Connection-based proof search in propositional BI logic

Didier Galmiche and Daniel Méry

LORIA - Université Henri Poincaré
Campus Scientifique, BP 239
Vandœuvre-lès-Nancy, France

Abstract. We present a connection-based characterization of propositional **BI** (logic of bunched implications), a logic combining linear and intuitionistic connectives. This logic, with its sharing interpretation, has been recently used to reason about mutable data structures and needs proof search methods. Our connection-based characterization for **BI** is based on standard notions but involves, in a specific way, labels and constraints in order to capture the interactions between connectives during the proof-search. As **BI** is conservative w.r.t. intuitionistic logic and multiplicative intuitionistic linear logic, we deduce, by some restrictions, new connection-based characterizations and methods for both logics.

1 Introduction

The logic of bunched implications (**BI**) provides a logical analysis of the basic notion of resource, that is central in computer science, with well-defined proof-theoretic and semantic foundations [10,12]. Its propositional fragment freely combines multiplicative (or linear) $*$ and $-\!\!*$ connectives and additive (or intuitionistic) $\wedge$, $\rightarrow$ and $\vee$ connectives [11] and can be seen as a merging of intuitionistic logic (**IL**) and multiplicative intuitionistic linear logic (**MILL**). **BI** has a Kripke-style semantics (interpretation of formulae) [11] which combines the Kripke semantics of **IL** and Urquhart's semantics of **MILL**. The latter uses possible worlds, arranged as a commutative monoid and justified in terms of "pieces of information" [14]. The key property of the semantics is the *sharing interpretation*. The (elementary) semantics of the multiplicative conjunction, $m \models A * B$ iff there are n_1 and n_2 such that $n_1 \bullet n_2 \sqsubseteq m$, $n_1 \models A$ and $n_2 \models B$, is interpreted as follows: the resource m is sufficient to support $A * B$ just in case it can be divided into resources n_1 and n_2 such that n_1 is sufficient to support A and n_2 is sufficient to support B. Thus, A and B do not share resources. Similarly, the semantics of the multiplicative implication, $m \models A -\!\!* B$ iff for all n such that $n \models A$, $m \bullet n \models B$, is interpreted as follows: the resource m is sufficient to support $A -\!\!* B$ just in case for any resource n which is sufficient to support A the combination $m \bullet n$ is sufficient to support B. Thus, the function and its argument *do not share* resources. In contrast, if we consider the standard Kripke semantics of the additives $\wedge$ and $\rightarrow$ the resources are shared. Because of the interaction

of intuitionistic and linear connectives and its sharing interpretation, **BI** is different from linear logic (LL) [11] and does not admit the usual number-of-uses reading. Proof-search methods dedicated to **IL** or multiplicative LL (**MLL**) introduced the notion of prefix in order to capture the order of rule applications from which a proof in the associated sequent calculus can be constructed [9,16]. Such an approach cannot be adapted to **BI** that needs more general notions, in order to capture the actual interactions between multiplicative and additive connectives. It appears that the introduction of labels and constraints is a natural way to do it. Recently, we have proposed a particular tableau method for propositional **BI** (without inconsistency) [6] with label and constraints attached to the expansion rules. This first proof search method for **BI** includes intrinsic and specific redundancies and their related problems. Our aim in this paper is to propose a connection (or matrix) characterization of provability in **BI** and a related connection-based proof method, knowing that connection methods drastically reduce the search space compared to calculi analyzing the outer structure of formulae such as sequent or tableau calculi [3,16].
In this setting, in addition to the standard notions, we introduce specific labels and constraints that in fact allow to capture the semantic consequence relation of **BI**. Such a constraint-based approach would provide powerful methods for substructural logics mixing different kinds of connectives, as in **BI** with intuitionistic and linear connectives. With connection methods based on labels and constraints, we propose an appropriate and natural way to capture the essence of the provability in **BI**, for which based-on prefixes methods, defined for instance in **IL** or **MLL** [9], cannot be easily extended. As **BI** is a new logical foundation for logic programming [1], interference in imperative programs [10] and reasoning about mutable data structures using pointers [7], our results could be fruitful in these settings. Moreover, as **BI** is conservative w.r.t. **IL** and **MILL**, the restriction and specialization of our results to both logics, provide new connection-based characterizations and proof methods for **IL** and **MILL**. The latter is strongly related to proof nets construction in **MILL** [2]

2 The logic of bunched implications (BI)

The propositional language of **BI** consists of a multiplicative unit I, the multiplicative connectives $*$, $-\!*$, the additive units $\top$, $\bot$, the additive connectives $\wedge$, $\rightarrow$, $\vee$, a countable set $L = p, q, \ldots$ of propositional letters. $\mathcal{P}(L)$, the collection of **BI** propositions over L, is given by the following inductive definition: $A ::= p \mid I \mid A * A \mid A -\!* A \mid \top \mid \bot \mid A \wedge A \mid A \rightarrow A \mid A \vee A$.
Judgements of **BI** are sequents of the form $\Gamma \vdash A$, where A is a proposition and Γ is what is called a "bunch", more precisely, a tree (rather than a list or a multiset) having propositions as leaves and whose internal nodes are labelled with ";" or ",". Therefore, we have two context-forming operations, namely, ";" and "," the key point being that ";" admits the structural rules of contraction and weakening while "," does not. More formally, bunches are given by the following grammar $\Gamma ::= A \mid \emptyset_m \mid \Gamma, \Gamma \mid \emptyset_a \mid \Gamma ; \Gamma$.

– Identity and structures

$$\frac{}{A \vdash A}\, ax \qquad \frac{\Gamma \vdash A}{\Delta \vdash A}\, \Delta \equiv \Gamma \qquad \frac{\Gamma(\Delta) \vdash A}{\Gamma(\Delta; \Delta') \vdash A}\, w \qquad \frac{\Gamma(\Delta; \Delta) \vdash A}{\Gamma(\Delta) \vdash A}\, c$$

– Unities

$$\frac{}{\bot \vdash A}\, \bot_L \qquad \frac{\Gamma(\emptyset_m) \vdash A}{\Gamma(I) \vdash A}\, I_L \qquad \frac{}{\emptyset_m \vdash I}\, I_R \qquad \frac{\Gamma(\emptyset_a) \vdash A}{\Gamma(\top) \vdash A}\, \top_L \qquad \frac{}{\emptyset_a \vdash \top}\, \top_R$$

– Multiplicatives

$$\frac{\Gamma(A, B) \vdash C}{\Gamma(A * B) \vdash C}\, *_L \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A * B}\, *_R \quad \frac{\Gamma \vdash A \quad \Delta(\Delta', B) \vdash C}{\Delta(\Delta', \Gamma, A \mathrel{-\!\!*} B) \vdash C}\, \mathrel{-\!\!*}_L \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \mathrel{-\!\!*} B}\, \mathrel{-\!\!*}_R$$

– Additives

$$\frac{\Gamma(A; B) \vdash C}{\Gamma(A \wedge B) \vdash C}\, \wedge_L \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma; \Delta \vdash A \wedge B}\, \wedge_R \qquad \frac{\Gamma \vdash A \quad \Delta(\Delta'; B) \vdash C}{\Delta(\Delta'; \Gamma; A \to B) \vdash C}\, \to_L$$

$$\frac{\Gamma; A \vdash B}{\Gamma \vdash A \to B}\, \to_R \qquad \frac{\Gamma(A) \vdash C \quad \Delta(B) \vdash C}{\Gamma(A \vee B); \Delta(A \vee B) \vdash C}\, \vee_L \qquad \frac{\Gamma \vdash A_{i\,(i=1,2)}}{\Gamma \vdash A_1 \vee A_2}\, \vee_{Ri}$$

Fig. 1. The **LBI** sequent calculus

We write $\Gamma(\Delta)$ to represent a bunch in which Δ is a subtree. Since bunches may be represented in various way, we consider them up to a structural equivalence ($\equiv$) defined as the least equivalence satisfying commutative monoid equations for $\emptyset_a$ and ";" and for $\emptyset_m$ and "," together with congruence: $\Delta \equiv \Delta'$ implies $\Gamma(\Delta) \equiv \Gamma(\Delta')$.

