad \mapsto \quad \frac{\overline{3}, \overline{3}', (P \rightarrow N)_{x} \Rightarrow M}{\Gamma(\overline{3}, \overline{3}', P_{+} \rightarrow N)_{x} \Rightarrow M} \quad \text{cut}}
\]

\[
\frac{\overline{3}, \overline{3}', P_{+} \Rightarrow N \quad \xrightarrow{R} \quad \overline{3}' = (P) \quad \Gamma(\overline{3}, N)_{x} \Rightarrow M \quad \mapsto \quad \frac{\overline{3}, \overline{3}', (P \rightarrow N)_{x} \Rightarrow M}{\Gamma(\overline{3}, \overline{3}', P_{+} \rightarrow N)_{x} \Rightarrow M} \quad \text{cut}}
\]

\[
\frac{\overline{3}, \overline{3}', P_{+} \Rightarrow N \quad \xrightarrow{R} \quad \overline{3}' = (P) \quad \Gamma(\overline{3}, N)_{x} \Rightarrow M \quad \mapsto \quad \frac{\overline{3}, \overline{3}', (P \rightarrow N)_{x} \Rightarrow M}{\Gamma(\overline{3}, \overline{3}', P_{+} \rightarrow N)_{x} \Rightarrow M} \quad \text{cut}}
\]

\[
\frac{\overline{3}, \overline{3}', P_{+} \Rightarrow N \quad \xrightarrow{R} \quad \overline{3}' = (P) \quad \Gamma(\overline{3}, N)_{x} \Rightarrow M \quad \mapsto \quad \frac{\overline{3}, \overline{3}', (P \rightarrow N)_{x} \Rightarrow M}{\Gamma(\overline{3}, \overline{3}', P_{+} \rightarrow N)_{x} \Rightarrow M} \quad \text{cut}}
\]
\[
\Delta \Rightarrow P \\
\Delta, \Delta' \Rightarrow P \rightarrow Q \quad \ast R \\
\Gamma\{P, Q\} \Rightarrow N \quad \Gamma\{P \rightarrow Q\} \Rightarrow N \quad \ast L \\
\text{cut} \\
\Delta' \Rightarrow Q \\
\Gamma\{\Delta, \Delta'\} \Rightarrow N \quad \text{cut}
\]

\[
\Rightarrow \\
\Delta \Rightarrow N_1 \\
\Delta \Rightarrow N_2 \quad \ast R \\
\Gamma\{(N_1)\} \Rightarrow M \\
\Gamma\{(N_1 \land N_2)\} \Rightarrow M \\
\text{cut} \\
\Gamma\{\Delta\} \Rightarrow M
\]

\[
\Rightarrow \\
\overline{\Delta} \Rightarrow (P_1 \lor P_2) \quad \ast R \\
\overline{\Gamma}\{P_1\} \Rightarrow N \\
\overline{\Gamma}\{P_2\} \Rightarrow N \\
\text{cut} \\
\overline{\Delta} \Rightarrow (P_1 \lor P_2)
\]

\[
\Rightarrow \\
\Delta \Rightarrow (P) \\
\Gamma\{P, P\} \Rightarrow N \\
\text{cut} \\
\Delta \Rightarrow (N) \\
\Gamma\{N, (N)\} \Rightarrow M \\
\text{cut} \\
\Delta \Rightarrow (N) \\
\Gamma\{N, (N)\} \Rightarrow M \\
\text{cut}
\]

\[
\Delta \Rightarrow (P) \\
\Gamma\{P, P\} \Rightarrow N \\
\text{cut} \\
\Delta \Rightarrow (N) \\
\Gamma\{N, (N)\} \Rightarrow M \\
\text{cut}
\]
C  Simulation

\[
\Delta \Rightarrow \varphi^+ \quad \Gamma[\Delta', \varphi^+ \rightarrow \psi^-]_x \Rightarrow \chi \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\downarrow \varphi^+ \Rightarrow \varphi^+ \quad \downarrow \psi^- \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\{\varphi^+ \rightarrow \psi^-, \downarrow \varphi^+\}_x \Rightarrow \downarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\downarrow \varphi^+ \Rightarrow \varphi^+ \quad \downarrow \psi^- \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]

\[
\Gamma \Rightarrow \varphi^+ \quad \Delta \Rightarrow \psi^- \quad \Rightarrow L \quad \text{simulated by}
\]
\[
\frac{\Gamma \Rightarrow \varphi \quad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \land \psi} \quad \text{simulated by}
\]

\[
\frac{\Gamma \Rightarrow \varphi^+ \quad \Gamma \Rightarrow \psi^+}{\Gamma \Rightarrow \varphi^+ \land \psi^+} \\
\frac{(\Gamma, \Gamma)_+ \Rightarrow \varphi^+ \land \psi^+}{(\Gamma, \Gamma)_+ \Rightarrow (\varphi^+ \land \psi^+)}
\]

\[
\frac{\Gamma \Rightarrow \varphi^+ \land \psi^+}{\Gamma \Rightarrow \varphi^+ \land \psi^+} \\
\frac{(\Gamma, \Gamma)_- \Rightarrow \varphi^+ \land \psi^+}{(\Gamma, \Gamma)_- \Rightarrow (\varphi^+ \land \psi^+)}
\]

\[
\frac{\Gamma \Rightarrow \varphi^+ \land \psi^+}{\Gamma \Rightarrow \varphi^+ \land \psi^+}
\]