The **LBI** sequent calculus is given in figure 1. A proposition A is a theorem iff $\emptyset_m \vdash A$ is provable in **LBI**. Let us mention that an elementary Kripke semantics has been defined for **BI** but the completeness result only holds for **BI** without $\bot$. Some topological semantics have been proposed in order to have completeness for full propositional **BI** [11].
Proof-search in **BI** is difficult because of the interaction between additive and multiplicative connectives which, in the **LBI** calculus, results in sequents whose contexts are structured as bunches. Due to the properties of the context building operations, bunches admit various structural transformations which are delicate to handle, but on the other hand, often required during the proof-search process in order to achieve a proof. In order to alleviate each type of redundancies induced by proof-search in the **LBI** calculus, namely: notational redundancy, irrelevance and non-permutability [16], we propose a connection-based characterization of **BI** provability and a related proof-search method.

3 A connection-based characterization of BI

The matrix (or connection) characterization of **BI** provability we propose is based on *labels* and *constraints* that are the appropriate concepts that allow to capture the consequence relation, and thus the interaction of connectives at a semantical level. Semantically speaking, the interaction between additive and multiplicative connectives can mainly be seen as a problem of sharing and distributing the resources and based-on prefixes methods, defined for instance in **IL** or **MLL** [9], cannot naturally deal with it.

3.1 Labels and constraints

Given an alphabet C (for example $a, b, c, \cdots$), C^0, the set of *atomic labels over* C, is defined as the set C extended with the unit symbol 1. We then define $C^\star$, the set of *labels over* C, as the smallest set containing C^0 and closed under composition ($x, y \in C^\star$ implies $xy \in C^\star$). Labels are defined up to associativity, commutativity and identity 1. Therefore, $aabcc$, $cbaca$ and $cbcaa1$ are simply regarded as equivalent.
A *label constraint* is an expression $x \leq y$ where x and y are labels. A constraint of the form $x \leq x$ is called an *axiom* and we write $x = y$ to express that $x \leq y$ and $y \leq x$. We use the following inference rules to reason on constraints:

$$\frac{x \leq y}{xz \leq yz}\, func \qquad\qquad \frac{x \leq z \quad z \leq y}{x \leq y}\, trans$$

The $trans$ rule formalizes the transitivity of $\leq$ while the $func$ rule corresponds to the functoriality (also called compatibility) of label composition w.r.t. $\leq$. In this formal system, given a constraint k and a set of constraints H, we write $H \mid\!\approx k$ if there is a deduction of k from H. The notation $H \mid\!\approx K$, where K is a non-empty set of constraints, means that for any $k \in K$, $H \mid\!\approx k$.

3.2 Labelled indexed formula tree

Let us recall the standard notions of matrix characterizations of provability [9,16]. In order to illustrate the concepts and results, we consider the **BI** formula $(p * ((q \to r) * s)) \mathrel{-\!\!*} ((p * (q \to r)) * s)$ as an example.
The *formula tree* for a formula A is its representation as a syntax tree the nodes of which are called *positions*. A position u exactly identifies a subformula of A, denoted $f(u)$. An *atomic position* is a position whose formula is atomic. The formula tree induces a partial ordering $\ll$ on positions, such that the root is the least element and if $u \ll v$ then u dominates v (u is in a path from the root to v) in the formula tree. In fact, we do not distinguish between a formula A and its formula tree. Thus, a position u in A is the position in its formula tree. A position u is assigned a polarity $pol(u)$, a principal type $ptyp(u)$ and a secondary type $styp(u)$. We identify in **BI** four principal types $\alpha, \beta, \pi\alpha, \pi\beta$ depending on the principal connective, the polarity and their related secondary types.

Specific concepts for BI. Depending on its principal type, we associate to a position u two notions, namely a label $slab(u)$ and possibly a (label) constraint $kon(u)$. Such a label is either a position or a position with a tilde superscript (in order to identify the formula that introduces the resource) and a label constraint is introduced in order to capture the composition and distribution of the formulae (seen as resources). For a position u, we define its *labelled signed formula* $lsf(u)$ as the triple $(slab(u), f(u), pol(u))$, denoted $slab(u) : f(u)^{pol(u)}$.

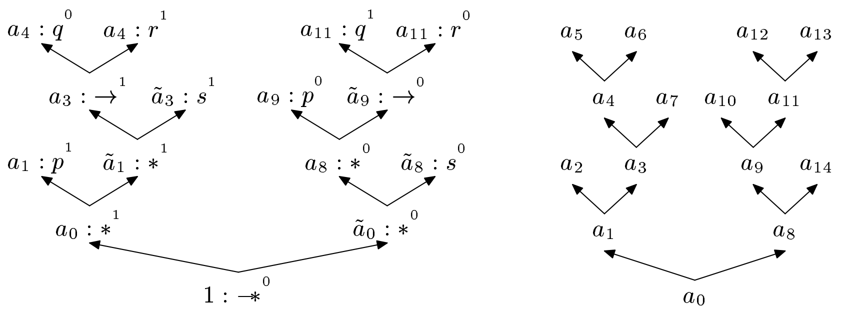
Figure 2 presents the inductive definition of such formulae, knowing that, given a **BI** formula A, we first consider the root position a_0 with polarity $pol(a_0) = 0$ and label $slab(a_0) = 1$, namely, the signed formula $1 : (A)^0$ (1 being the identity of the label composition). Let us note that u_1 and u_2 respectively stand for the first and second child of the position u. For example, if a position u has formula $A * B$, polarity 1 and label x, then u_1 has formula A, polarity 1 and label u while u_2 has formula B, polarity 1 and label $\tilde{u}$. Positions u_1 and u_2 inherit formulae and polarities from parent position u. The principal type of a position u is determined from its principal connective and polarity. Moreover, the labels are inherited by the positions u_1 and u_2, depending on the principal connective and label of u. The label constraint, associated to u, is computed from its principal connective and label.

$lsf(u)$	$ptyp(u)$	$kon(u)$	$lsf(u_1)$	$lsf(u_2)$
$x : (A \wedge B)^1$	α	$-$	$x : A^1$	$x : B^1$
$x : (A \vee B)^0$	α	$-$	$x : A^0$	$x : B^0$
$x : (A \rightarrow B)^0$	$\pi\alpha$	$x \leq u$	$u : A^1$	$u : B^0$
$x : (A \multimap B)^0$	$\pi\alpha$	$xu = \tilde{u}$	$u : A^1$	$\tilde{u} : B^0$
$x : (A * B)^1$	$\pi\alpha$	$u\tilde{u} \leq x$	$u : A^1$	$\tilde{u} : B^1$
$x : (A \wedge B)^0$	β	$-$	$x : A^0$	$x : B^0$
$x : (A \vee B)^1$	β	$-$	$x : A^1$	$x : B^1$
$x : (A \rightarrow B)^1$	$\pi\beta$	$x \leq u$	$u : A^0$	$u : B^1$
$x : (A \multimap B)^1$	$\pi\beta$	$xu = \tilde{u}$	$u : A^0$	$\tilde{u} : B^1$
$x : (A * B)^0$	$\pi\beta$	$u\tilde{u} \leq x$	$u : A^0$	$\tilde{u} : B^0$

Fig. 2. Signed formulae for **BI**

Let us come back to our example. In fact the formula A is first represented as a syntax tree each node of which being identified with a position (tree on the right-hand side of figure 3). Moreover, we can associate an indexed formula tree (with labelled signed formulae as nodes), inductively built from $1 : (A)^0$ and the rules of Figure 2 (tree on the left-hand side of figure 3). In parallel, we have the generation of constraints for positions with principal type $\pi\alpha$ and $\pi\beta$ (see constraints $kon(u)$ in the array of figure 3).

Multiplicity. The **BI** logic deals with resource sharing and distribution. The **LBI** rules $*_R$, $\multimap_L$ and $\rightarrow_L$, related to $\pi\beta$ formulae, split the context and thus distribute the resources. Due to contraction and weakening, we may have to copy the resources (or formulae) before distribution and it could be necessary to have multiple instances of some formulae to achieve proofs. In our connection method for **BI** this necessity is captured by a notion of *multiplicity* μ attached to the $\pi\beta$ formulae. Similarly to [16], we now consider a formula A together with a multiplicity μ and we call this pair an *indexed formula* A^μ. The *indexed formula tree* for A^μ $(indt(A^\mu))$, which is a tree of indexed positions u^κ, κ being a sequence of positive integers, is defined as the following extension of the basic formula tree for A.



u	$pol(u)$	$f(u)$	$ptyp(u)$	$styp(u)$	$slab(u)$	$kon(u)$
a_0	0	$(p * ((q \rightarrow r) * s)) -\!* ((p * (q \rightarrow r)) * s)$	$\pi\alpha$	–	1	$1a_0 = \tilde{a}_0$
a_1	1	$p * ((q \rightarrow r) * s)$	$\pi\alpha$	$\pi\alpha_1$	a_0	$a_1\tilde{a}_1 \leq a_0$
a_2	1	p	–	$\pi\alpha_1$	a_1	–
a_3	1	$(q \rightarrow r) * s$	$\pi\alpha$	$\pi\alpha_2$	$\tilde{a}_1$	$a_3\tilde{a}_3 \leq a_1$
a_4	1	$q \rightarrow r$	$\pi\beta$	$\pi\alpha_1$	a_3	$a_3 \leq a_4$
a_5	0	q	–	$\pi\beta_1$	a_4	–
a_6	1	r	–	$\pi\beta_2$	a_4	–
a_7	1	s	–	$\pi\alpha_2$	$\tilde{a}_3$	–
a_8	0	$(p * (q \rightarrow r)) * s$	$\pi\beta$	$\pi\alpha_2$	$\tilde{a}_0$	$a_8\tilde{a}_8 \leq \tilde{a}_0$
a_9	0	$p * (q \rightarrow r)$	$\pi\beta$	$\pi\beta_1$	a_8	$a_9\tilde{a}_9 \leq a_8$
a_{10}	0	p	–	$\pi\beta_1$	a_9	–
a_{11}	0	$q \rightarrow r$	$\pi\alpha$	$\pi\beta_2$	$\tilde{a}_9$	$\tilde{a}_9 \leq a_{11}$
a_{12}	1	q	–	$\pi\beta_1$	a_{11}	–
a_{13}	0	r	–	$\pi\beta_2$	a_{11}	–
a_{14}	0	s	–	$\pi\beta_2$	$\tilde{a}_8$	–

Fig. 3. Indexed formula tree for $(p * ((q \rightarrow r) * s)) -\!* ((p * (q \rightarrow r)) * s)$

Definition 3.1. *u^κ is an* indexed position *of $indt(A^\mu)$ iff*
1) u is a position of the formula tree for A.
2) let $u_1 \ll \cdots \ll u_n$ be all the $\pi\beta$-positions that dominate u in the formula tree for A, then a) $\mu(u_i) \neq 0$, $1 \leq i \leq n$ and b) $\kappa = m_1 \cdots m_n$, $1 \leq m_i \leq n$, $1 \leq i \leq n$.

For example, having an indexed position u^{23} means that, in the basic formula tree, u is dominated by a $\pi\beta$-position v_2 which is itself dominated by another $\pi\beta$-position v_1. Moreover, in the indexed formula tree, u^{23} is the copy of u which is in the subtree of the third copy of v_2 and of the second copy of v_1. The tree ordering $\ll^\mu$ is given, for any two indexed positions u^κ and v^τ, by the following definition: $u^\kappa \ll^\mu v^\tau$ iff $u \ll v$ and κ is an initial sequence of τ. Given an indexed formula A^μ, we denote $\mathcal{O}cc(u^\kappa)$ the set of indexed positions $u^{\kappa\tau}$ occurring in A^μ. All other notations straightforwardly extend to indexed formula trees and indexed positions. Here, we only consider examples with a multiplicity of 1 and we do not write indexes.

3.3 Paths and connections

We adapt to **BI** the standard notions of path, connection and spanning set of connections defined for classical and intuitionistic logics [16].

Definition 3.2 (Path). *Let A^μ be an indexed formula, u^κ be an indexed position of indt(A^μ) and u_1, u_2 be the immediate successors of u in the basic formula tree of A, the set of* paths *through A^μ is inductively defined as the smallest set such that:*
1. $\{ a_0 \}$ is a path, where a_0 is the root position;
2. if s is a path containing u^κ then
a) if $ptyp(u^\kappa) \in \{ \alpha, \pi\alpha \}$ then, so is $(s \backslash u^\kappa) \cup \mathcal{O}cc(u_1^\kappa) \cup \mathcal{O}cc(u_2^\kappa)$,
b) if $ptyp(u^\kappa) \in \{ \beta, \pi\beta \}$ then, so are $(s \backslash u^\kappa) \cup \mathcal{O}cc(u_1^\kappa)$ and $(s \backslash u^\kappa) \cup \mathcal{O}cc(u_2^\kappa)$.

We say that a path s' obtained from a path s by decomposing a non-atomic position u^κ as explained in case 2 is obtained by *reduction* on u^κ. An *atomic path* is a path that cannot be reduced anymore and therefore only contains atomic positions.

Definition 3.3 (Connection). *A* connection *is a pair $\langle u, v \rangle$ of atomic positions such that $f(u) = f(v)$, $pol(u) = 1$ and $pol(v) = 0$*[1]*.*
A set of connections spans *an indexed formula A^μ iff every atomic path through A^μ contains a connection from this set.*

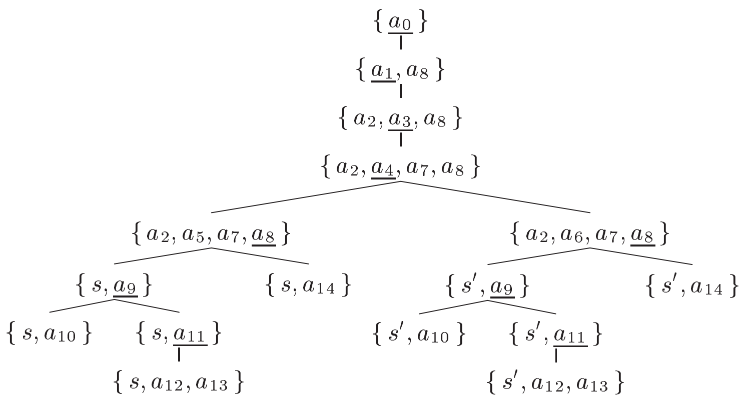


Fig. 4. Path reduction and connection sets

In our example, the reduction of the initial path $\{ a_0 \}$ results in six atomic paths as depicted in figure 4. At each step, we indicate the position which is reduced with an underscore. For conciseness, we write s and s' as shortcuts for a_2, a_5, a_7 and a_2, a_6, a_7. The set $\{ \langle a_2, a_{10} \rangle, \langle a_{12}, a_5 \rangle, \langle a_7, a_{14} \rangle, \langle a_6, a_{13} \rangle \}$ of connections spans the formula A^μ. Let us note that $\langle a_2, a_{10} \rangle$ and $\langle a_7, a_{14} \rangle$ both cover two paths at the same time.

[1] Note that the first position of the connection is the one with polarity 1.

3.4 Characterizing validity in BI

Let A^μ be an indexed formula, Π_α (resp. Π_β) is the set of positions of A^μ with $\pi\alpha$ (resp. $\pi\beta$) as principal type. For $i = 1, 2$, $\Pi_{\alpha i}$ (resp. Π_{β_i}) are the sets of positions of A^μ with $\pi\alpha_i$ (resp. $\pi\beta_i$) as secondary types. $\Sigma_\alpha = \{ \mathit{slab}(u)/u \in (\Pi_{\alpha 1} \cup \Pi_{\alpha 2}) \}$ and $\Sigma_\beta = \{ \mathit{slab}(u)/u \in (\Pi_{\beta_1} \cup \Pi_{\beta_2}) \}$ are respectively the sets of labels corresponding to positions of secondary types $\pi\alpha_i$ and $\pi\beta_i$.
In our example, $\Sigma_\alpha = \{ a_0, \tilde{a}_0, a_1, \tilde{a}_1, a_3, \tilde{a}_3, a_{11} \}$ and $\Sigma_\beta = \{ a_4, a_8, \tilde{a}_8, a_9, \tilde{a}_9 \}$.
Moreover, we split the set of constraints into two sets $\mathcal{K}_\alpha = \{ \mathit{kon}(u)/u \in \Pi_\alpha \}$ and $\mathcal{K}_\beta = \{ \mathit{kon}(u)/u \in \Pi_\beta \}$, by distinguishing constraints introduced by positions with $\pi\alpha$ and $\pi\beta$ as principal type.
Thus, we obtain $\mathcal{K}_\alpha = \{ a_0 = \tilde{a}_0, a_1\tilde{a}_1 \le a_0, a_3\tilde{a}_3 \le \tilde{a}_1, \tilde{a}_9 \le a_{11} \}$ and $\mathcal{K}_\beta = \{ a_8\tilde{a}_8 \le \tilde{a}_0, a_9\tilde{a}_9 \le a_8, a_3 \le a_4 \}$.
The constraints of $\mathcal{K}_\alpha$ are called *assertions* and those of $\mathcal{K}_\beta$ are called *requirements*. We have already used a similar terminology in the labelled tableau calculus defined for **BI** without $\perp$ [6]. In our matrix-characterization, the resource distribution problem is handled via a so-called **BI**-*substitution* which provides a solution to the constraints derived from the set of connections $\mathcal{S}$ by considering the inequality $\mathit{slab}(u) \le \mathit{slab}(v)$ for each connection $\langle u, v \rangle$ in $\mathcal{S}$.
In our example, we have $\mathcal{S} = \{ \langle a_2, a_{10} \rangle, \langle a_{12}, a_5 \rangle, \langle a_7, a_{14} \rangle, \langle a_6, a_{13} \rangle \}$ and we generate the set of constraints $\mathcal{C}on = \{ a_1 \le a_9, a_{11} \le a_4, \tilde{a}_3 \le \tilde{a}_8, a_4 \le a_{11} \}$.

A **BI**-*substitution* is a mapping $\sigma : \Sigma_\beta \to \Sigma_\alpha{}^\star$. Therefore, the elements of Σ_β play the role of variables that must be ground to labels built upon the elements of Σ_α which play the role of constants. It induces a relation $\sqsubset$ on $\Pi_\alpha \times \Pi_\beta$ in the following way: if $\sigma(u) = s$ then $\mathit{pos}(v) \sqsubset \mathit{pos}(u)$ for any v occurring in s, where pos is the function that returns the input u without tildes (i.e., $\mathit{pos}(\tilde{a}_3) = a_3$ and $\mathit{pos}(a_3) = a_3$). From this relation, we define the reduction ordering $\lhd := (\ll \cup \sqsubset)^+$, where $+$ stands for transitive closure and also the set $\mathcal{K}_\alpha^\lhd(u) = \{ \mathit{kon}(v)/v \in \Pi_\alpha \text{ and } u \not\lhd v \}$ for a position u.

Definition 3.4 (admissible BI-substitutions). *Let A^μ be an indexed formula, a* **BI**-*substitution σ is* admissible *w.r.t. A^μ iff*
1. the induced reduction ordering $\lhd$ is irreflexive
2. $\forall\, u \in \Pi_\beta$, $\sigma(\mathcal{K}_\alpha^\lhd(u)) \mathrel{|\!\approx} \sigma(\mathit{kon}(u))$.

The condition 2. means that a **BI**-substitution σ is admissible if the requirements are derivable from the assertions by means of the constraint deduction system (see section 3.1), but with a precise order: the assertions involved in the proof of a requirement must correspond to $\pi\alpha$ positions that may be reduced, with the reduction ordering induced by σ, before the $\pi\beta$ position corresponding to this requirement.

Definition 3.5 (Complementarity). *Let A^μ be an indexed formula and σ be a* **BI**-*substitution, a connection $\langle u, v \rangle$ is said to be* complementary under σ, *or* σ-complementary, *iff $\sigma(\mathcal{K}_\alpha) \mathrel{|\!\approx} \sigma(\mathit{slab}(u)) \le \sigma(\mathit{slab}(v))$.*
A path is σ-complementary *if it contains a σ-complementary connection. A set of paths is σ-complementary if all its paths are σ-complementary.*

Theorem 3.1 (Connection-based characterization). *A formula A of* **BI** *is valid iff there is a multiplicity μ, an admissible* **BI**-*substitution σ and a set of σ-complementary connections that spans A^μ.*

Proof. see the next subsection.

Coming back to our example, from the set of connections that spans A^μ, we must find an admissible **BI**-substitution $\sigma : \Sigma_\beta \to {\Sigma_\alpha}^\star$ which makes each connection σ-complementary, but how to compute such a substitution ?
The easiest way to satisfy the constraints of $\mathcal{C}on$ is to consider them as equalities, rather than inequalities, namely, $\sigma(a_9) = a_1, \sigma(a_4) = a_{11}, \sigma(\tilde{a}_8) = \tilde{a}_3, \sigma(a_4) = a_{11}, \sigma(a_8) = X, \sigma(\tilde{a}_9) = Y$. This substitution makes each connection of $\mathcal{C}on$ σ-complementary. and we have to find $X, Y \in {\Sigma_\alpha}^\star$ such that σ is admissible in the sense of definition 3.4. Hence, we must verify that $\sigma(\mathcal{K}_\alpha) \mid\!\approx \sigma(\mathcal{K}_\beta)$. It corresponds to show that

1. $a_0 = \tilde{a}_0, a_1\tilde{a}_1 \leq a_0, a_3\tilde{a}_3 \leq \tilde{a}_1, Y \leq a_{11} \mid\!\approx X\tilde{a}_3 \leq \tilde{a}_0$
2. $a_0 = \tilde{a}_0, a_1\tilde{a}_1 \leq a_0, a_3\tilde{a}_3 \leq \tilde{a}_1, Y \leq a_{11} \mid\!\approx a_3 \leq a_{11}$
3. $a_0 = \tilde{a}_0, a_1\tilde{a}_1 \leq a_0, a_3\tilde{a}_3 \leq \tilde{a}_1, Y \leq a_{11} \mid\!\approx a_1 Y \leq X$

A trivial solution for condition 2. is to consider $Y \equiv a_3$. Moreover, condition 3. is trivially verified with $X \equiv a_1 a_3$ and condition 1. is also verified, because of

$$\dfrac{\dfrac{\dfrac{a_3\tilde{a}_3 \leq \tilde{a}_1}{a_1 a_3\tilde{a}_3 \leq a_1\tilde{a}_1}\,func \qquad a_1\tilde{a}_1 \leq a_0}{a_1 a_3 \tilde{a}_3 \leq a_0}\,trans \qquad a_0 = \tilde{a}_0}{a_1 a_3 \tilde{a}_3 \leq \tilde{a}_0}\,trans$$

Let us remark that the proof of the constraint of a_8 ($a_1 a_3 \tilde{a}_3 \leq \tilde{a}_0$) does not depend on the constraint of a_{11} ($a_3 \leq a_{11}$) that is the only one that is $\lhd$-greater than a_8. Therefore, we deduce that $\sigma(a_8) = a_1 a_3$ and $\sigma(\tilde{a}_9) = a_3$.
Finally, we have to show that the reduction ordering induced by σ is irreflexive. In fact, the relation $\sqsubset$ induced by σ is: $a_{11} \sqsubset a_4$, $a_1 \ll a_3 \sqsubset a_8 \ll a_9$. Combined with the relation $\ll$, which represents the tree ordering on the positions, we have the relation $\lhd$ depicted on figure 5, that is easily proved irreflexive (acyclic). The dotted arrows correspond to the parts of the relation coming from the substitution. Since the graph is acyclic, σ is admissible, and so, the formula is valid.
The present characterization can be extended to take the units into account. There is no problem with $\top$ and I since we can respectively translate all of their occurrences inside a formula A by the formulae $(\alpha \to \alpha)$ and $(\beta \mathrel{-\!\!*} \beta)$, where α and β are new propositional variables not occurring in A. The treatment of $\perp$ is more delicate [11] and needs the following additions to the notions of connection and complementarity: a $\perp$-*connection* is a pair $\langle u, v \rangle$ of atomic positions such that $pol(u) = 1$, $pol(v) = 0$ and $f(u) = \perp$. Let A^μ be an indexed formula and σ be a **BI**-substitution, a $\perp$-connection $\langle u, v \rangle$ is said to be *complementary* under σ iff $\sigma(\mathcal{K}_\alpha) \mid\!\approx \sigma(slab(u))x \leq \sigma(slab(v))$ for some label $x \in {\Sigma_\alpha}^\star$. The next proof of our characterization is given for **BI** without units but with the previous extensions it can be easily adapted to **BI** with units.

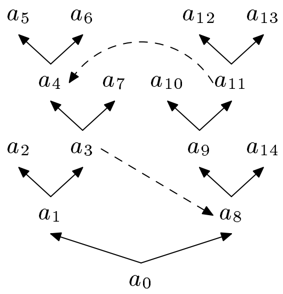


Fig. 5. Reduction ordering

3.5 Proof of the matrix characterization

Let us give the proof of the previous characterization of **BI** provability. It can be split in two parts, namely soundness and completeness. The proof of soundness follows the ideas given in [16] for modal logics, but we have to recall some points about **BI** validity. A *Kripke resource interpretation* $\mathcal{K} = \langle\, (M, \sqsubseteq, \bullet, e), \models, [\![-]\!]\,\rangle$ is a *Kripke resource monoid* $\langle\, (M, \sqsubseteq, \bullet, e)\,\rangle$, i.e., a preordered monoid in which $\bullet$ is order preserving (or functorial w.r.t. $\sqsubseteq$), together with a forcing relation $\models$ parametrized by an interpretation $[\![-]\!]$ of propositional variables. A formula A is *valid in* $\mathcal{K}$ iff $e \models A$. A formula A is *valid* if it is valid for all $\mathcal{K}$.
Let us introduce some appropriate definitions. A *reduction* is a finite sequence $(\mathcal{S}_i)_{0\leq i\leq n} = \mathcal{S}_0, \cdots, \mathcal{S}_n$ where $\mathcal{S}_0$ is a set of paths and for any $i \geq 0$, $\mathcal{S}_{i+1}$ is obtained by reducing a position u_i in a path s_i of $\mathcal{S}_i$. We say that $\mathcal{S}_{i+1}$ is obtained by reduction of $\mathcal{S}_i$ on (s_i, u_i). Note that, given $\mathcal{S}_0$, $\mathcal{S}_{i+1}$ is completely determined by the sequence $(s_0, u_0), \cdots, (s_i, u_i)$. Moreover, for each $\mathcal{S}_i$, we define $\Pi_\alpha(\mathcal{S}_i)$ as the set of all the $\pi\alpha$-positions that were reduced before reaching $\mathcal{S}_i$ and $\Sigma_\alpha(\mathcal{S}_i)$ as the language of labels introduced by those reductions. More formally, $\Pi_\alpha(\mathcal{S}_i) = \{\, u_j/(j < i)$ and $ptyp(u_j) = \pi\alpha\,\}$ and $\Sigma_\alpha(\mathcal{S}_i) = \bigcup_{1\leq j\leq i}\{\, slab(u)/u \in s, s \in \mathcal{S}_j\,\}$. $\mathcal{K}_\alpha(\mathcal{S}_i) = kon(\Pi_\alpha(\mathcal{S}_i))$ is the set of label-constraints introduced by the positions in $\Pi_\alpha(\mathcal{S}_i)$. Similarly for $\Pi_\beta(\mathcal{S}_i)$,$\Sigma_\beta(\mathcal{S}_i)$ and $\mathcal{K}_\beta(\mathcal{S}_i)$ w.r.t. $\pi\beta$-positions. A reduction $(\mathcal{S}_i)_{0\leq i\leq n}$ is *complete* if it cannot be further reduced, *i.e.*, if $\mathcal{S}_n$ only contains atomic paths.

Definition 3.6. *Let σ be a* **BI***-substitution, a reduction $(\mathcal{S}_i)_{0\leq i\leq n}$ is σ-*proper *if (i) for any $i \geq 0$, $\sigma(\mathcal{K}_\alpha(\mathcal{S}_i)) \models\!\!\approx \sigma(\mathcal{K}_\beta(\mathcal{S}_i))$ and (ii) for any $j > i \geq 0$, $u_j \not\lhd u_i$.*

Lemma 3.1. *A* **BI***-substitution σ is admissible iff there exists a σ-proper reduction from $\{\,\{\,a_0\,\}\,\}$, where $\{\,a_0\,\}$ is the initial path containing the root position.*

Proof. The proof is direct since condition (ii) can only be satisfied if σ is irreflexive and condition (i) means that the assertions needed to deduce a requirement associated to a $\pi\beta$-position are the ones corresponding to previously reduced $\pi\alpha$-positions, *i.e.*, $\pi\alpha$-positions that are not $\lhd$-greater than this $\pi\beta$-position.

Definition 3.7. *Let σ be a* **BI***-substitution and $(\mathcal{S}_i)_{0\leq i\leq n}$ be a complete σ-proper reduction. An* interpretation *of $\mathcal{S}_i$ in a Kripke resource model $\mathcal{K}$ denoted $\langle\,(M,\sqsubseteq,\bullet,e),\models,[\![-]\!]\,\rangle$ is a mapping $\bar{\iota}:\Sigma_\alpha(\mathcal{S}_i)\rightarrow M$. The mapping ι obtained by homomorphic extension of $\bar{\iota}$ to $\Sigma_\alpha(\mathcal{S}_i)^\star$ is also called an interpretation.*
An interpretation ι realizes *$\mathcal{S}_i$* under *σ, or $\mathcal{S}_i$ is σ-realizable, if there is a path s in $\mathcal{S}_i$ such that:*

1. *for any $x\leq y\in\mathcal{K}_\alpha(\mathcal{S}_i)$, $\iota\sigma(x)\sqsubseteq\iota\sigma(y)$.*
2. *for any $u\in s$ such that $lsf(u)=x:A^1$, $\iota\sigma(x)\models A$.*
3. *for any $u\in s$ such that $lsf(u)=x:A^0$, $\iota\sigma(x)\not\models A$.*

$\mathcal{S}_i$ is σ-realizable if it is σ-realizable for some interpretation.

Lemma 3.2. *If $\mathcal{S}_i$ is σ-realizable then $\mathcal{S}_{i+1}$ is σ-realizable.*

Proof. By case analysis on the position which is reduced.

Lemma 3.3. *If $(\mathcal{S}_i)_{0\leq i\leq n}$ is a complete σ-proper reduction then, if $\mathcal{S}_n$ is σ-complementary it is not σ-realizable.*

Proof. Suppose $\mathcal{S}_n$ is σ-realizable for some interpretation ι. There is an atomic path $s\in\mathcal{S}_n$ which satisfies the conditions of definition 3.7. But, since $\mathcal{S}_n$ is σ-complementary, s contains a connection $\langle\,u,v\,\rangle$ such that, $f(u)=f(v)$, $pol(u)=1$, $pol(v)=0$ and $\sigma(\mathcal{K}_\alpha)\models\!\!\!\approx\sigma(slab(u))\leq\sigma(slab(v))$. $\mathcal{S}_n$ is σ-realizable and we have $\iota\sigma(slab(u))\models f(u)$, $\iota\sigma(slab(v))\not\models f(u)$.
Moreover, since the reduction is complete, $\mathcal{K}_\alpha=\mathcal{K}_\alpha(\mathcal{S}_n)$ and then $\iota\sigma(slab(u))\sqsubseteq\iota\sigma(slab(v))$, which yields a contradiction because we should have $\iota\sigma(slab(v))\models f(u)$ by Kripke monotonicity.

Theorem 3.2 (Soundness). *Let A be a* **BI** *formula, if there is a multiplicity μ, an admissible* **BI***-substitution σ and a set of σ-complementary connections that spans A^μ then A is valid.*

Proof. Let us suppose that there is a multiplicity μ and an admissible **BI**-substitution σ which makes the set of atomic paths through A^μ σ-complementary. Let us assume that A does not hold in Kripke resource semantics. Then, there exists $\mathcal{K}=\langle\,(M,\sqsubseteq,\bullet,e),\models,[\![-]\!]\,\rangle$ such that $e\not\models A$ and the set $\mathcal{S}_0=\{\,\{\,a_0\,\}\,\}$ is therefore trivially σ-realizable. Since σ is admissible, lemma 3.1 ensures that there is a complete σ-proper reduction from $\mathcal{S}_0$. Because $\mathcal{S}_0$ is realizable, by lemma 3.2 the set of atomic paths is also realizable. Lemma 3.3 finally entails that this set cannot be σ-complementary, which is a contradiction.

Let us now consider the completeness. We say that a formula A is *connection-valid* if there is a multiplicity μ, an admissible **BI**-substitution σ and a set of σ-complementary connections that spans A^μ. A sequent $\Gamma\vdash A$ is *connection-valid* if the formula $\Phi_\Gamma \mathrel{-\!\!*} A$ is connection-valid, where Φ_Γ is the formula obtained from Γ by replacing each $\emptyset_a$ by $\top$, each $\emptyset_m$ by I, each ";" by $\wedge$ and each "," by $*$. Now, we prove completeness by showing that connection-validity is closed under each rule of the **LBI** sequent calculus.

Lemma 3.4. *if $\Gamma \vdash A$ is* **LBI**-*provable then it is connection-valid.*

Proof. By induction on the derivation in **LBI** (see appendix A).

Theorem 3.3 (Completeness). *If a* **BI** *formula A is valid then there is a multiplicity μ, an admissible* **BI**-*substitution σ and a set of σ-complementary connections that spans A^μ.*

Proof. If A is valid then $\emptyset_m \vdash A$ is provable in **LBI**. Then, by lemma 3.4, we deduce that $I \twoheadrightarrow A$ is connection-valid and since $I \twoheadrightarrow A \dashv\vdash A$, there is a multiplicity μ, an admissible **BI**-substitution σ and a set of σ-complementary connections that spans A^μ.

3.6 The related connection method

From this connection characterization of **BI** provability and as illustrated in the previous example, we can define a connection-based proof search method for **BI**. For that, we can follow the standard steps of connection methods proposed for intuitionistic logic or linear logic [9,16] but replace the procedures dedicated to the treatment of prefixes by procedures dealing with constraints. In the context of **BI** where intuitionistic and linear connectives cohabit, labels and constraints that capture the semantic consequence relation of the logic appears as a necessary generalization of prefixes that mainly capture the permutabilities of rules in a sequent calculus.
Let us focus now on the specific treatment of constraints in our method. A first step consists in computing the **BI**-substitutions (from a set of connections). As illustrated in the example, it mainly requires an unification algorithm on labels. First, we could use an existing T-string unification algorithm [9] or an AC-algorithm [4,13] on our labels. Further work will focus on this step in order to analyze if it is necessary to propose other unification algorithms, specially dedicated to our constraint-based method. The second step consists in proving the admissibility of such a substitution, *i.e.*, whether requirements hold w.r.t. assertions. For that, as illustrated in the example, we have to develop bottom-up proof search in the formal system including the inference rules *func* and *trans*, presented in section 3.1. What we can say about proof search in this calculus, in this step, is that it can be based on the following order of rule applications, namely, from the bottom we have all *trans* rules first and then a *func* rule with axioms or assertions as premises. Moreover, in the *trans* rules, the right-hand side premiss is always an assertion. Consequently, it reduces the possible instantiations of the left-hand side premiss $x \leq z$ from the known constraint $z \leq y$. From these results, that are not detailed here, we have significant improvements of the standard bottom-up proof search in the constraint calculus. This analysis of two specific steps of our method will lead to a first implementation that can be based on known and improved algorithms [9] but with specific modules dedicated to constraint generation and solving. Further work will be devoted to such an implementation with, in addition, a study of the effective use of multiplicity in this proof-search process. Here, we do not give an explicit bound but

if we have enough copies of $\pi\beta$ formulae to have each atom involved in only one connection, we can show that it is sufficient to decide a **BI** formula with our method.

4 A new connection-based characterization for IL

One important feature about **BI** is that it is conservative w.r.t. **IL** and **MILL** logics [11]. Thus, we propose a new characterization of **IL** provability which is directly derived from **BI** by restricting the formulae to the additive connectives. A main consequence is that $\pi\alpha$ and $\pi\beta$ positions respectively correspond to signed formulae of the form $(A \to B)^0$ and $(A \to B)^1$. Looking at figure 2, we can notice that such a restriction discards the need for atomic labels with a tilde and leads to constraints dealing only with *atomic* labels. Concerning the constraint deduction, this simplified constraints allow to forget about the *func* rule. Since the only rule is *trans*, checking whether a requirement is derivable from a set of assertions simply means checking whether this requirement belongs to the transitive closure of the assertions, which is a significant simplification. The necessary adaptations to substitutions, complementarity and validity are direct. All other definitions remain unchanged.
Let A^μ be an indexed formula, a **IL**-*substitution* is a mapping $\sigma : \Sigma_\beta \to \Sigma_\alpha$. The definitions of admissibility and complementarity are the same as for **BI** but involves the new (restricted) definition of **IL**-substitution. Thus, we obtain a new matrix characterization for **IL**.

Theorem 4.1 (Connection-based characterization). *A* **IL** *formula A is valid iff there is a multiplicity μ, an admissible* **IL**-*substitution σ and a set of σ-complementary connections that spans A^μ.*

The connection method for **BI** can directly be adapted to provide a new connection method for **IL**. Here, the label constraints involve either two Σ_α labels, two Σ_β labels or a Σ_α label and a Σ_β label. Therefore, the unification step is drastically simplified and the computation of **IL**-substitutions becomes quite immediate. Then, in the case of **IL**, the main step lies in the verification of the admissibility of the substitutions, which reduces to checking if the requirements belong to the transitive closure of the assertions.
Standard matrix characterizations of **IL** [9,16] consider prefixes that essentially describe the position of a subformula in the syntax tree. The order of rule applications in the search process is captured by an intuitionistic substitution which makes the prefixes of connected subformulae identical and a specialized string unification is required for this step. Deriving our method for **IL** from **BI** provides an alternative to this approach by considering constraints in place of prefixes. Because of the simple form of the constraints in **IL**, we can refine and improve the specific steps dedicated to the constraints.
Further work will be devoted to a more accurate study of the constraints solving in **IL**. This new method emphasizes that, in a logic with a resource semantics like **BI** but also **IL**, labels can be used in order to capture, in a simple way,

its consequence relation. It seems to be a more general approach than trying to capture the right order of rules applications in the sequent calculus, with prefixes for instance. This point will be more deeply analyzed but the interest of constraint-based proof calculi [15] for substructural logics is clearly confirmed.

5 A new connection-based characterization for MILL

There exists connection-based characterizations and related connection methods for multiplicative (commutative) linear logic (**MLL**) [5,8] but not for its intuitionistic fragment, namely **MILL**. The matrix characterization proposed in [8] is based on particular prefixes and substitutions dedicated to **MLL**. In order to extend or adapt it to **MILL**, it would be necessary to add intuitionistic prefixes but it seems to be difficult and not natural. Since **BI** is conservative over **MILL** [11], we can define the first connection-based characterization for **MILL**, by restriction of the one for **BI** to the multiplicative connectives (see table 2). We only have to adapt all the definitions with such a restriction.

Theorem 5.1 (Connection-based characterization). *A* **MILL** *formula A is valid iff there is a multiplicity μ, an admissible* **MILL**-*substitution σ and a set of σ-complementary connections that spans A^μ.*

This result emphasizes the power of labels and constraints that are more natural and accurate than prefixes to capture the semantic consequence relation and consequently the order of rule applications in the related sequent calculus of **MILL**. From this characterization of provability, we can propose, like in **BI** a connection-based proof search method for **MILL**. As an example, we can consider the **BI** formula $(p * ((q \to r) * s)) -\!\!* ((p * (q \to r)) * s)$ in which we replace $\to$ by $-\!\!*$, namely $(p * ((q -\!\!* r) * s) -\!\!* ((p * (q -\!\!* r)) * s)$. The development of the method on the example is the same as for **BI** with minor changes on the constraints to satisfy and thus we can show that this formula is provable.
We have previously defined a connection-based characterization of provability in **MLL** and studied the relationships between connection methods and proof nets construction in linear logic fragments [5]. We can show, like for **MLL**, that a connection-based method for **MILL** provides an algorithm for the construction of **MILL** proof nets [2]. Conversely, we have defined a connection method for **MLL** that is based on the automatic construction of proof nets (and of sequent proofs in parallel). This method starts with the formula (or decomposition) tree and builds, step by step and automatically, axiom-links (or connections) following particular strategies. We can define such an algorithm for **MILL** by taking into account labels and constraints attached to different positions in the formula tree. In this case, some steps of proof nets construction are only possible if some constraints are satisfied. It can be seen as another new connection method for **MILL**, that is based on proof nets construction.

6 Conclusion

In this paper, we define a connection-based characterization of propositional **BI** (logic of bunched implications) and a related proof-search method. In order to avoid redundancies of a labelled tableau method for propositional **BI** (without $\perp$) [6], we design a connection method in which we introduce specific labels and constraints which naturally capture the semantic consequence relation of the logic. The fact to have such a simple connection method w.r.t. a sequent calculus with bunches emphasizes the power and the adequacy of methods based on labels and constraints, compared to more standard methods based on prefixes. As **BI** is conservative w.r.t. **IL** and **MILL** we also define, by restriction and refinements, new connection-based characterizations of provability for both logics. Including labels and constraints into connection-based methods appears as a necessary and general approach for some substructural logics as illustrated by **BI** and **MILL**. Further work will be devoted to the implementation of the method, mainly to the steps concerning constraints generation and solving. Moreover, we will study the problems about the management of the multiplicity in such methods.

References

1. P.A. Armelin and D. Pym. Bunched logic programming (extended abstract). In *First International Joint Conference on Automated Reasoning, IJCAR 2001, LNCS 2083*, pages 289–304, Siena, Italy, 2001.
2. V. Balat and D. Galmiche. *Labelled Deduction*, volume 17 of *Applied Logic Series*, chapter Labelled Proof Systems for Intuitionistic Provability. Kluwer Academic Publishers, 2000.
3. W. Bibel. On matrices with connections. *Journal of ACM*, 28(4):633–645, 1981.
4. K. Broda, M. Finger, and A. Russo. LDS-natural deduction for substructural logics (extended abstract). *Logic Journal of the IGPL*, 4(3):486–489, 1996.
5. D. Galmiche. Connection Methods in Linear Logic and Proof nets Construction. *Theoretical Computer Science*, 232(1-2):231–272, 2000.
6. D. Galmiche and D. Méry. Proof-search and countermodel generation in propositional BI logic - extended abstract -. In *4th Int. Symp. on Theoretical Aspects of Computer Software, TACS 2001, LNCS 2215*, pages 263–282, Sendai, Japan, 2001.
7. S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *28th ACM Symposium on Principles of Programming Languages, POPL 2001*, pages 14–26, London, UK, 2001.
8. C. Kreitz, H. Mantel, J. Otten, and S. Schmitt. Connection-based proof construction in linear logic. In *14th Int. Conference on Automated Deduction*, pages 207–221, Townsville, North Queensland, Australia, 1997.
9. C. Kreitz and J. Otten. Connection-based theorem proving in classical and non-classical logics. *Journal of Universal Computer Science*, 5(3):88–112, 1999.
10. P. O'Hearn. Resource interpretations, bunched implications and the $\alpha\lambda$-calculus. In *Typed Lambda Calculi and Applications, TLCA'99, LNCS 581*, pages 258–279, L'Aquila, Italy, 1999.
11. P.W. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

12. D. Pym. On bunched predicate logic. In *14h Symposium on Logic in Computer Science*, pages 183–192, Trento, Italy, July 1999. IEEE Computer Society Press.
13. M. E. Stickel. A unification algorithm for associative-commutative functions. *Journal of ACM*, 28(3):423–434, 1981.
14. A. Urquhart. Semantics for relevant logic. *Journal of Symbolic Logic*, 37:159–169, 1972.
15. A. Voronkov. Proof-search in intuitionistic logic based on constraint satisfaction. In *Int. Workshop Tableaux'96, LNAI 1071*, pages 312–327, Terrasini, Italy, 1996.
16. L.A. Wallen. *Automated Proof search in Non-Classical Logics.* MIT Press, 1990.

A Completeness of the BI characterization

We say that a formula A is *connection-valid* if there is a multiplicity μ, an admissible **BI**-substitution σ and a set of σ-complementary connections that spans A^μ. To prove completeness we show that the connection-validity is closed under each rule of the **LBI** sequent calculus.

Lemma A.1. *if $\Gamma \vdash A$ is* **LBI***-provable then it is connection-valid.*

Proof. The proof is by induction on the derivation in **LBI**.

- base case: an axiom $A \vdash A$, corresponding to $A \mathrel{-\!\!*} A$, is connection-valid.
- induction step: let us consider the $\mathrel{-\!\!*}_R$ case.
  By induction hypothesis, we suppose that $\Gamma, A \vdash B$ is connection-valid and we show that $\Gamma \vdash A \mathrel{-\!\!*} B$ is also connection-valid.
  Since $\Gamma, A \vdash B$ is connection-valid, there exists a multiplicity μ and an admissible **BI**-substitution σ which makes σ-complementary every atomic path through $((\Phi_\Gamma * A) \mathrel{-\!\!*} B)^\mu$. Therefore, we can construct a complete σ-proper reduction $\mathcal{R}_1$ from $\{\{a_0\}\}$, where $lsf(a_0) = 1 : ((\Phi_\Gamma * A) \mathrel{-\!\!*} B)^0$, which results in σ-complementary atomic paths. From this reduction $\mathcal{R}_1$ we build a complete σ-proper reduction $\mathcal{R}_2$ for $\{\{a_0\}\}$, where $lsf(a_0) = 1 : (\Phi_\Gamma \mathrel{-\!\!*} (A \mathrel{-\!\!*} B))^0$. In the following figure, the first steps of $\mathcal{R}_1$ are shown on the left-hand side, while those of $\mathcal{R}_2$ are shown on the right-hand side. For clarity, we have represented paths as sets of labelled signed formulae rather than sets of positions.

$$\begin{array}{c|c}
\{\, 1 : ((\Phi_\Gamma * A) \mathrel{-\!\!*} B)^0 \,\} & \{\, 1 : (\Phi_\Gamma \mathrel{-\!\!*} (A \mathrel{-\!\!*} B))^0 \,\} \\
| & | \\
\{\, a_0 : (\Phi_\Gamma * A)^1, \tilde{a}_0 : B^0 \,\} & \{\, a_0 : {\Phi_\Gamma}^1, \tilde{a}_0 : A \mathrel{-\!\!*} B^0 \,\} \\
| & | \\
\{\, a_1 : {\Phi_\Gamma}^1, \tilde{a}_1 : A^1, \tilde{a}_0 : B^0 \,\} & \{\, a_1 : {\Phi_\Gamma}^1, a_i : A^1, \tilde{a}_i : B^0 \,\} \\
| & | \\
\vdots & \vdots
\end{array}$$

  Both reductions result in a path containing the same signed formulae. The only problem is that they have distinct labels. However, if the relationships between the labels of $\mathcal{R}_2$ are the same or stronger than those between labels of $\mathcal{R}_1$, then, any constraint of $\mathcal{R}_2$, corresponding to a constraint of $\mathcal{R}_1$ which is derivable in $\mathcal{R}_1$, is derivable in $\mathcal{R}_2$. Consequently, if $\mathcal{R}_1$ can be properly continued to reach σ-complementary atomic paths, reusing the same continuation process for $\mathcal{R}_2$ results, upto the labels, in the same atomic paths. Moreover, these paths are also σ-complementary by the previous argument on the strength of the relationships. Now, the assertions generated by $\mathcal{R}_1$ after the first three steps are $\{ a_0 = \tilde{a}_0, a_1\tilde{a}_1 \leq a_0 \}$, while $\mathcal{R}_2$ generates $\{ a_0 = \tilde{a}_0, \tilde{a}_0 a_i = \tilde{a}_i \}$. If we rewrite the labels more explicitly, those two sets respectively correspond to $\{ slab(\Phi_\Gamma * A) = slab(B), slab(\Phi_\Gamma)slab(A) \leq slab(\Phi_\Gamma * A) \}$ and $\{ slab(\Phi_\Gamma) = slab(A \mathrel{-\!\!*} B), slab(A \mathrel{-\!\!*} B)slab(A) = slab(B) \}$. Simplifying synonyms introduced via the $\mathrel{-\!\!*}$ connective, we can see that the relationship assumed in $\mathcal{R}_1$ is in fact $slab(\Phi_\Gamma)slab(A) \leq slab(B)$, while $\mathcal{R}_2$ assumes $slab(\Phi_\Gamma)slab(A) = slab(B)$, which is obviously stronger.

The other cases are similar or immediate by the translation of bunches.

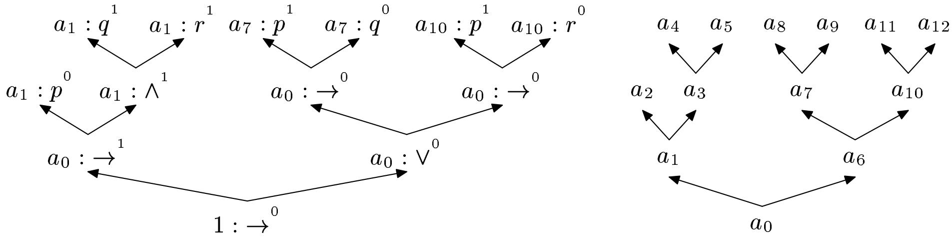
Theorem A.1 (Completeness). *If a formula A of* **BI** *is valid then there is a multiplicity μ, an admissible* **BI***-substitution σ and a set of σ-complementary connections that spans A^μ.*

Proof. If A is valid then $\emptyset_m \vdash A$ is provable in **LBI**. Then, by lemma A.1, we deduce that $I \mathrel{-\!\!*} A$ is connection-valid and since $I \mathrel{-\!\!*} A \dashv\vdash A$, there is a multiplicity μ, an admissible **BI**-substitution σ and a set of σ-complementary connections that spans A^μ.

B An example of the IL characterization

Let us illustrate the new matrix characterization of **IL** provability with the following formula of **IL**: $A^\mu = (p \to (q \wedge r)) \to ((p \to q) \vee (p \to r))$.
The indexed formula tree is given in figure 6. Then we have the two sets of constraints $\mathcal{K}_\alpha = \{ 1 \leq a_0, a_0 \leq a_7, a_0 \leq a_{10} \}$ and $\mathcal{K}_\beta = \{ a_0 \leq a_1 \}$.
By application of the path reduction process to $\{ a_0 \}$, we obtain two atomic paths, namely, $s_1 = \{ a_2, a_8, a_9, a_{11}, a_{12} \}$ and $s_2 = \{ a_4, a_5, a_8, a_9, a_{11}, a_{12} \}$. The path s_1 contains two connections $\langle a_8, a_2 \rangle$ and $\langle a_{11}, a_2 \rangle$ and s_2 contains $\langle a_4, a_9 \rangle$ and $\langle a_5, a_{12} \rangle$. Let us choose $\langle a_8, a_2 \rangle$ for s_1 and $\langle a_4, a_9 \rangle$ for s_2. From these connections, we can directly derive the following set of constraints $\mathcal{C}on = \{ a_7 \leq a_1, a_1 \leq a_7 \}$ is directly derived.
Now, we must find an admissible **IL**-substitution. For that, we compute the two sets Σ_α and Σ_β which are respectively equal to $\{ a_0, a_7, a_{10} \}$ and $\{ a_1 \}$. By transitivity on constraints of $\mathcal{C}on$, we deduce that $\sigma(a_1) = a_7$. The next point is to show that $\sigma(\mathcal{K}_\alpha) \mathrel{|\!\!\approx} \sigma(\mathcal{K}_\beta)$. Since $a_0 \leq a_1$ is the only requirement and $\sigma(a_1) = a_7$, we only need to prove that $a_0 \leq a_7$ holds w.r.t. to the assertions, which is immediate because $a_0 \leq a_7$ is an assertion. The substitution $\sigma(a_1) = a_7$ induces the relation $a_7 \sqsubset a_1$, which, combined with the tree ordering of the positions, yields the relation $\lhd$ shown in figure 7. As the reader might check, the resulting graph is acyclic and the formula A^μ is therefore valid in **IL**.

This example illustrates how our connection method based on labels and constraints is an alternative to the standard methods based on prefixes as mentioned in section 4.



u	$pol(u)$	$f(u)$	$ptyp(u)$	$styp(u)$	$slab(u)$	$kon(u)$
a_0	0	$(p \to (q \wedge r)) \to ((p \to q) \vee (p \to r))$	$\pi\alpha$	–	1	$1 \leq a_0$
a_1	1	$p \to (q \wedge r)$	$\pi\beta$	$\pi\alpha_1$	a_0	$a_0 \leq a_1$
a_2	0	p	–	$\pi\beta_1$	a_1	–
a_3	1	$q \wedge r$	α	$\pi\beta_2$	a_1	–
a_4	1	q	–	α_1	a_1	–
a_5	1	r	–	α_1	a_1	–
a_6	0	$(p \to q) \vee (p \to r)$	α	$\pi\alpha_2$	a_0	–
a_7	0	$p \to q$	$\pi\alpha$	α_1	a_0	$a_0 \leq a_7$
a_8	1	p	–	$\pi\alpha_1$	a_7	–
a_9	0	q	–	$\pi\alpha_2$	a_7	–
a_{10}	0	$p \to r$	$\pi\alpha$	α_2	a_0	$a_0 \leq a_{10}$
a_{11}	1	p	–	$\pi\beta_1$	a_{10}	–
a_{12}	0	r	–	$\pi\beta_2$	a_{10}	–

Fig. 6. Indexed formula tree for $(p \to (q \wedge r)) \to ((p \to q) \vee (p \to r))$

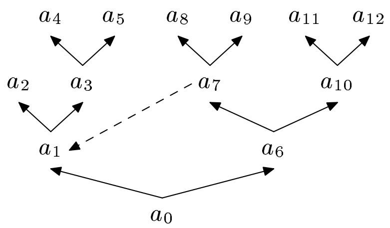


Fig. 7. Reduction ordering for $(p \to (q \wedge r)) \to ((p \to q) \vee (p \to r))$